Alma Mater Studiorum - Universitá di Bologna
Universitá degli Studi di Padova

Dottorato di Ricerca in Informatica
XXVI Ciclo

Opportunistic Data Gathering and Dissemination in Urban Scenarios

Author:
Armir Bujari

Supervisor:
Claudio E. Palazzi
Co-Supervisor:
Tullio Vardanega

Ph.D. Coordinator:
Prof. Maurizio Gabbrielli

Settore Concorsuale di afferenza: 01/B1
Settore Scientifico disciplinare: INF/01

Esame Finale Anno 2014

I dedicate this thesis to

my family and long standing friends

for their constant support and unconditional love.

I love you all dearly.

Acknowledgements

After finishing a journey, you look back and realize how it changed you, how it helped you grow, and how it helped shape who you are. This has been undoubtedly a magnificent experience which has enriched me both personally and professionally.

Foremost, I would like to express my sincere gratitude to my advisor Prof. Claudio Enrico Palazzi for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and knowledge. His guidance helped me in all the time of my research activity and the writing of this dissertation.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Tullio Vardanega, Prof. Dario Maggiorini, for their encouragement, insightful comments, and hard questions.

My sincere thanks also goes to the UCLA Network Research Laboratory team, especially to Prof. Mario Gerla and Prof. Giovanni Pau, for offering to me the opportunity to work in their groups and leading me working on diverse exciting projects.

I thank my fellow lab colleagues at the University of Padua for the stimulating discussions and for all the fun we have had in the last three years.

Last but not the least, I would like to thank my family: my parents Urim and Lida Bujari and my sister Ada for their support and sustain.

Abstract

Only a decade ago, digital information was mainly confined to electrical wires, traveling bits around the Internet, while today we are immersed in a digital fountain and information is being produced everywhere around us. In the era of the Internet of Everything, a user with a handheld or wearable device (e.g., Smartphone or even Google Glass) equipped with sensing capability has become a producer as well as a consumer of information and services. The more powerful these devices get, the more likely it is that they will generate and share content locally, leading to the presence of distributed information sources and the diminishing role of centralized servers.

As of current practice, we rely on infrastructure acting as an intermediary, providing access to the data. However, infrastructure-based connectivity might not always be available or the best alternative (e.g., it is cost-attributed). Moreover, it is often the case where the data and the processes acting upon them are of local *scopus*. Answers to a query about a nearby object, an information source, a multimedia content, a process, a person, an experience, an ability, etc. could be answered locally without reliance on infrastructure communication platforms. The data might have temporal validity limited to or bounded to a geographical area and/or the social context where the user is immersed in.

Altogether, these characteristics are in contrast to current location, infra-structure-based systems which rely on the user to go through and declare its location and identity on the infrastructure. In our envisioned scenarios users could interact locally without the need for a central authority, hence, the claim of an infrastructure-less, provider-less communication platform. The data is owned by the users and consulted locally as opposed to the current approach of making them available globally

and stay on forever.

Despite this nice portrait, the mobile and volatile nature of this network presents severe challenges demanding new networking techniques able to cope with the unpredictable and resource-constrained nature of mobile entities comprising it. From this basis, ad hoc networking could be employed as an enabler which can factually support an infrastructure-less, provider-less communication platform encompassing the above requirements. From a technical viewpoint, this network resembles a Delay/Disruption Tolerant Network (DTN) where consumers and producers might be spatially and temporally decoupled exchanging information with each other in an ad hoc, delay-tolerant fashion. Terrestrial DTNs are by nature local making them a perfect candidate at supporting these kind of communication platform.

To this end, we propose and contribute with novel data gathering and dissemination strategies for use in urban-wide environments which do not rely on strict infrastructure mediation. While preserving the general aspects of our study and without loss of generality, we focus our attention toward practical applicative scenarios which help us capture the characteristics of opportunistic communication networks, and devise efficient solutions for use in real applicative contexts.

Table of Contents

List of Figures

List of Tables

List of Acronyms

DTN	Delay/Disruption Tolerant Network
IoE	Internet of Everything
UGC	User Generated Content
ICT	Information and Communication Technology
MANET	Mobile Ad hoc NETworks
OppNet	OPPortunistic NETworks
NDN	Named Data Networking
P2P	Peer-to-peer
PTS	Public Transportation System
TCP/IP	Transport Control Protocol/Internet Protocol
UDP	User Datagram Protocol
RFC	Request For Comment
IPN	InterPlanetary Internet
tDTN	terrestrial-Delay/Disruption Tolerant Networks
PSN	Pocket Switched Network
BSN	Bus Switched Network
ICN	Information-Centric Networking
DHT	Distributed hash tables

PROPHET	PRObabilistic Protocol using History of Encounters and Transitivity
ORION	Optimized Routing Independent Overlay
AODV	Ad hoc On demand Distance Vector
SN	Super Node
ON	Ordinary Node
MPP	Mobile Peer-to-Peer
DSR	Dynamic Source Routing
RSSI	Received Signal Strength Indication
AP	Access Point
M2MShare	Mobile-to-Mobile Content Sharing
MAC	Medium Access Control
CRT	Content Routing Table
PQT	Pending Query Table
ONE	Opportunistic Network Environment
MDTN	Mobile Delay/Disruption Tolerant Network
IG	Internet Gateway
URBes	Urban Routing Backbone Simulator
BS	Base Station
MinHop	Minimum hop
AoI	Area of Interest
AC	AirCache
FI	FrameInformation
RR-ALOHA	Reliable Reservation-ALOHA

BCH	Basic Channel
FIT	Frame Information Time
NS	Network Simulator

The highest forms of understanding we can achieve are laughter and human compassion.

Richard P. Feynman

1

Introduction

The technological evolution is palpable now more then ever. Only a decade ago, digital information was mainly confined to electrical wires, traveling bits around the Internet. Then we had the advent of the wireless technology and shortly after researchers proposed the Internet of Things, envisioning connectivity among computers and different objects (e.g., home appliances). This dissertation aims to go beyond and produce technology that will enable the Internet of Everything (IoE), which will allow sharing any context-related digital information, including data from RFID tag readers, IP cameras, user profile updates, online reviews, augmented reality systems etc. Indeed, with interconnected devices that have recently surpassed in number the people on Earth, our lives are currently immersed in a digital fountain with information being produced everywhere around us. A user with a handheld or wearable device (e.g., smartphone or even Google Glass) equipped with sensing and communication capabilities can now be both producer and consumer of information and services.

Content is increasingly generated in a participatory fashion by the users themselves, following the user-generated content (UGC) model best exemplified by Web 2.0 services such as blogs, YouTube. Clearly, a user with a smartphone in hand that includes a camera, microphone, speaker, and other sensing capabilities has become a producer as well as a consumer of information and services [19]. The more powerful the user devices are, the likelier it is that they will generate and share content, leading to the presence of distributed information sources and the diminishing role of centralized servers. This is undoubtedly an opportunity which will lead to an increase demand for mobile services enabling content sharing and distribution among *mobile disconnected devices* which are now typically offered based on infrastructure supported communication platforms [94, 1].

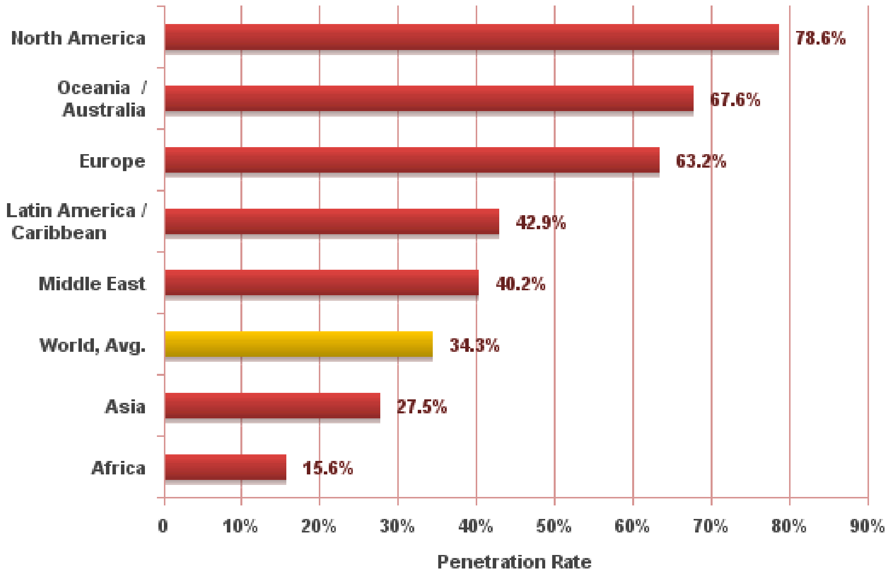


Figure 1.1: Source: Internet World Stats [41]. Penetration Rates based on a world population of 7B and 2.4B estimated Internet users until June 30, 2012.

A major advantage of infrastructure supported communication is that of enabling people to access their services regardless of their location. This is however partly true. If you live in a poor area in a developing country, or even in a remote area of one of the leading Information and Communication Technology (ICT) nations in the world, chances are that you will not have access to this plethora of network services, and risk feeling like a second-class citizen. Recent statistics, illustrated in Fig. 1.1, show a large skew in the distribution of Internet penetration towards the Western world and certain parts of Asia. Thus, the digital divide between people in different parts of the world, and between different groups within communities still exists. In addition, this mobile revolution is posing major stress on current infrastructure due an ever increasing data traffic. There is need now more than ever for new and alternative networking techniques able either to provide/extend connectivity where it is not present [66, 71], or that could be employed as an alternative to help alleviate network criticalities (e.g., offloading [29, 87]).

Yet, due to the widespread availability of mobile appliances equipped with a wireless interface, mobile applications could leverage other forms of connectivity which do not necessarily require pervasive infrastructure presence. Mobile Ad hoc NETworks (MANETs [21]) have emerged as viable networking paradigm, where mobile nodes can dynamically self-organize in a wireless overlay network offering application-specific support for communication. Due to their distributed nature, applications in this environment typically employ a peer-to-peer (P2P) communication model rather than client-server and potentially every node could act as a router, relaying messages toward destination(s). In these settings, mobile users have to rely on node and service discovery and delay-tolerant applications that are able to cope with network disruptions due to mobility [28, 68].

From these bases, it seems natural to employ ad hoc networking as an enabler, providing communication support to the disconnected mobile content-centric world. In this scenario, interactions among users could take place without the strict requirement of infrastructure mediation. This can be achieved by exploiting contact opportunities between pairwise devices enabling data transfer to take place. Due to the distributed and dynamic nature of this system, content producers and consumers might never be connected at the same time in the same network. These peculiarities taken all together demand for robust data gathering and dissemination techniques able to cope with mobility and the unpredictability of contact opportunities. To this end, into our help comes the OPPortunistic NETworking (OppNet [68]) paradigm, considered as a natural evolution of legacy MANETs, enabling end-to-end communication even in situations when nodes are temporally and spatially decoupled. In contrast with the MANET paradigm, where mobility is seen as a negative factor disrupting the network processes, OppNet leverage on node(s) mobility to create additional means for communication. This is achieved by exploiting node(s) unplanned contact opportunities in order to forward messages until they reach some destination(s).

Figure 1.2: Message exchange between node A and B exploiting an urban-wide mobile backbone.

Indeed, the OppNet paradigm is considered as a special case of terrestrial Delay/Disruption Tolerant Networks (DTNs [14]), employing an asynchronous *store-and-forward* communication model whereby nodes store the message locally and forward them opportunistically during contact opportunities. Figure 1.2 brings an exemplifying scenario for this networking paradigm. In the depicted scenario, node mobility is exploited in order to deliver a message from source toward a destination. The path from source to destination is dynamically built; each node makes the next-hop forwarding decision on the fly, bringing the message hopefully near to the destination.

Throughout this dissertation we propose and study some innovative scenarios and networking techniques aimed at supporting them. Before delving into technicalities, we spend few words at depicting an envisioned opportunistic, urban-wide service delivery scenario.

Opportunistic Urban Environment

The urban environment represents a dense populated area, comprised by a multitude of actors, being them either human or non human, where the majority is equipped with a wireless interface. Every day encounters occur with us not knowing about it: moving in restricted physical spaces like university campuses, coffee shops, urban transportation means or everyday walking activities etc., give rise to a huge number of contact opportunities. The all-pervasive and ubiquitous nature of mobile devices equipped with a wireless interface, and the everyday growing capabilities of the latter constitute an enormous unexploited resource, which we recently have began to recognize as such.

The huge number of actors in play, do provide a critical mass for opportunistic solutions which could enable sharing and distribution without strict infrastructure reliance. In this context, it is often the case where the data and the processes acting upon them are of local *scopus*. Answers to a query about a nearby object, an information source, a multimedia content, a process, a person, an experience, an ability, etc. could be answered locally. A local published content might be consumed (consulted) just by being in proximity with it. The data once checked might be further disseminated into this local network of information exploiting the mobility of the autonomous actors involved.

This urban-wide data network grows and shrinks meeting the peoples demand and vanishes whenever it is not required anymore (e.g., not alimented by the people). As opposed to multicasting techniques on infrastructure networks (e.g., Internet), here, the set of recipients is unknown; its cardinality is unpredictable and changes dynamically over time as a result of mobility and temporary disconnections. The data hence floats back and forth, from user to user, until it is not required anymore. As of current practice we rely on infrastructure to provide us the answer. However, data in these settings are inherently local and have temporal validity. This is in contrast with the current infrastructure data model where the data is never expired and available globally. In addition, by accessing data that matters locally alleviates privacy concerns related to the user declaring himself on the infrastructure users location and/or identity is kept private and not revealed. The data is created and caused locally, alimented by the users and not owned by a third party.

Scenarios Under Scrutiny

After this depicted frame, we now introduce the studied scenarios and their rationale. Following, we argue how they could be employed in synergy with one another to deliver a potentially urban-wide opportunistic service platform enabling message exchange amongst mobile users without strict infrastructure reliance.

S.1 *Mobile-to-mobile Content Sharing*: in the UGC era content is increasingly being generated by the users themselves. Also, it is often the case that content has local significance and could be consumed locally without reliance on infrastructure communication platforms. This objective aims is at devising an opportunistic solution enabling content exchange when users are in proximity with each other;

S.2 *Urban-wide Service Delivery*: this scenario aims at identifying a suitable autonomous actor in the urban environment that could aid in scaling and boosting the opportunistic network toward an urban-wide, provider-less communication platform;

S.3 *Floating Data*: this scenario introduces the concept of *floating data*. The rationale is that often the data has local validity, binded to a geographical area or explicitly imposed by a time attribute expressed by the metadata. The data once produced could be consumed when in proximity, alleviating the need to publish and make them available through infrastructure supported platforms.

A detailed analysis of these scenarios is provided later on this dissertation (refer to Chap. 3 to 5). Through the rest of this chapter we discuss the challenges that the OppNet paradigm represents, along with the strategy taken toward their resolution. At first, in Sec 1.1 we provide a panoramic view on the paradigm under scrutiny, enumerating the characteristics differentiating it from similar technologies in use. Following, in Sec. 1.2 we discuss the challenges that arise, providing some insights on how we address them. Next, in Sec. 1.3 we state the contributions of this work. Concluding, in Sec. 1.4 the overall organization of this dissertation is presented.

1.1 Fundamentals

Before delving into the section, we spend few words in evidencing the dual nature of the gathering and dissemination processes. A gathering process presumes that data are being disseminated into the network and vice versa, a dissemination process presumes that there is a destination(s) gathering these data. The way these processes are orchestrated are context specific, this depending on the context and deployment scope. From now on we will simply use dissemination to denote both the processes.

The notion of opportunism in wireless networks extends at different layers of the protocol stack. Spatial- and temporal-diversity are common layer-2 approaches providing communication guarantees in multiuser wireless environments [98]. The notion of *route opportunism* identifies level-3 approaches exploiting the broadcast nature of the wireless medium to advance messages into the network. This is in contraposition with classical multihop wireless networking where the packet is shielded from other nodes, rather than the intended next-hop recipient determined at the sender side [9].

In essence, opportunistic wireless networks are a kind of challenged wireless mobile network, where prolonged disconnections, unpredictable and unstable topologies, and partitions can frequently occur. This has resulted in a paradigm shift for the design of network services where intermittent connectivity is a rule rather than an exception. In addition to legacy MANET exploiting channel diversity to cope with the time variation of the wireless medium, OppNets further exploits multiuser diversity and route opportunism to move data between endpoints [68].

In their seminal work, Grossglauser and Tse in [37] provided a first theoretical result which spurred interest in the field. In essence, they proved that node mobility seen as a type of multiuser diversity increases per-user throughput and network capacity as whole, counteracting the well-known broadcast storm effect in wireless networks [88]. Further, Fall in [28] launched a concrete architectural design aimed at interconnecting these challenged networks with the interplanetary infrastructure. At the core of his proposal was a delay-tolerant, *store-and-forward* communication model, first proposed in the realm of interplanetary Internet project, providing communication primitives between heterogeneous networks.

OppNets are identified as wireless ad hoc networks comprised of mobile nodes enabled to communicate through a wireless interface. They share key common characteristic with MANETs, which derive from the common nature of their distributed components, dictating that they are faced with the common fundamental challenge, that is, to provide connectivity in a decentralized, dynamic environment. However, OppNets are considered as an evolution of legacy MANETs, omitting their key assumption that a full path between two endpoints exists at any given time. Communication in this setting is achieved by leveraging on node mobility to physically carry the message toward the intended destination. In these settings, node and service discovery are the basic application building blocks, enabling nodes to discover each other before establishing a communication link. How discovery is performed, is a major design issue from both a networking performance and node perspectives [72].

Supporting end-to-end communication, delay tolerant techniques, in terms of involving asynchronous, store-and-forward, message exchange (bundles), have been largely investigated (e.g., [28, 89]). OppNets can be considered as a special case of DTNs, although, differently from the original outer space DTNs, where an estimation of delivery delays can be computed in advance due to some degree of determinism of system variables, OppNets are characterized by unpredictable delays due to the unpredictability of the next forwarding step. In essence, OppNets are characterized by:

- heterogeneous nodes in terms of capabilities and communication means;
- heterogeneous contact rates;
- variable/high mobility;
- limited or no network information;

- endpoints might be spatially and temporally decoupled;
- destination might not be known beforehand.

Opportunistic solutions have been employed in providing service access in scenarios where infrastructure is lacking or not feasible [69]. In natural disaster or environmental monitoring scenarios, the deployment of an infrastructure-based network could result time consuming and not cost effective; e.g., mobile gateways once deployed can self-organize in a wireless integrated network capable of sustaining service continuity [12]. In tactical scenarios OppNets can hypothetically provide a highly decentralized, robust network resilient to attacks enabling participating entities to coordinate toward a common objective. Other applications can be found in scenarios where service access is attributed to costs or traffic offloading from infrastructure to the ad hoc backbone network [54]. In this case, ad hoc networking in terms of involving multi-hop delay tolerant networking has proven as a viable alternative at providing service intermittent connectivity for elastic non real-time applications [56].

Despite major research efforts, current field deployments are limited to few niche scenarios which were started as proof-of-concepts field tests. The reasons for this are diverse and range from technological constraints to more pertinent issues such as node mobility and its impact on network management and performance. Indeed, the power-constrained nature of mobile devices and connection-oriented, battery-consuming nature of some wireless technologies pose a prohibitive cost on the deployment of such solutions. A lot of research has been conducted on human dynamics and resulting mobility patterns, however, the circumstances and data gathering processes are not sufficient for general-purpose solutions.

In order to design useful applications, it is vital to have a good understanding of the target environment and its users. Different types of user behavior can result in dramatically different network conditions and will have a huge impact on whether or not a particular application will be of interest to the user base. Another important aspect, but unfortunately harder to measure as it demands the solution to be already deployed, is to understand traffic and usage models of the network. A definite solution to all these problems is difficult, if not impossible to provide. In our work we focus our attention toward practical applicative scenarios for use in real contexts. We rely on software simulation to assess our proposals as this is a common approach used in the domain. Whenever possible we provide a complementary assessment of the solutions by employing real-field measured data. In the following section we introduce the problem statement, discussing about the criticalities that emerge from the depicted scenarios along with the strategy adopted toward their resolution.

1.2 Problem Statement

In this section we state the problems that need to be addressed in order to enable the aforementioned applicative scenarios. Along with the issues we provide some insights as how they could be tackled. The approach and networking techniques adopted to address the issues are discussed through Chap. 4.

S.1-A Devising a pull-style, content sharing service in a dynamic, disconnected network(s) of information presents several challenges. In this scenario content producers might be temporally and spatially decoupled from consumers, hence the chances of finding/retrieving the content are low when compared to infrastructure supported solutions. This fact is exacerbated when considering the *resource-constrained* nature of mobile devices. How content *search* and *retrieval* are to orchestrated in this context, paying attention to resource-constrained nature of mobile devices, is what we aim to investigate.

In this context, mobile users could exploit pairwise contact opportunities among other encountered nodes to consult data available in its vicinity. Moreover, the system could give users the possibility to reach data beyond their connected neighborhood, hence, departing from synchronous pairwise to multihop communication capabilities. However, we argue that even in these settings nodes have a constrained data horizon - limited to the context(s) they visit. The system should provide the means to explore data content available elsewhere, outside their reach. We anticipate that in achieving this goal we depart from a pure synchronous communication model to an hybrid communication model augmented with a *store-and-forward* message exchange similar in spirit with that pioneered by the DTN (refer to Sec. 4.1.3.2).

S.1-B The content-centric nature of the scenario considered above lends itself to the NDN design philosophy. In the realm of mobile NDN, several proposals have been advanced addressing data dissemination in networks bereft of infrastructure like vehicular networks ([91, 92]). In these scenarios names are bounded to geographical locations and forms of epidemic forwarding are employed pushing the data into the network. However, in our network model the data does not have any physical boundaries and consumer are producers might be temporally and spatially decoupled. Means of an asynchronous message exchange extending a nodes reach area to other connected local networks are necessary. In addition, epidemic message dissemination of requests and data might prove resource consuming and particular attention toward this direction is needed.

Named-data communication *tailored* to the characteristics of the scenario *S.1* is what we aim at pursuing. We anticipate that we depart from a name-based data solution to hybrid communication scheme, whereby node-based communication is employed when the *delegation-forwarding* mechanism as described in Sec. 4.1.3.2 is triggered.

S.2 The unpredictable nature of human behavior and the resource-constrained nature of mobile hand-held devices is not sufficient in providing an urban-wide opportunistic communication platform. To this end, we set on a trial to investigate a carrier-based approach which could be employed in synergy within the previous context (refer to Sec. 6). We exploit the Public Transportation System (PTS) aimed at supporting elastic, non real-time applications (e.g., push/pull news or advertisements). In this context, the problem that arises is the *deployment architecture* for this kind of scenario *per se*: the entities involved in communication and the orchestration of data dissemination process. Moreover, from related works in this context there is *few comparison analysis* between the forwarding schemes that have been proposed.

Our proposal employs the PTS as a routing backbone for user issued requests and responses. We provide some insights on routing proposals performance under different scenarios employing realistic mobility models taken from real urban-wide topological data (Sec. 4.2).

S.3 In this envisioned scenario the data has local validity, supplied and maintained by the users themselves. Once a content is published locally (e.g., by a nearby mobile node), means of guaranteeing the chances of *data survivability* and *accessibility* need to be put in place. How to provide such guarantees in a decentralized, dynamic network is what we set on a trial to investigate.

A possible solution would be that of disseminating the data to all potential nodes in the area. While this solution could augment the chances of data survivability and lower the costs of accessing the data (e.g., in terms of the number of hops required to retrieve it) in the binded geographical area it has the negative effect of being too resource-consuming. At the same time it does not provide any guarantee that data could survive due to mobility factor. We anticipate that out solution adopts node cooperation to achieve this objective (refer to Sec. 4.3).

Altogether these scenarios are complementary and could be employed in synergy with one another to deliver a potentially urban-wide opportunistic service delivery platform enabling message exchange amongst mobile users without strict infrastructure reliance. The use-cases are vast and include a multitude of system orchestrations: e.g., a user searching or wanting to divulge a data content, requesting it to be delivered in a certain place where data is to be anchored until certain conditions are met. This scenario could employ both human and carrier autonomous actors in synergy, exploiting them as an urban routing backbone to deliver data to a targeted location. Another example is to employ the data anchoring technique to help alleviate criticalities emerging from the PTS scenario, introducing resilience counteracting unpredictable changes in the PTS timetable (refer to Sec. 3.2.2).

Throughout this dissertation we study the specifics of each depicted scenario and relative networking techniques. We do not provide any study of their synergistic use leaving it as a future work. Our contribution is to analyze these problems, and to propose some possible directions towards their resolution. In the next section we enumerate the contributions of this work put in context of the chosen scenarios.

1.3 Contributions

We now introduce the overall contributions of our work along with a brief description of the applicative scenarios providing some reason on why pursuing them. Whenever possible we also provide a synthetic comparison to works in the OppNet domain. At this point we only state their innovative aspect, providing a brief description; a detailed study of each scenario is provided later on in separate sections of this dissertation (refer to Chap. 3 and 4). In specifics, the overall contributions of this work are as follows:

S.1-A Propose a content-sharing solution for the mobile disconnected world exploiting the history of encounters guiding message routing/forwarding: A killer application demanding a pull-based service model as a networking primitive is the peer-to-peer (P2P) content-sharing scenario. We port a similar solution into the mobile disconnected world, enabling message exchange when users are in proximity with each other. While our aim is at devising an information-centric approach similar to the Publish/Subscribe proposals found in the OppNet literature, these approaches differ in several ways as detailed in [16]. The most relevant difference worth pointing out stands in the way information flows. Indeed, in the publish/subscribe paradigm the information-flow is initiated by the producer while in the broader, content-based networking it is consumer initiated. That is, the system requires search capabilities and how data gathering and dissemination is orchestrated in a limited information opportunistic communication platform is what we address (refer to Sec. 4.1);

S.1-B Propose a Named Data Networking (NDN [42]) solution of the former approach enabling sharing and distribution using named-data: We propose a named-data solution for the former proposal. In essence, the solution devised in point 1 is revisited and a named-data approach is proposed. In addition to previous study in the realm of NDN, we propose a novel Interest-forwarding strategy enabling message exchange in scenarios where mobile users are not connected with each other at the same time in the same network (refer to Sec. 4.1.4);

S.2 Propose a concrete urban-wide distribution architecture employing the public transportation system as a routing backbone: With the aim of enhancing the performance of the former scenario we depart from an infrastructure-less approach and investigate the PTS as a service delivery platform. The network is comprised of PTS carriers and mobile users issuing requests for content available elsewhere (e.g., the Internet). In this context, we propose a practical deployment architecture leveraging the PTS carriers as a routing backbone and investigate the performance of a state-of-the-art routing protocols under different realistic topological and traffic data (refer to Sec. 4.2);

S.3 Propose a protocol built upon existing solutions supporting an innovative scenario, that of the *floating data* concept: In some scenarios data is of local relevance, confined to the context they are produced and could be consumed just by being in proximity with it. One could imagine the data moving back and forth, from user to user, confined in the interest area, hence the name floating data. As an example, any instant advertising of goods, attractions, events and/or news can be consumed locally, disseminated and kept alive in the area where the information is of relevance creating a digital footprint surrounding real objects. In this envisioned scenario users interact locally with the information without the need for a central authority to provide it. The data has geographical and temporal validity as opposed to the current approach making them available globally and stay on forever. We study this envisioned scenario and propose a protocol able to provide the desired service. To this end, we rely on existing protocols and built the necessary logic able to guarantee data survivability in an area of interest (refer to Sec. 4.3).

1.4 Overview

A note for the reader: parts of the written chapters are copied from published material of the candidate, enumerated in the Appendix A of this dissertation.

This dissertation is organized as follows:

- **Chapter 2** provides an extensive background regarding the OppNet communication paradigm, starting from related paradigms and how they evolved into the concept of OppNets (Sec. 2.1). Following, in Sec. 2.2, a taxonomy of opportunistic communication networks is provided based on the autonomous actors involved in communication. Concluding, in Sec. 2.3 we provide a brief introduction to the Information-Centric Networking (ICN) paradigm proposed as a paradigm shift from the current host-centric approach of the Internet.
- **Chapter 3**, surveys the state-of-the-art proposals contextualized to our studied applicative domains, evidencing their limitations when compared to our chosen goals.
- **Chapter 4**, discusses the work undertaken during this activity, presenting a more detailed picture of the case studies, and the networking techniques devised at supporting them.
- **Chapter 5** presents the methodology taken to evaluate our proposals along with the assessment outcome.
- **Chapter 6**, draws our conclusions and discusses future research directions deemed worth pursuing.

2

Background

This chapter provide the necessary background information regarding the technologies under scrutiny. To this end, we start by providing a general picture regarding the challenged networks, stating their peculiarities and proposed architectural model enabling the Internet of Everything. Following in Sec 2.2 a taxonomy of opportunistic networks is provided along with representative routing/forwarding techniques in each category. Next, in Sec. 2.3 we provide some basics regarding the information-centric approaches advanced as a replacement of the current host-centric Internet approach, paying particular emphasis on the Named Data Networking (NDN [42]) approach. Concluding, we survey a reference architectural proposal for content-centric mobile networks.

2.1 A Delay Tolerant Architecture for the Internet of Everything

As soon as the Internet was developed, there was a desire to connect more *things* to it. From the handful of computers making Internet in the beginnings, it now connects anywhere tens of billions of devices. At the same time, this evolution has given spur to heterogeneous networking environments, not compatible with one another which do not interact well with current practices of the Internet. This *challenged networks* have underlying characteristics that differ from the Internet and demand specialized networking techniques able to cope with environment specific requirements. In the following section we provide a concise survey on these emerging *challenged networks*, outlining their characteristics and challenges. Concluding, rather then focusing in domain specific protocols we provide a sketch of the DTN architecture proposed to enable interoperability among internets, providing a common ground for the Internet of Everything.

2.1.1 Challenged Networks

The existing TCP/IP based Internet communication model provides end-to-end inter-process communication using a concatenation of potentially dissimilar link-layer technologies. The standardization of the IP protocol and its mapping into network-specific link-layer data frames at each router supports interoperability using a packet-switched model of service. Although often not explicitly stated, a number of key assumptions are made regarding the overall performance characteristics of the underlying links in order to achieve this service: an end-to-end path exists between a data source and its peer(s), the maximum round-trip time between any node pairs in the network is not excessive, and the end-to-end packet drop probability is small. Unfortunately, a class of challenged networks,

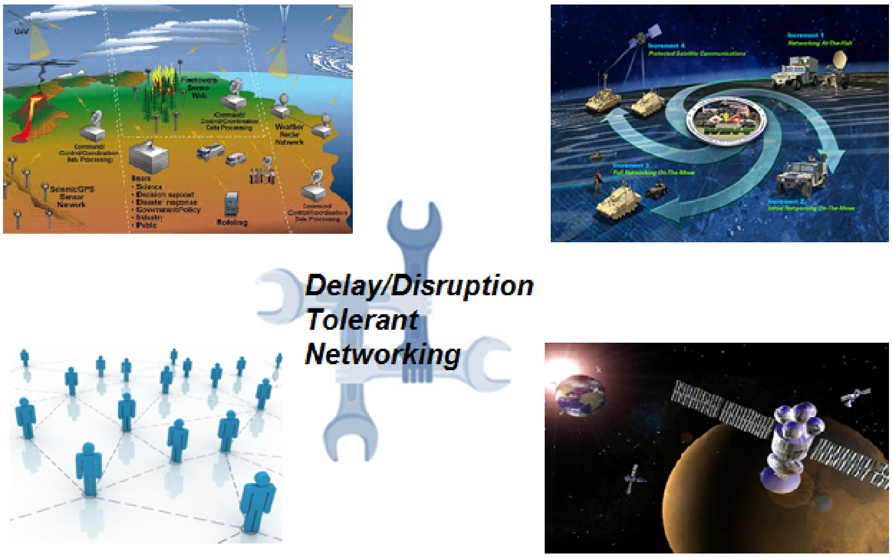


Figure 2.1: An all encompassing Delay/Disruption Tolerant Networking approach.

which may violate one or more of the assumptions, are becoming important and may not be well served by the current end-to-end TCP/IP model. Examples include:

- **Terrestrial Mobile Networks**: Some of these networks may become unexpectedly partitioned due to node mobility or changes in signal strength (e.g. interference), while others may be partitioned in a periodic, predictable manner. For example, a commuter bus could act as a store and forward message switch with only limited-range wireless communication capability. As it travels from place to place, it provides a form of message switching service to its nearby clients to communicate with distant parties it will visit in the future.

- **Military Ad-Hoc Networks**: These systems may operate in hostile environments where mobility, environmental factors, or intentional jamming may be cause for disconnection. In addition, data traffic on these networks may have to compete for bandwidth with other services at higher priority. As an example, data traffic may have to unexpectedly wait several seconds or more while high-priority voice traffic is carried on the same underlying links. Such systems also may also have especially strong infrastructure protection requirements.

- **Sensor/Actuator Networks**: These networks are frequently characterized by extremely limited end-node power, memory, and CPU capability. In addition, they are envisioned to exist at tremendous scale, with possibly thousands or millions of nodes per network. Communication within these networks is often scheduled to conserve power, and sets of nodes are frequently named (or addressed) only in aggregate. They typically employ 'proxy' nodes to translate Internet protocols to the sensor network native protocols.
- **Exotic Media Networks**: Exotic communication media includes near-Earth satellite communications, very long-distance radio or optical links (e.g. deep space communications with light propagation delays in the seconds or minutes), acoustic links in air or water, and some free-space optical communications. These systems may be subject to high latencies with predictable interruption (e.g. due to planetary dynamics or the passing of a scheduled ship), may suffer outage due to environmental conditions (e.g. weather), or may provide a predictably-available store-and-forward network service that is only occasionally available (e.g. low-earth orbiting satellites that pass by periodically each day).

Qualitatively, these challenged internetworks are characterized by latency, bandwidth limitations, error probability, node longevity, or path instability that are substantially worse than is typical of todayâs Internet. Given the large accumulated experience and number of systems compatible with the TCP/IP protocols, it is natural to apply the highly successful Internet architectural concepts to these new or unusual types of networks.

While such an application is conceivable, the effects of very significant link delay, non-existence of end-to-end routing paths, and lack of continuous power or large memory at end nodes present substantial operational and performance challenges to such an approach. Encompassing all these characteristics into tomorrows IoE are the proposal sketched in [28, 26] which we briefly survey in the following section.

2.1.2 A Delay/Disruption Tolerant Architecture

In an effort to adapt Internet to unusual environments, one class of approaches attempted to engineer problem links to appear more similar to the types of links for which TCP/IP was designed. In effect, these approaches, referred to as link-repair approaches, *fool* the Internet protocols into believing they are operating over a comparatively well-performing physical infrastructure.

Another common approach to deal with challenged networks has been to attach them to the edge of the Internet by means of a special proxy agent. This provides access to and from challenged networks from the Internet, but does not provide a general way to use such networks for data transit. Without supporting transit, the full capabilities of these networks may go unrealized.

As from the Internet experience, the most desirable framework for supporting challenged internets would be a network service and API providing a sort of least common denominator interface: non-interactive messaging. This system had to combine some overlay routing capability such as is present in peer-to-peer systems with the delay-tolerant and disconnection-tolerant properties of the electronic mail model. The implementation would occur at the application layer (in the form of a proxy) and such a system could conceivably provide a gateway function between radically dissimilar networks.

From this basis, the architecture proposed for interoperability between and among challenged networks was coined as the delay tolerant networking architecture (DTN), based on an abstraction of message switching, employing an asynchronous store-and-forward message exchange. Message aggregates are known as *bundles* and are adopted from [27]. The routers that handle them are called *bundle forwarders* or DTN gateways.

At its inception, the concepts behind the DTN architecture were primarily targeted at tolerating long delays and predictably-interrupted communications over long distances (i.e., in deep space). At this point in time, the work was an architecture for the Interplanetary Internet (IPN). By March 2003, when the first draft of the eventual Request For Comment 4838 (RFC [26]) was published, one of the authors had coined the term delay tolerant networking suggesting the intention to extend the IPN concept to other types of networks, specifically including terrestrial wireless networks. Terrestrial wireless networks also suffer disruptions and delay, and the DTN architectural emphasis grew from scheduled connectivity in the IPN case to include other types of networks and patterns of connectivity (e.g., opportunistic mobile ad-hoc networks with nodes that remain off for significant periods of time).

At this stage, DTN started as a network of regional networks but now the association between nodes and territorial regions is no more strict [28]. DTN achieve interoperability by accommodating long delay between and within networks and translating between network communications characteristics. Therefore it can accommodate the mobility and limited power evolving wireless communication devices. As an overlay architecture, DTN is intended to operate above the existing protocol stacks in various network architectures and provide a store-and-forward gateway function between them when a node physically touches two or more dissimilar networks. For example, within the Internet the overlay may operate over TCP/IP, and in delay- disconnection- tolerant sensor/actuator networks it may provide interconnection with some yet-to-be-standardized sensor transport protocol. Each of these networking environments have their own specialized protocol stacks and naming semantics developed for their particular application domain. Achieving interoperability between them is accomplished by special DTN gateways located at their interconnection points.

In the following we provide a survey of a particular class of challenged networks which are also subject of this dissertation: opportunistic networks or terrestrial-Delay/Disruption Tolerant Networks (tDTN). At the focus our discussion is the routing/forwarding issue. To this end, we provide a taxonomy along with some examples for each representative category.

2.2 Opportunistic Networks: A Taxonomy

While medium access and data transmission techniques can be addressed by means of existing solutions, routing/forwarding in such environments is the most compelling challenge [68]. The design of efficient routing strategies for opportunistic networks is generally a complicated task due to the absence of knowledge about the topological evolution of the network. Routing performance improves when more knowledge about the expected topology of the network can be exploited. Unfortunately, this kind of knowledge is not easily available, and a trade-off must be found between performance and knowledge.

A first classification is between algorithms designed for flat ad-hoc networks (*infrastructure-less*), and algorithms in which the ad-hoc networks exploit some form of infrastructure to opportunistically forward messages (*infrastructure-based*). Both categories adopt the same basic strategy whereby node mobility is exploited in order to transfer data. Data can be stored and carried by taking advantage of node mobility and then forwarded during opportunistic contacts. When this happens,

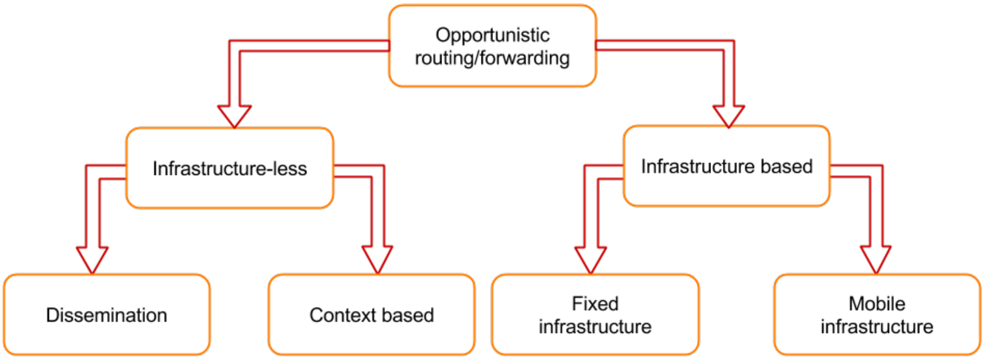


Figure 2.2: A taxonomy for OppNets.

entire chunks of messages (bundles) are transferred from one storage place to another along a path that is expected to eventually reach the destination. Indeed, there is no guarantee that a path toward destination will be actually found. Routing schemes often introduce redundancy, delivering the same message to different paths, this, in order to increase delivery rate and keep delays acceptable. In the following we provide evidence two emerging opportunistic network scenarios, which are necessary in order to provide a complete the picture of this paradigm.

Depending on the availability of infrastructure entities, the exploited context information, movement/encounter prediction possibilities, two different OppNets schemes arise. In specific, through the rest of this section, we outline and discuss the peculiarities exhibited by: *(i)* the Pocket Switched Networks (PSNs), an infrastructure-less approach where the network is composed solely by mobile users, and *(ii)* the carrier-based approach where some form of mobile and/or static infrastructure is present.

2.2.1 An Infrastructure-less Approach

Represents an opportunistic networking approach whereby planned/unplanned human contact opportunities carrying wireless-equipped appliances (e.g., smartphones) are exploited in forwarding messages between intended endpoints. It follows an opportunistic infrastructure-less approach and is characterized by:

- A high degree of heterogeneity in terms of capabilities and contact rates.
- Unpredictable behavior (e.g., movement dynamics) of actors operating such devices.

This approach represents a concrete opportunity; the frequency and potential of opportunistic contacts are mind boggling. Given the plethora of wired and wireless communication technologies, along with device capabilities, opportunistic contacts among pair-wise devices are the norm rather than a rarity. The necessary infrastructure for opportunistic networking is thus all-pervasive, providing a concrete communication support in our urban areas [20]. The concept of people-centric sensing starts from basis. Based on these sensory readings gathered from a multitude of devices, it is possible to infer knowledge in order to create smart applications that benefit the community [15]. Applications of this paradigm can be found in the cooperative environmental monitoring domain (public sensing), where mobile sensor-equipped smartphone users can act as data producers - sensing their surroundings, also as data collectors - collecting data from nearby special purpose deployed sensors and disseminating this data through opportunistic means toward infrastructure endpoints.

Despite this colorful picture of future PSNs, solutions of this kind are presently limited to few social applications like presence, messaging, synchronization [56] etc. Routing in these environments is particularly challenged due to the unpredictability of human mobility. In general, topological information is in fact quite unstable and not very reliable. It needs to be complemented with context and social information to obtain better understanding of the evolution of the contacts among users, and therefore of near-future opportunities for communication [79]. It is thus clear that collecting and managing context information is of paramount importance in this environment.

Routing schemes in such scenarios are divided into context-oblivious and context-based algorithms. Literature material discussing that regards this issue, flood-based and dissemination-based techniques are used interchangeably. Dissemination-based algorithms are essentially forms of controlled flooding, and differentiate themselves by the policy used to limit flooding. The devised strategies diffuse the message in the entire network, at each opportunistic encounter nodes are infected with the message [89]. The heuristic behind this policy is that, since there is no knowledge of a possible path toward the destination, nor an appropriate next-hop node, a message should be sent everywhere, and eventually it will reach the destination. However, this comes at an additional cost in terms of bandwidth, energy and often large induce redundancy.

Context-based approaches, in contrast with the dissemination-based schemes, leverage some form of context, gathered knowledge of network nodes in order to identify the best next hop at each forwarding step. Indeed, nodes in this approach maintain a local state, history of past encounters, and the next-hop routing/forwarding decision is made based on some utility metric [68]. In this way system redundancy is limited at the cost of a lower delivery probability and higher delivery delays. In this context, yet another crucial problem is the energy preservation issue. Indeed, when exploiting resources of human carried mobile devices, particular attention should be paid at the energy resource consumption. This is a crucial problem for the wireless world in general, which involves power management techniques, including network adapters that can trigger power resume of the device while offloading certain network activity, and application layer protocols that reduce power consumption.

Redundancy could indeed increase the chances of message delivery toward the intended destination, but at the cost of a higher number of wireless mobile transmissions, incurring a higher energy overhead. Therefore, when designing data gathering and dissemination schemes for the infrastructure-less pocket switched networks a tradeoff between energy and chances of message delivery is required. In the following section we provide some representative examples of data dissemination strategies evidencing their *modus operandi*. Concluding, we provide a discussion, arguing about the inherent limitations of these proposals.

2.2.1.1 Data Dissemination in Pocket Switched OppNets

An important result in forwarding algorithms for human comprised OppNets was given by Chaintreau *et al.* in [18]. The authors, analyzed different human mobility traces, concluding that the inter-contact times are power-law distributed for values up to one day. From this bases, they proved that a special class of context-oblivious routing algorithms, algorithms that do not exploit any kind of information from the network only that of message destination, have an infinite expected average delay for message delivery. That is, no guarantee can be given *a priori* for message delivery. This results have since revisited but still there is no definite or conclusive answer to the problem [67].

Simulation studies show that context-aware approaches can help alleviate known issues deriving from context-oblivious approaches. Indeed, when no other more traditional network properties (topology, connectivity, etc.) can be exploited to find paths between the source to the destination of a message, being able to predict the future behavior of the users of the network by means of past context information can be the only way to deliver messages while at the same time posing less stress on network and device resources.

Knowledge about the network topology and connectivity can help discriminate the best next hop toward a destination. In scenarios where such an information is not available or complex to achieve (e.g., high mobility scenarios), flooding might be the only means for communication. Such schemes, during our survey are also referred to as epidemic or context-oblivious schemes. The basic concept of epidemic dissemination is to flood messages, like the virus spreading in an epidemic. That is, a node copies its message to all the nodes that come in contact with it, provided the recipient node does not have a copy of it already.

Vahdat and Becker are perhaps the earliest proponent of such a scheme [89]; probably they were inspired by the algorithm proposed by Demers *et al.* [23]. To identify if the node has already seen a message, each node maintains a summary vector. This is an index of the messages that it has already seen. When two nodes meet, this summary vector is exchanged. This enables the nodes to identify the new messages and request for them. In order to control the resource utilization, the authors propose the use of a hop counter and limit the hop of each message.

Undoubtedly, flooding the network with messages will consume network resources like bandwidth, buffer, node energy etc. As demonstrated by Tseng *et al.* [88], this can seriously degrade the performance, if the resources are scarce. Hence there is a need to control flooding. To this end, several controlled flooding techniques have been devised which aim at limiting the number of copies in the network. Spyropoulos *et al.* [82] proposed several single copy schemes where the simplest is the case where the source directly delivers the message to the destination.

Differently from the epidemic and controlled flooding schemes, where the former provide only the means for controlling packet dissemination, history-based (partial context-aware) schemes are used instead as the means to guide the dissemination process. Indeed, the first approaches of context-awareness are the history-based schemes where nodes of the network utilize the history of past encounters, to make a more informed routing decision. Intuitively, a node that has encountered the destination many times is likely to encounter it again in the future. This is the principle standing at the core of history based routing protocols.

Lindgren *et al.* proposed *PRObabilistic Protocol using History of Encounters and Transitivity* (PROPHET [55]), a probabilistic routing protocol. This protocol uses the history of past encounters, to compute a local delivery confidence of the nodes. This probability indicates how likely it is that this node will be able to deliver a message to that destination. When two nodes meet, they exchange summary vectors, and also a delivery predictability vector containing the delivery predictability information for destinations known by the nodes. This additional information is used to compute the transitive predictability of encountering these other unknown destinations; this, under the assumption the frequently encountered property is transitive. Also the authors propose an aging function for predicted entries applied after K unit times. Here several dissemination strategies can be adopted, however, the authors show by simulations, that the PROPHET version disseminating messages only to nodes with higher delivery predictability achieves a higher delivery ratio than epidemic approaches, while incurring a much lower communication overhead. This of course considering a realistic scenario where mobile nodes have limited resources in terms of storage.

The intuition is that, the more information is gathered about past and present encounters, the more accurate is the forwarding process. For many years, research studies assumed traffic and node movement to be random. In reality, however, mobile nodes in pocket switched networks are operated by people, whose behaviors are better described by social models. This idea has opened up new possibilities in the opportunistic networks, since the knowledge that behavior patterns exist allows better decisions to be made.

To the best of our knowledge the first work that incentivized and proved to some degree the benefits of the social approach was first conducted by Hui *et al.* in [39], which introduced the *LABEL* scheme. In this scenario, each node is assumed to have a label that informs other nodes of its affiliation; next-hop nodes are selected if they belong to the same affiliation (same label) as the destination. They showed by simulation means that LABEL significantly improves forwarding efficiency over oblivious dissemination using their one dataset. This was perhaps the beginning of social based dissemination strategies in PSN, but without a concise concept of what a label was and lack of mechanisms to move messages away from the source when the destinations are socially far away.

Hui *et al.* [40] proposed the *Bubble* algorithm, which is also based on the two aspects that emerge in social structures: community and centrality. Bubble, combines the knowledge of community structure with the knowledge of node centrality to make forwarding decisions. Intuitively each node has a community in which it resides, and central nodes which have greater knowledge about the network are involved in the routing process. By far, the most important part of this work is the proposal of a decentralized community and centrality detection algorithm which the authors show to perform decently with respect to offline centralized algorithms.

ProfileCast in [45] proposes a context-centric forwarding algorithm leveraging the behavioral patterns of mobile network users for delivering messages to a sub-group of users as derived by their visiting patterns. A content generated by a node is addressed to, is of interest for, nodes used to visit the same locations as the source. The proposed forwarding strategy corresponds to a scoped-flooding in the profile space: nodes keep forwarding the message to those who are similar to them under the considered profile, but ignore those who are dissimilar. The presented network model is comprised of mobile handheld devices and content distribution toward encounters is based on a threshold-based similarity metric between profiles.

SocialCast [22], instead exploits the social dimension between mobile nodes to guide the forwarding process. The rationale is that users with the same interests have the attitude to meet with each other more often than with other users. Extending further the notion of sociality and bounded content is ContentPlace [10] which assumes that users belong to social communities and that communities are bound to physical places. Although it is accepted that communities may include different interests, it is assumed that there is a predominant interest inside a community and this drives communications that happen to have a community-based granularity. Both systems propose a network model comprised of mobile devices where data is generated by the users themselves following the UGC model.

2.2.1.2 Discussion

A first observation regarding the surveyed literature work, irrespective of the adopted forwarding scheme, is that communication is sender-initiated and the *a priori* assumption that destination(s) are known beforehand. Depending on the replication scheme employed by the forwarding algorithm the proposals could be employed to deliver an anycast service model. Toward this direction a lot of research effort has been invested in devising multicasting solutions where the message targets groups of nodes having some common identity rather then a single individual.

In this scenario, the senders (publishers) send messages without any explicit destination address, but with some structured content visible to the network, while receivers declare a predicate (a kind of query) that, when applied to the structured content of a message, tells whether the message matches the receiver's interests. The forwarding algorithms exploit the metadata available on the message to route it toward the desired destination (e.g., a bounded physical location). The network then transmits each message to any and all receivers interested in its content, that is, to all receivers whose declared predicate is matched by the content of the message.

However, we argue that in the opportunistic networking environment the set of recipients is unknown; its cardinality is unpredictable and changes dynamically over time as a result of mobility and node behavior. The assumptions made in proposals like those discussed above, following a Publish/Subscribe paradigm, greatly simplify the original problem because they inherently confine message delivery within a specific location and/or community. Unfortunately, human interests are not only bound to specific locations or closely assigned to a given community. In [38] the authors observe that the correlation among all plans is not immediate, while in [61] it is shown that this correlation varies with changing scenarios. Another practical matter concerns the time required to gather the information which is a crucial building block of the forwarding algorithms. The more time the system is in place, the more accurate the algorithms should get, hence, the implicit pact is an ever growing system accuracy.

Given the hype on this research stream, the lack of comparison analysis between different proposed strategies on common settings, we set on a trial to investigate innovative service delivery methods. Toward this goal, we focus our attention on innovative applicative scenarios which are covered through the rest of this dissertation.

2.2.2 A Carrier-based Approach

Routing with infrastructure in this context is used to denote an opportunistic networking scenario whereby special infrastructure entities are involved in the routing/forwarding decision making. The term infrastructure is used to denote the presence of special nodes being either static, e.g., access points deployed at specific areas involved in specific tasks, or mobile nodes which are more capable with respect to the other nodes commonly present in the ad hoc network in terms of storage capacity and energy.

Algorithms that exploit some form of infrastructure can be divided (depending on the type of infrastructure they rely on) into fixed infrastructure and mobile infrastructure. In both cases the infrastructure is composed by these special nodes located at specific geographical points, whereas nodes of a mobile infrastructure move around in the network following either predetermined known paths or completely random paths.

In fixed infrastructure networks a source node wishing to deliver a message generally keeps it until it comes within reach of a base station belonging to the infrastructure, then forwards the message directly to it. Base stations are gateways towards less challenged networks (e.g., access points providing Internet or local area network access). Variations of the protocol involving multi-hop routing between nodes are also used. The goal of the latter scheme is the same, with the main difference that node-to-node routing is involved in order to reach the base station.

While, in mobile infrastructure networks, also referred to as carrier-based approach, nodes of the infrastructure are mobile and participate in the data collection process. They move around in the network area, following either predetermined or arbitrary routes, and gather messages from the nodes they pass by. These special nodes are also referred to as carriers, mules or forwarders. They can be the only entities for message delivery, when only node-to-carrier communications are allowed, or they can simply help increasing connectivity in sparse networks and guaranteeing that also isolated nodes can be reached. In the latter case, delivery of messages is accomplished by both carriers and ordinary mobile nodes, and both node-to-node and node-to-carrier communication types are allowed [48, 81].

In the context of opportunistic networks leveraging a mobile infrastructure, the deployment of an opportunistic network on top of the public transportation system has drawn the attention of researchers as they have proven capable of providing a cost-free, infrastructure delay tolerant service access by means of opportunistic communication. This is achieved by leveraging the mobility and capabilities of PTS buses employing a carrier-to-carrier communication to reach infrastructure end-points. In this deployment scenario we depart from a pure opportunistic network to a hybrid one, whereby in addition to the mobile infrastructure there are also deployed access points providing access to the infrastructure. Indeed, Bus Switched Networks (BSNs [31]) inherently help mitigate the well-known criticalities of PSNs and at the same time take advantage of some particular behaviors:

(i) buses compared to other hand-held devices are powered nodes whose lifetime cannot be affected by routing operations;

(ii) their mobility is governed by a partially deterministic schedule, in comparison with their human counterpart;

(iii) PTS involves a relatively large number of buses and this ensures a pervasive coverage of the urban area.

These features taken all together, promise a packet delivery platform which may lead to the deployment of an urban-wide, infrastructure-free, and provider-less wireless network platform [69, 13, 97].

2.3 Information-Centric Networking

The current Internet architecture revolves around a host-based conversation model and fixed computers. This host-centric interaction model is being challenged by todays use of the network which focuses on information distribution and retrieval. The architectural approach of Information-

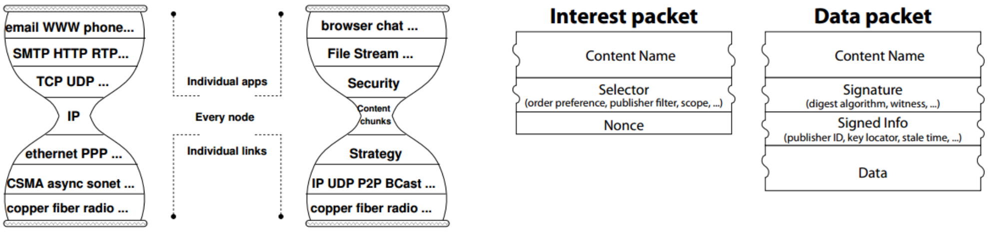


Figure 2.3: NDN architecture and packet format.

Centric Networking (ICN) provides an alternative to this communication model by focusing on the data, using the network as an intermediate storage. In order to support this, ICN combines a number of concepts such as naming, caching and the publish-subscribe paradigm [2]. Many content-centric architectures have been advanced, differentiating on how the above mentioned network functions are provided. In this section we provide some insights on an emerging approach namely the Named Data Networkig (NDN [42]). Concluding, we provide a brief survey of mobile content-centric proposals, those relying on solely mobile nodes in performing data gathering and dissemination processes.

2.3.1 Named Data Networking (NDN)

NDN is an emerging networking paradigm considered as a possible replacement for the current IP-based, host-centric Internet infrastructure. NDN promotes content as a first-class citizen and content is directly named. Indeed, while the Internet communication model is revolved on *where* the data stands, NDN is focused on *what* the user is interested in. Although NDN is known a clean-slate approach to the current Internet, it can be layered on top of everything including IP itself.

Jacobson *et al.* in [43] sketch the overall architecture of the NDN proposal, delineating its design principles and *modus operandi*. NDN packets, as opposed to IP packets, carry data names rather than source or destination addresses (Fig. 2.3). Consumers issue Interest packets, identifying data content they are interested in, without specifying where the data is located.

Figure 2.3 compares the IP and NDN protocol stacks. Most layers of the stack reflect bilateral agreements; e.g., a layer 2 framing protocol is an agreement between the two ends of a physical link and a layer 4 transport protocol is an agreement between some producer and consumer. The only layer that requires universal agreement is layer 3, the network layer. Much of IPs success is due to the simplicity of its network layer (the IP packet - the thin waist of the stack) and the weak demands it makes on layer 2, namely: stateless, unreliable, unordered, best-effort delivery. NDNs network layer is similar to IPâs and makes fewer demands on layer 2, giving it many of the same attractive properties.

NDN departs from IP in a number of critical ways. Two of these, *strategy* and *security*, are shown as new layers in its protocol stack. NDN can take maximum advantage of multiple simultaneous connectivities (e.g., ethernet, 3G, Bluetooth and the 802.11 family) due to its simpler relationship with layer 2. The strategy layer makes the fine-grained, dynamic optimization choices needed to best exploit multiple connectivities under changing conditions, while the security layer secures content itself, rather than the connections over which it travels, thereby avoiding many of the host-based vulnerabilities that plague IP networking.

Data naming is of paramount importance for NDN communications and Interest packet propagation toward the content producer is binded to the structure of the content name. To this end, NDN proposes a hierarchical naming structure composed of a number of components, similar to the semantic meaning of IP addresses.

An Interest can identify precisely what content is required but in most cases the full name of the data is not known so the consumer specifies it relative to something whose name is known. As in the IP Internet, NDN routing/forwarding employs a prefix-based longest match lookup strategy. The profound implications of this overall design make this proposal suitable to address mobility issues arising in the current Internet. Indeed, by decoupling the data from their location, caching of data along the path, and native multi-homing support at the strategy layer mitigate the problems arising from the current host-centric approach.

2.3.2 Mobile Content-centric Proposals

Prior work on efficient content dissemination in networks subject to severe disruption can be categorized into node-centric and information-centric networking, which differ on how naming and addressing are done. In address/host-centric approaches, network nodes establish routes proactively or on demand to the addresses of destinations where services or content reside, and a directory service provides the mapping between the names of services and content to the addresses where they are located. By contrast, in information-centric approaches, network nodes establish routes to content and services using their names directly. Given that no efficient solutions exist for directory services operating in disrupted environments, we focus our summary of related work on ICN approaches applied to delay/disruption-tolerant networks.

The key differences among ICN schemes stem from the ways in which content is named or routed. The main approaches to naming in ICN consist of using self-certifying flat names (e.g., [51]), human readable hierarchical names (e.g., [42]), or metadata expressed as attribute-value pairs (e.g., [17]). Broadly ICN schemes aimed at DTNs are based on either the epidemic dissemination of content (e.g., [89]), the dissemination of interests (e.g., [80]), or the maintenance of distributed hash tables (DHT) to allow nodes to publish and subscribe to content at specific nodes or geographical locations (e.g., [33]). As argued, epidemic content dissemination consumes considerable bandwidth; and DHT-based approaches proposed to date are such that content (or links to content) may have to be placed far away from where it is produced or consumed in a DTN, which can incur substantial delays.

Haggle in [83] is a pioneering project that aims at bringing a content-centric solution for the PSNs. It provides the application layer with a rich set of functionalities which could be further extended to built applications based on opportunistic communications. The key feature of Haggle is its search-based data dissemination framework, making it easy to share content directly between intermittently connected mobile devices. Haggleâs approach is based on its identification of search as a first class operation for data-centric applications, providing the underlying functionalities for neighbor discovery, resource management and resolution â thus removing the need to implement such features in applications. With search-based resolution the proposal upgrades searching as a networking primitive within the architecture, such that matching data is also transparently received from peers as they are encountered in the network. The novelty of this approach compared to Publish/Subscriber proposals is that the matching between data and receivers is not binary; a top ranked match is the best only relative to lower ranked ones. Each device can hence limit the amount of disseminated data to only the top ranked nodes with the most interest in the data. In contrast, other approaches to dissemination make use of binary matching filters or topic channels (refer to Sec. 2.2.1.1) that are static and lack relative matching and ranking.

ICEMAN in [93] is another proposal designed for military tactical scenarios. It is a generalization of the Haggle architecture where information exchange occurs in proximity encounters among two nodes. ICEMAN uses UDP broadcast for the dissemination of interests, and UDP and TCP for the exchange of content. It uses network coding, the exchange of Bloom filters, and utility-based content caching to make the dissemination of content more robust in the presence of severe disruptions to network connectivity. While Haggle takes an agnostic stand about data naming specification, ICEMAN takes a declarative naming approach in which subscribers identify content as weighted attribute-value pairs and specify a satisfaction threshold and maximum number of matches. The process of finding matching data objects in the local cache occurs whenever a data object is received. A node effectively computes a degree of satisfaction metric for each description denoting the satisfaction of the match. Data objects are retrieved, ranked, and prioritized at each node using a lexicographical ordering based on the degree of satisfaction and the creation time stamp (i.e., greatest satisfaction and freshest first). Only data objects exceeding the threshold specified by an application constitute a match and are eligible for dissemination.

The Haggle architecture due to its flexible design has served as a starting point for many proposals delivering messaging capabilities in OppNets. The representative applicative scenarios devised in this dissertation were implemented and studied in simulation environments and the porting of such solutions into this reference architecture is left as a future work.

3

Related Work

This chapter surveys state-of-the-art literature material relevant to our research work. The material exhibited is as consequence of our approach at tackling the issues of opportunistic communications networks. To this end, in Sec. 3.1 we provide a concise survey of content sharing proposals in the realm of mobile networks, outlining their network model, the entities composing the system along with the networking techniques enabling content sharing and distribution. Next, in Sec. 3.2 we introduce research work employing the public transportation system as a routing backbone, aimed at providing service support to elastic, non real-time applications. In this context, we also discuss the proposed forwarding schemes, pointing out the differences and rationale behind the choices. Concluding this chapter in Sec 3.3 is brief introduction on the floating data concept and a survey of relevant research work in this stream.

3.1 Content Sharing in Mobile Networks

Overlay networks are an application layer instrument used in todays content distribution platforms such as P2P content sharing, providing a set of networking primitives not available at lower layers of the protocol stack (e.g., content search). The overlay is typically an application layer abstraction of the physical network providing basic functionalities such as naming and message routing capabilities.

Classical overlay approaches proposed for infrastructure communication platforms are based on the assumption of stable end-to-end path between communicating nodes. However, mobile networks are characterized by frequent disconnections, partitioning the network resulting in a major traffic overhead due to overlay maintenance algorithm(s) [32]. Below we review two relevant research streams aimed at porting a content sharing solution to the mobile scenario differing from one another depending on whether routing capabilities are provided or not.

Optimized Routing Independent Overlay (ORION [50]) proposes a special purpose overlay network organization and maintenance algorithm where routes are set up on demand by the search algorithm and maintained as long as necessary (e.g., mobile node is out of reach). The search algorithm is tied to the *file routing table* which is filled during this phase and stores several redundant paths for copies of the same data content. Due to changing network conditions, content producer might change during a transfer; thereby, control over the transfer is kept on the receiver-side and, opposed to TCP, the ORION transfer protocol does not maintain an end-to-end semantic. ORION follows an unstructured overlay approach and does not depend on the deployment or support of any specific MANET routing protocol. The system provides search capabilities and the way search is orchestrated is by employing a controlled-flooding technique in the connected portion of the network. ORION poses no restrictions on the network size and data requests are routed to the appropriate destination by exploiting the information built during the search phase. From their experimentation the authors show that ORION outperforms the Gnutella [74] unstructured overlay approach, drastically reducing maintenance overhead.

FastTrack over AODV [84], is based on the FastTrack unstructured overlay network using Ad hoc On demand Distance Vector (AODV [70]) for routing purposes. FastTrack differently from the above approach assuming a flat addressing scheme, follows a hierarchical architecture in which high-capacity peers are Super Nodes (SN) and low-capacity peers are Ordinary Nodes (ON). Each ON is associated to one SN, and SN have many ON associated to them. Searching through the network is done by employing a controlled-flooding through the SN backbone. The authors, validate their approach in a controlled, small scale scenario comprised of solely mobile nodes. From the experimentation the authors show that an integrated approach, establishing the connection immediately and not sequentially after a file has been found, helps drastically reduce search delays in compression to the two-step approach.

Mobile Peer-to-Peer (MPP [75]) is yet, another system combining the Gnutella unstructured overlay over the reactive Dynamic Source Routing (DSR [47]) protocol. Gnutella adopts a similar strategy as FastTrack for overlay organization and searching, that is, queries are flooded through SN backbone. The authors validate the system and demonstrate good performance for mobility scenarios up to 20 m/s, above this speed the network performance decreases drastically.

The above approaches shield the upper layers from the negative effects of node mobility by employing a reactive approach to network topology construction, maintaining connections until they are necessary. Another popular stream of research relaxes the assumption of an overlay approach but rather rely solely on pair-wise encounters between nodes for message exchange. The assumption is that node mobility and availability is so unpredictable that any end-to-end semantic has little usefulness, thus in practice there is no necessity for a networking layer.

One of the first advocates of such an approach is BlueTorrent [49]. BlueTorrent proposes a P2P content sharing application based on ubiquitous Bluetooth-enabled devices such as PDAs, smartphones etc. The envisaged scenario is comprised of static access points supplying the data content and mobile nodes exchanging pieces of data with each other when outside the access point reach. The authors identify node discovery as a key ingredient in system efficiency and discuss some augmentations speeding up the basic inquiry mechanism. Differently from the above approaches, BlueTorrent does not employ routing but makes use of a swarming protocol whereby data availability is piggybacked into inquiry packets and broadcasted to neighboring nodes. Content transfer takes place during pair-wise associations between nodes and when multiple connection opportunities are available the one with the best Receive Signal Strength Indication (RSSI) is selected.

Similar in spirit to BlueTorrent, PodNet [59] proposes a system architecture for delay-tolerant public content distribution which could be implemented upon any link layer technology. PodNet does not rely on a networking layer, hence, no nodes are no addressed directly and content is the solely addressable entity. The proposal relies on opportunistic data transfers between the mobile nodes to spread the content when they are outside the coverage of fixed access points. As in the previous approach, PodNet does not exploit multi-hop data transfer but relies solely on pair-wise associations between the mobile nodes for data exchange.

Our aim is at porting a *self-sustained* content sharing solution between mobile disconnected devices enabling sharing and distribution without infrastructure reliance. Differently from the surveyed literature above, our envisaged network is comprised of solely mobile devices producing content and consuming it when in proximity with each other. This mobile opportunistic network could be comprised of human-operated mobile devices moving in restricted physical spaces, such as conferences, university campuses, refectories, clubs and in many other social settings. In essence, they are characterized by nodes with heterogeneous contact rates, high mobility and limited information. Moreover, it might be the case when content producers and consumers are spatially and temporally decoupled, hence, how search and retrieval is orchestrated in this disconnected environment is what we aim to investigate.

The communication platform aimed at sustaining this kind of service cannot rely solely on pair-wise encounters for message exchange. The multitude of contexts nodes are immersed in could give rise to multi-hop capabilities that can help extend their reach are to other data not available on its immediate vicinity. We anticipate that we depart from a synchronous communication model, equipping the system with a complementary mechanism following an asynchronous data exchange similar in spirit to DTNs.

The content-centric nature of the applicative scenario we are considering lends itself to the NDN design philosophy, introduced previously through Sec. 2.3. To this end, we deem worth investigating a similar design to our proposal as an additional minor contribution to the domain itself. In the next section we provide a survey of state-of-the-art proposals following the NDN paradigm, aimed at providing service connectivity in mobile dynamic environments.

3.1.1 NDN Proposals in Mobile Networks

The NDN architecture has built-in mechanisms addressing mobility issues arising from the host-centric Internet design. Indeed, by decoupling the data from their location, caching of data along the path by leveraging the network as an intermediate repository, and native multi-homing support at the strategy layer could help alleviate these emergent criticalities. While this communication paradigm has been proposed as an architectural replacement for the current Internet, some research effort has been focused on tailoring its design to mobile networks bereft of infrastructure.

This research is mainly confined to the vehicular networking domain aimed at supporting named data communication among vehicles and vehicles and infrastructure entities. The first study showing the effectiveness of NDN communications in this domain is represented by work [90]. In this work the authors show through simulations that a NDN approach outperforms MobileIP in terms of delivery profiles in scenarios where both consumer and producer vehicles are mobile and communication between the former is supported by the network. This is due to the intrinsic nature of NDN communication networks whereby the data (Interest) is decoupled from the individual communication channel with the consequence of the data being able to exploit (routed) through multiple in-network paths toward the producer and vice versa. MobileIP instead ties the particular request (consequently the response) to the specific network interface (address space) which due to mobility might not be available anymore.

Departing from infrastructure-supported communication toward a fully qualified vehicular network bereft of infrastructure are works [92, 91, 36]. These works have a common denominator that of exploiting data muling, identified as a critical function in NDN-based mobile networks and a departure from the NDN operations in wired Internet. In these proposals data muling serves two critical roles: not only it helps disseminate the information to vehicles in many different locations, it also contributes to keep information available even when the original publisher has gone off the system (e.g. reached its destination and turned off). Basically, mules contribute into pushing data further into the network making it available at the edges. In specifics, in work [92] the authors propose a vehicular information service comprised of the following entities: *(i)* consumers interested into the data, *(ii)* producers, registering events from sensing their surrounding environment, *(iii)* data mules, caching and physically carrying sensed data. The retrieval process occurs by broadcasting the Interest packet to the neighborhood, *hopping* that a nearby vehicle does have a match satisfying the issued request.

The same authors in [92] propose a collision avoidance technique at the NDN-layer aimed at addressing the inefficiencies introduced by the RTS/CTS 802.11 link layer mechanism. They validate their proposal by studying the effectiveness of this mechanism in a push-style information dissemination scenario whereby a producer injects/broadcasts a message into the network and intermediate nodes forward/broadcast it on their turn away from the producer. The novelty of this work, although it is not stated explicitly, is that it introduces the concept of active mules whereby data generated into one location is further pushed by intermediate nodes so as to make it available elsewhere. Differently from [92] mules can also act as forwarders of data instead of merely physically carrying it.

By far the first work in this domain, explicitly introducing a forwarder role into the picture is the work [36]. As in previous work, mules physically carry sensed data even if there is no matching PIT entry with the addition that both requests (Interests) and data packets can be carried and later on forwarded if conditions are met. A forwarder acts as such whenever it is in communication range with other vehicle(s).

Our contribution in this domain is to port a content-sharing solution for human-comprised networks, tailored to the design of NDN architecture. Consumers and producers in our network model might be spatially and temporally decoupled, hence data muling and forwarding are both roles of crucial importance. The Interest/Data-forwarding model proposed by the above surveyed works resembles that of epidemic message dissemination approach might prove harmful when considering our targeted scenario. Through Sec. 4.1 we discuss the networking techniques aimed at sustaining the targeted scenario. Later on, in Sec. 4.1.4 we re-formulate our design to the specifics of the NDN architecture.

3.2 Public Transportation System as a Service Delivery Platform

Delay tolerant networking has evolved from an outer space architectural paradigm to a viable architecture for terrestrial applications. For this purpose many research efforts have been devoted to devise efficient and reliable data distribution strategies [76, 44, 96, 62]. In particular, Bus switched networks (BSN), OppNets deployed on top of PTS, have gained a lot of interest due to their practical use.

3.2.1 Deployment Architecture

The first contributions [69, 24, 55], propose a carrier-based approach aimed at interconnecting rural villages of developing areas. Their common goal is to provide network access for elastic non real-time applications so that the local population may enjoy basic Internet services (e.g., e-mail and non-real time web browsing). The deployment context of these works is characterized by a small number of nodes with fewer contacts when compared to an urban environment.

Campus bus networks (e.g., [97, 7, 64]) are designed to serve students and faculties who commute between colleges or from/to nearby towns. These kinds of services are usually characterized by a relatively small number of nodes when compared to a fully fledged urban environment. The main contribution in this direction is represented by [64] and [97], where five colleges are linked with nearby towns and to one another over an area of 150 square miles. On this same bus network, a system of throwbox nodes [13] was deployed to enhance the capacity of the DTN.

Scaling up in terms of number of nodes, we find urban environments where a considerable number of lines are densely deployed to enable people to commute inside a city. Bus networks in urban environment (e.g., [8, 46, 3, 57]) are usually characterized by many contact opportunities and frequent contacts. In [8], authors propose a commercial application called Ad Hoc City. Based on a multi-tier wireless ad hoc network routing architecture it provides elastic Internet access by means of access points (APs), which are responsible for a geographical area. The proposed system targets general-purpose wide area communication. Messages from mobile devices are carried to the AP and back using an ad hoc backbone that exploits buses. The authors verified the validity of the proposed approach against real movement traces by King County Metro bus system in Seattle, WA.

Using the same real data as for [8], the authors of [46] propose a cluster-based routing algorithm for intra-city message delivery. In [46] an efficient large-scale clustering methodology is devised: nodes are clustered based on the basis of encounter frequency while multi-copy forwarding takes place between members of the same cluster hosting the destination node. To reduce the overhead effect of having multiple copies in the network, the authors of [3] model forwarding as an optimal stopping rule problem. The contribution from [57] uses data from the PTS of Shanghai to test the performance of a single-copy forwarding mechanism. This is a probabilistic routing strategy where probabilities are related to intra-contact times as in [77].

3.2.2 Routing in BSNs

Among all possible real-field applications of DTN we can identify OppNets build on top of the public transportation system. In this context, buses inter-contact times are quite long although their contacts happen to occur according to their schedule, because connections are mostly respected. Routing policies are heavily influenced by the determinism of such a scenario. Despite the everyday metropolitan experience suggests to mistrust determinism in bus contacts, some routing proposals in the literature still attempt to exploit it through different types of oracles. In this scenario, the time tables are required to take into account the traffic conditions, and the concept of connection between two bus lines has a relaxed time constraint, which is eventually respected by increasing the number of buses belonging to each line. These environments experiences intermittent connectivity due to a variety of reasons, yet the topology often has an underline stability.

In [28], a *Contacts Oracle* is proposed that, given two buses IDs, returns the next encounter time. This oracle requires an *a priori* knowledge of encounters, and is impossible to implement in a real system. In the same work, a more feasible approach uses a *Contacts Summary Oracle* that, given two buses IDs, returns the average inter-contact time.

A rural environment (e.g., [69, 64, 24]) consists of a number of villages spread on a large territory and connected typically by buses. In these cases, the set of neighbors for every node is usually small but does not change frequently during time; failures to delivery a message is the mostly the result of a missed encounter rather than unpredictable node mobility. These proposals do not employ routing. Both requests and responses are locally stored at an infrastructure entity, acting as proxy servers between end-users and the Internet, and it is up to the carrier to download the queued requests and upload the outcomes.

In [69] the public transportation system is used as an opportunistic backbone to carry messages between collection points (kiosks) located in the participating villages; buses are used as *data mules* with a best-effort approach. A more refined approach is proposed by [64], where bus-to-bus connectivity is exploited to forward messages on a multiple hops. In this last case timetables are considered: authors propose an algorithm to provide a delivery probability for messages forwarded on each path. In the past paper, ([24]), authors propose a modified link-state routing protocol capable to exploit link uptime predictability. The proposed algorithm build forwarding paths by means of sending link-state advertisements with very long lifetimes and keeping caches of sent advertisements on intermediate nodes.

The shared goal of all mentioned projects is to provide network access for elastic non real-time applications in order to enable base Internet services to the population (e.g., mail and non-real time web browsing).

Campuses bus networks (e.g., [13, 97]) are designed to commute students and faculties between colleges or from/to nearby towns. These kind of services are usually characterized by lower number of nodes if compared to a fully fledged urban environment. The reference contribution in this direction is represented by [13], where five colleges are connected with nearby towns and between each other. Authors of the aforementioned paper propose a multi-copy routing algorithm, namely MaxProp, based mainly on messages priority. These priorities are based on the path likelihoods to destination nodes accordingly to historical data and other complementary mechanisms. By means of simulation MaxProp is shown to outperform oracle-based protocols based on knowledge of deterministic meetings between peers. This work has been extended in [97], where inter-contact times distribution have been analyzed both at bus and line level. A generative model for inter-contact times based on real traces to has been proposed in order to generate synthetic traces to drive simulations on routing protocols performance.

Scaling up in term of number of nodes with find the truly urban environment where a considerable number of lines is densely deployed to assist people transfer inside a city. Bus networks in urban environment (e.g., [8, 46, 78]) are usually characterized by many contact opportunities and frequent contacts.

In [8] authors propose a commercial application, namely *Ad Hoc City*, based on a multi-tier wireless ad-hoc network routing architecture to provide elastic Internet access by means of Access Points which are responsible for a geographical area. The system target general-purpose wide area communication. Messages from mobile devices are carried to the AP and back using an ad-hoc backbone exploiting buses. The validity of the proposed approach has been validated against real movement traces by *King County Metro bus system* in Seattle, WA. Using the same real data as for [8] authors of [46] propose a cluster-based routing algorithm fro intra-city message delivery. In [46] an efficient large-scale clustering methodology is devised; nodes are clustered based on encounter frequency and a multi-copy forwarding happens toward any member of the same cluster of the destination node.

However, adopting a multi-copy approach is scarcely appealing in a wide urban environment. To lessen the overhead effect of having multiple copies in the network [3] models every forwarding as a optimal stopping rule problem; this way traffic overhead is sensibly reduced while delivery ratio comparable with a fully epidemic approach.

All the city-oriented strategies mentioned above follow one common approach: they perform a multiple-copy routing. As already stated in this paragraph, having multiple copies of the same packet using network resources may be an hindrance when scaling up to city level using dozens – if not hundreds – of routes counting many hundreds of buses. Single copy strategies have not been really considered in literature due to the possibly low delivery ratio and long delivery time. Contribution from [78] is, at the best of our knowledge, the only approach using a single-copy forwarding mechanism on a very large scale. This last work considers intra-contact times as a metric to use for a link-state protocol; as a result, routing is performed through the most encountered line in term of time regardless of the real encounter probability.

Further contributing in this realm of research, we propose a concrete deployment architecture and applicative scenario exploiting the PTS as a service delivery platform. To this end, we have modeled two realistic deployment scenarios where carriers are public buses with routes corresponding to actual PTS lines in Milan, Italy, and Chicago, Illinois, and users are mobile entities owning handheld devices issuing data requests for data available elsewhere. We study the performance trend of our Mobile-Delay Tolerant Network (MDTN) in an urban-wide deployment under different routing strategies and data distribution schemes.

3.3 Mobile Floating Data

In the foreseeable future everything will be interconnected and producing data, from inanimate things or objects to us the people bringing into this digital frame our social profiles, interests, processes etc. Answers to a query about a nearby producer, a multimedia content, a process, a person, an experience, an ability, etc. could be answered locally without relying strictly on infrastructure communication platforms. This network of data grows and shrinks meeting the peoples demand and *vanishes* whenever it is not required anymore (e.g., not alimented by the people). In this envisioned scenario data has geographical and temporal validity, which is in contraposition with the current approach of infrastructure networks where data is made available globally and stays on forever.

We identify data survivability and accessibility in the interest area as key ingredient in enabling the above depicted scenario. Once the data is *injected* into the network a mechanism is needed to guarantee its immediate availability in the anchored area. To this end, we could employ a solution where data is replicated on all the network nodes (e.g., just time node encounter the data source or mule) augmenting the chances of data survival under the mobility scenario. However, this approach results resource-consuming in terms of transmission costs and network storage as a whole. At the same time it does not provide any guarantee of data being present in the anchor area.

To address this problem we delved into the literature of content placement and replication techniques to find if similar approaches could suit our case. Content placement and replication techniques have been thoroughly studied in the scenario of infrastructure networks like the Internet [6]. In this context, topological information is stable and known *a priori*, in contraposition to mobile networks where topological data is volatile and transient in time. Hence, in the mobile scenario we cannot bind (anchor) the data to fixed storage locations as in the infrastructure approach.

Opportunistic caching solutions like HybridCache in [95] and Hamlet in [30] rely on information exchange amongst neighboring nodes to fulfill their respective objectives. HybridCache allows nodes on the data path to cache the relaying item if its size is small, otherwise nodes just cache the data path. Hamlet has the goal is to save storage space while achieving content diversity via estimating the cached items in the neighborhood. Both these techniques, rely on local information to perform their duty, hence, lead to suboptimal, best-effort solution in achieving their goal.

An emerging idea is that instead of binding data to physical, fixed storages we anchor data to geographic location, decoupling the content placement problem from the changing network topology. The authors of [53] propose a location-based strategy for content placement considering content popularity as an index for data replication inside a computed area. Through experimentation it is shown that their proposal does outperform the other techniques, reducing data access costs (e.g., number oh hops). The underlying idea of this approach is to define multi-level virtual grids and allocate popular contents to fine-grained grids so that they are more likely available from nearby. However [15] requires content popularity to be known in advance (a closed-world assumption) in order to compute content anchoring strategy. However in our scenario, the data is produced and maintained by the network nodes itself. In these settings the proposal above reduces to simply maintaining an arbitrary number of replicas for each data in a bounded geographical area.

Communing all the reviewed proposals is the common objective of reducing data access costs in terms of hops required to retrieve it. However, in the scenario depicted above the data is produced locally and is volatile. If no mechanism guaranteeing data survivability (availability) is in place the data vanishes without giving it a chance. To this end, enabling this depicted scenario, we propose a decentralized algorithm whereby nodes cooperate toward a common goal, that of guaranteeing data survivability. Our solution lends itself to another interesting feature, that of controlling the spatial distribution of data through the network, hence, the possibility to control the data access costs as in the solutions discussed above.

4

Opportunistic Data Gathering and Dissemination

Until now we have discussed the emerging OppNet paradigm and related works falling in the context of our chosen applicative scenarios (refer to Chap. 2 and Chap. 3). In particular, through Chap. 1, we announced the strategy we pursued aimed at enabling the envisioned opportunistic communication platform. Indeed, rather then facing the scenario in its general form, we do so by focusing our study toward representative applicative scenarios. To this end, we were guided from the lack of coverage in literature on pull-style service delivery for OppNets and investigate this scenario in its facets. With the objective of enhancing the performance of the former approach, we set on a trial of investigating an opportunistic carrier-based approach, exploiting the Public Transportation System (PTS) as a routing backbone, providing support for elastic, asynchronous data exchange. Given the inherent local nature of ad-hoc networks, we propose a protocol built upon existing solutions supporting an innovative scenario enabling the *floating data* concept. Altogether this scenarios are complementary and could be employed in synergy with one another to deliver a potentially urban-wide opportunistic communication platform enabling message exchange amongst mobile users without strict infrastructure reliance.

Through this chapter we will discuss each of the chosen scenarios and the path taken towards their implementation. In specifics, in Sec. 4.1 we discuss the content sharing scenario, outlining the network model and the devised protocol stack aimed at supporting communication in sparse networks paying particular emphasis on the proposed *delegation forwarding* strategy (refer to Sec. 4.1.3.2). Further, in Sec. 4.1.4, we discuss a named-data approach to the proposed solution tailored to the NDN reference architecture. The evaluation of the devised strategy and system components are postponed and discussed through Sec. 5.1. Following, through Sec. 4.2 we set on a trial of devising an urban-wide, carrier-based approach aimed at supporting elastic, non real-time applications. To this end, we start by outlining the system architecture and its *modus operandi*. Next, we provide some details regarding the simulation environment and the strategy adopted to evaluate the proposal. The evaluation of the devised architecture is postponed and discussed later on through Sec. 5.2. Concluding this chapter is Sec. 4.3, where we provide an analysis of the *floating data* concept and solution aimed at sustaining and infrastructure-less, provider-less network of data alimented by the mobile users themselves. In this context, we start by discussing the systems assumptions and rationale followed by the proposed solution. As for the other scenarios, the evaluation part is postponed and exhibited in Sec .5.3.

4.1 Mobile-to-mobile Content Sharing

The mobile user is facing many options for wireless access with highly varying characteristics, including shorter and longer disconnection periods. This mobile technology revolution and its growth in popularity will eventually lead these appliances to become the dominant mode by which users interconnect. Indeed, mobile users could leverage other forms of connectivity which do not necessarily require pervasive infrastructure presence. From these bases, it seems natural to employ ad hoc networking as an enabler, providing communication support to the disconnected mobile content-centric world. In this scenario, interactions among users could take place without the strict requirement of infrastructure mediation, providing service access through a carrier-independent, infrastructure-less communication platform.

4.1.1 Scenario Definition

Our aim is to devise a *self-sustained* content sharing solution through which mobile nodes could share and distribute content without relying on infrastructure. These mobile opportunistic network is comprised of human-operated mobile devices moving in restricted physical spaces, such as conferences, university campuses, refectories, clubs and in many other social settings. For instance, they could include networks of commuters sharing every morning and evening the same train/bus. In essence they are characterized by nodes with heterogeneous contact rates, unpredictable mobility and limited information.

In a content sharing context, low node density relates to a low data population, which if not addressed properly could undermine the system utility. To this end, we propose a mobile-to-mobile content sharing solution, *M2MShare*, tailored to the characteristics of the mobile disconnected networks. M2MShare introduces a novel solution aimed at

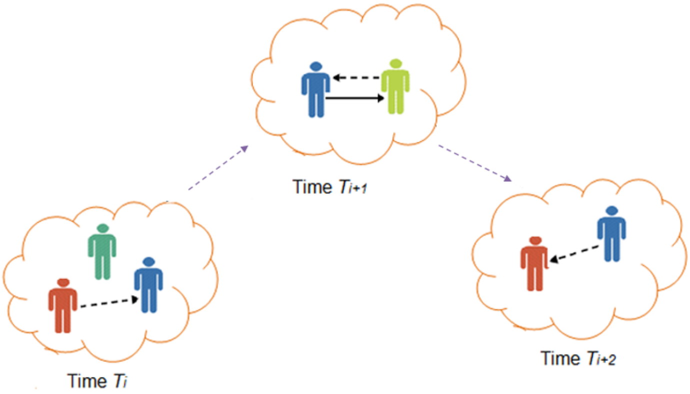


Figure 4.1: Delegation forwarding exemplified: a node at time $\mathrm{T_i}$ searches for a specific data content in the local connected network it is immersed in. The content is not found and is subject to delegation, triggered when certain criteria are met. At time $\mathrm{T_{i+1}}$ the servant node issues a query into a local connected network he is immersed, resulting in the data content being found and retrieved. At time $\mathrm{T_{i+2}}$ the cycle closes with the servant forwarding the consumer the requested content.

addressing communication in sparse networks. It does so by providing the means for an asynchronous data exchange similar in essence to that pioneered by outer space DTNs. In particular, content retrieval is performed by both disseminating a request packet in connected portions of the network but also by *delegating* the request, representing an unsatisfied, unaccomplished content retrieval task to encountering nodes.

Indeed, due to the distributed and dynamic nature of this environment, content producers and consumers might never be connected at the same time in the same network. Hence, a synchronous communication model does not always suffice. The delegation forwarding scheme addresses this issue by extending nodes reach area to other connected local networks (Fig. 4.1). When a node eligible for delegation, called servant node, is encountered a request is issued and assigned. It is up the servant to retrieve it and forward back the content once it encounters again the consumer. In synthesis, servant mobility is exploited to reach data content available in other standalone connected networks; a servant has the burden to perform the task and later on, when the consumer node is met, return its outcome (output).

To avoid excessive transmission overhead, requests are assigned only to *frequently encountered nodes.* Since no information about the producers is available we delegate the responsibility of finding that particular content to nodes that we might encounter again in the future. Our focus is on exploring a new mechanism allowing nodes to expand their reach area to other connected portions of the network. This is achieved by leveraging on node mobility and periodic encounters among users even if they are not aware of this social proximity (e.g., commuters utilizing the same train every morning).

The organization of this section is as follows: in Sec. 4.1.2 we discuss the general system assumptions with emphasis on the data plane. Following, in Sec.4.1.3 we introduce the M2MShare protocol stack, describing the duties and responsibilities of each individual layer. In particular we discuss the delegation forwarding scheme outlining its *modus operandi* and design criteria. Next, in Sec. 4.1.4, we introduce a named-data oriented approach for M2MShare tailored to the design of the NDN paradigm. Section 4.1.5 discusses the strategy taken to evaluate our proposals, starting by providing some insights on the simulation framework and the adopted mobility models. The experimentation outcome is later on discussed in Sec. 5.1.

4.1.2 Network Model and System Assumptions

The network under consideration is comprised of mobile devices operating M2Mshare, transparently exchanging messages when in proximity with each other. However, communication is not confined to pairwise encounters between nodes, but as we explain later on, M2MShare is capable of exploiting both multihop synchronous and asynchronous communication, this based on the data satisfiability on the current connected portion of the network. As for the addressing scheme, nodes are identified by their

Medium Access Control (MAC) addresses, hence, the assumption of a flat rather than hierarchical addressing scheme.

An important function intrinsic to content sharing systems is the search mechanism where a broad category of mechanisms have been proposed ranging from keyword and simple pattern matching to information retrieval and content-based retrieval. We take an agnostic stand in this direction and do not tailor the design at any search scheme, assuming each node has an indexed list of local files made available for others to retrieve it. Furthermore, in our experimentation we make the assumption that content identifiers are already known and requests are issued with the unique content identifier, that is the user already knows the targeted data. This might have happened due to a previous issued query (e.g., a keyword query).

4.1.3 Node Architecture Overview

A protocol stack was designed, providing core functionalities that a content sharing system must provide. The protocol stack is described with the help of Fig. 4.2 and main modules are listed below.

1. *Search module*: Each node is equipped with a filesystem module providing basic indexing and search capabilities. As stated queries are issued and consequently content is identified by a unique identifier (e.g., a string);

2. *DTN module*: This module is responsible for servant election and task delegation. Studies in routing algorithms for challenging environments such as OppNets demonstrated that they have a social dimension built-in [83]; knowing that behavior patterns exist allows for better routing decisions to be made. We exploit the fact that certain users frequently encounter each other (e.g. by taking the same bus in the morning, by eating in the same cafeteria at lunch, etc.) in

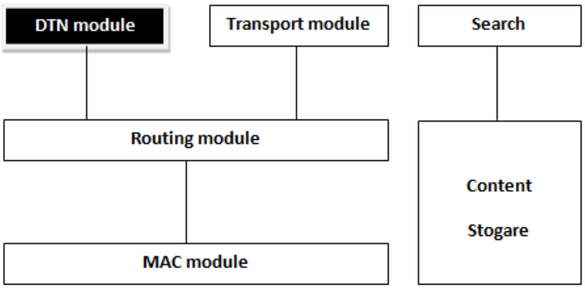


Figure 4.2: M2MShare node architecture.

order to dynamically build a DTN path from source to destination whenever it is required;

3. *Transport module*: It provides the task queuing mechanism and task life cycle management. An important part of this module is the communication protocol for data packet exchange among nodes. Also, it provides a smart data content, chunk division strategy, which allows for parallel and hence faster download while avoiding redundancy on downloaded data;

4. *Routing module*: This module provides multihop message forwarding capabilities and implements a controlled flooding technique like AODV in the connected portions of the network. Paths are set up on demand and maintained only as long as necessary;

5. *MAC module*: This module provides node discovery and message broadcast capabilities. The heart of this module is a fundamental service called *PresenceCollector* which periodically gathers presence information about in-reach area devices so as to determine which nodes are frequently met and have a reasonable expectation to be met again in future (these nodes will be used as servants in order to propagate unaccomplished or unsatisfied tasks).

4.1.3.1 Presence Collector

M2MShare actively collects presence information of encountered devices that are in direct reach area of communication so as to exchange data and assign delegations. This job is handled by an active daemon of the system, called *PresenceCollector*. It is important to understand that two nodes A and B are not aware of each other immediately after they enter in communication range. Rather, while being in communication range, they learn the existence of the other as soon as one of the two initiate a scan phase by sending out presence beacons and the other node answers. To this aim, the PresenceCollector is modeled as an active daemon which periodically scans the network with a periodicity configurable by the user; a high frequency (e.g. a period of 1 s) is not reasonable from the energy preservation point of view.

Instead, having a low frequency (e.g. a period of 10 min), the device may miss an encounter with another node that lasts, for instance, just 3 min, and hence the chance to elect and delegate an unaccomplished task to a potential good servant. Also a servant might miss the chance to initiate an output forward of a previously delegated task while the consumer device is in-reach area but not yet discovered by its periodic discovery service. Both this situations refer to a particular class of tasks in our system whose creation and execution depends directly on the inquiry frequency.

To better understand the impact of beaconing frequency on energy consumption, the histogram in Fig. 4.3 shows the battery lifetime of a Samsung Galaxy S2 smartphone running only the PresenceCollector service with different periods between two consecutive scans (5, 60, 100 and 1000 s). The resulting battery lifetime values were obtained by first measuring the energy expenditure of a single scan, computed as the average over 10 sampling points and assuming a linear energy consumption model. The measurements were done using the MONSOON power mon-

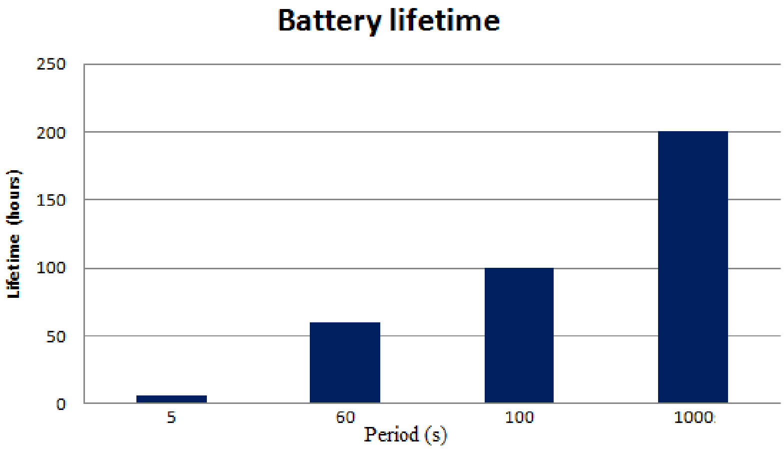


Figure 4.3: Battery lifetime with varying beaconing frequency.

itor solution [60]. The results are quite intuitive and expected. Tackling the problem of data transfer, let's assume that device A has delegated to another device B some particular retrieval task and that device B was able to accomplish it; then, the next time they will encounter each other, the servant B will notify the consumer A that it is ready to forward the output of that particular task. In this case, the quantity of data that the consumer A will retrieve from device B depends on different factors (Fig. 4.4):

1. the time interval between two subsequent beacon transmissions (T_p);
2. the duration (D) of the established communication link between device A and B. This embodies the amount of time that might be actually used for data transfer and that is generally smaller than the physical encounter duration (T_e);
3. the bandwidth available on the consumer side for data transfer (B_W), which we consider to be constant during all link establishment time, neglecting factors such as interference, protocol activities and other possible on-going transfers (upload or download).

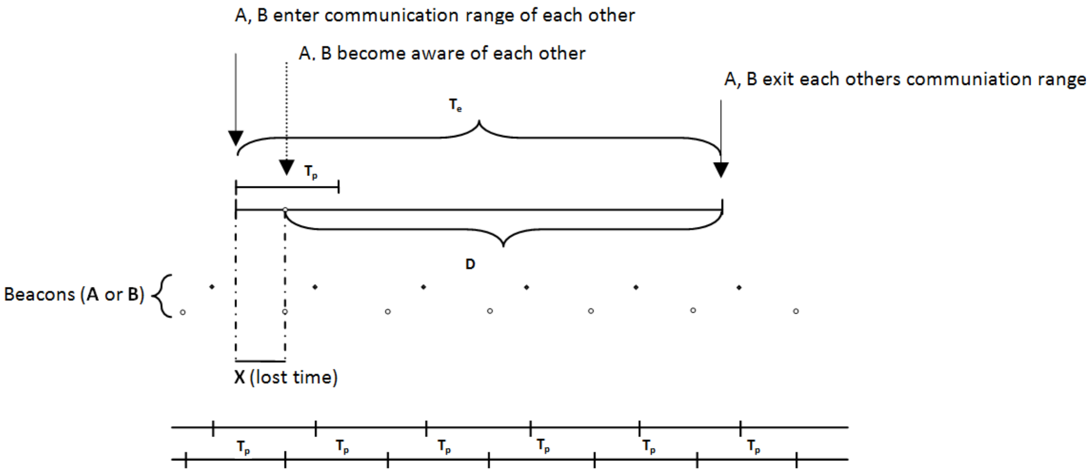


Figure 4.4: Problem description.

For the sake of simplicity, communication delays and queuing delays are not considered in the study; they are both considered as negligible amount of time. The two nodes may start the content transfer only after they become aware of each other; this happens when the first presence beacon is sent (by either A or B) after the two nodes has become in communication range. Therefore, as Fig. 4.4 shows, every time there is a lost time X before starting the content transfer.

Theorem: The average time lost for data transfer between devices A and B entering in communication range with each other, before becoming aware of each other, is

$$E[X] = \frac{1}{3} \times T_p \tag{4.1}$$

Proof: Let X denote the interval between the moment at which the distance between two devices A and B becomes smaller than the transmission range and the moment when the first device between A and B broadcasts a beacon message. Let X_A be the interval between the moment at which node A and node B enter in the transmission range of each other and the moment when node A transmits its beacon message. Similarly, X_B is the interval between the moment at which node A and node B enter in the transmission range of each other and the moment when node B transmits its beacon message. As node A and node B periodically transmit their beacon messages independently from each other, then X_A is independent from X_B and X = min (X_A, X_B).

From the single node perspective the probability P(X_A < t), where t ∈ [0, T_p] denotes the lost amount of time, is:

$$P(X_A < t) = \frac{t}{T_p} \iff P(X_A > t) = 1 - \frac{t}{T_p} \tag{4.2}$$

Since the two nodes independently transmit their beacons, the lost amount of time is characterized by the following probability function:

$$\begin{aligned} F(t) &= P(X \leq t) = P(min(X_A, X_B) \leq t) \\ &= 1 - P(X_A > t) \times P(X_B > t) \\ &= 1 - (1 - \frac{t}{T_p})^2 \end{aligned} \tag{4.3}$$

In order to compute the expected time lost, we need to integrate the product with its density function. To this aim we know that f(t) = dF(t)/dt, where F(t) and f(t) denote the Cumulative Distribution Function (CDF) and the Probability Density Function (PDF), respectively. The resulting formula for the expected time lost is hence expressed by (4)

$$E[X] = \int_0^{T_p} f(t) \times tdt = \int_0^{T_p} \frac{2}{T_p} \times (1 - \frac{t}{T_p}) \times tdt \tag{4.4}$$

From this result, we can derive the average data quantity that can be transferred when two nodes enters in the transmission range area of each other.

Corollary: Given T_e *the average time node* A *and node* B *stay in the communication range of each other,* T_p *the frequency of periodic inquiry of each node and* Bw *the bandwidth available at each node, then the average data quantity transferred is*

$$T_d = (T_e - \frac{T_p}{3}) \times B_w \tag{4.5}$$

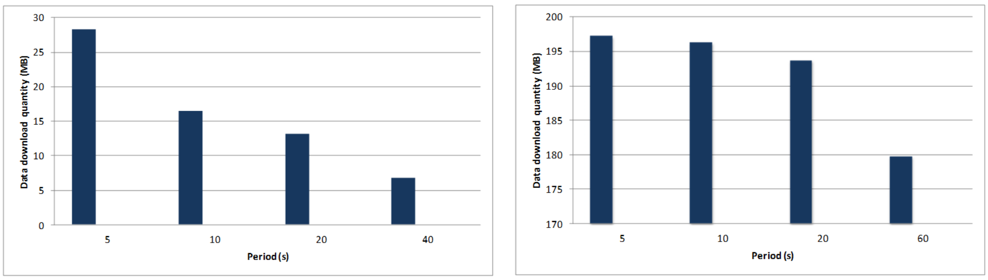
Proof: Given the overall expected time of the encounter (T_e) and the expected lost time ($T_p/3$) after which both devices can initiate the transfer, then the remaining time for data transfer is:

$$E[D] = T_e - T_p - T_a - T_c \tag{4.6}$$

Since we are under the assumption that T_e (queuing delay) and T_e (communication delay) are negligible and we know the average link duration between the two nodes from (4), then we can compute the average data quantity transferred after link establishment, which is:

$$T_d = (T_e - \frac{T_p}{3}) \times B_w \tag{4.7}$$

To understand the impact of this lost time on data transfer, depending on the beacon frequency, we have run a simulative experiment based on the introduced mathematical modeling and the outcome is reported in Fig. 4.5. In our simulations, we have fixed the physical encounter duration between nodes A and B, considering different periods of time between consecutive beacons (i.e. 5 s, 10 s, 20 s, and 40 s). In Fig. 4.3 we saw that to a lower beacon frequency corresponds a lower energy usage; but, as expected, we now see that it also corresponds to a lower data transfer amount.



(a) Beaconing frequency of 20s

(b) Beaconing frequency of 40s

Figure 4.5: Data transfer amount with varying beaconing frequency.

4.1.3.2 Delegation Forwarding

M2MShare implements an asynchronous communication mode between nodes where a consumer node can delegate an unsatisfied, unaccomplished task to a servant node. By task delegation, is meant that a task is locally encoded by the consumer and communicated to the servant, which locally stores it for later execution. When a servant accomplishes the task, it is ready to forward the output to the consumer that requested the task accomplishment. The *return* takes place the next time they encounter each other. In essence, we leverage on node mobility to reach data in other disconnected networks where they might be available. Obviously, each delegated task has a *TTL* (time to live); the task is stored in the servant's local storage and can be forwarded to the consumer if task has not expired before. The servant does not re-schedule a task that is unaccomplished at TTL expiration.

While in DTNs there are pre-deployed entities that store-and-forward data along the destination path (routers), M2MShare achieves this functionality in an infrastructure-less environment, where this forward route is established dynamically along the path to destination. In other words, at each hop a consumer node, which in turn might be acting as servant for another node along the chain, dynamically chooses its servant to which delegate the task. Unlike DTNs where source and destination are different entities in our case both source of the request and destination of the data (task output) are the same entities, while servants are intermediary nodes along the chain, which store-delegate-and-forward back the task output.

As stated earlier, mobile devices are used by people whose behaviors are better described by social models and the fact that behavior patterns exist allows better routing decisions to be made. Therefore, the underlying assumption of our solution is that each user has a routine of his own that matches other user's routines. For instance, a user staying at his office could be in communication range with colleagues during working hours, a user traveling by bus or train to go to work frequently encounters other commuters, the same every day.

Delegating an unaccomplished task to all the nodes in the established overlay is bandwidth and energy consuming therefore a criterion is needed to choose one node instead of others. Also, it is sound to delegate tasks to peer devices operated by users whom we expect to encounter again in the future. In this way we augment the chances of output return, in case the servant found the desired content. In the current implementation those devices that are seen more than a *Frequency Threshold* ($Freq_Th$) number of sessions are elected as servants to whom the system delegates a particular unaccomplished/unsatisfied task (if any).

A servant device is a frequently encountered device and the concept of frequently encountered changes in time, adapting to the observed dynamics. This because the contact rate of a single device operating M2MShare might vary from day to day. Moreover, some devices frequently encounter many other devices, while others encounter a small number of them. In the first case we would want to higher the expectations of a frequently encountered device in order to choose the best devices from those repeatedly encountered. In the second case, in order to have a certain number of frequently encountered devices we should be less selective by lowering the system expectations (parameters).

Algorithm 1: M2MShare: Parameter Tunning

```
   input  : Probation Window ($P_w$), Active Ration ($A_r$), Expected Ratio ($E_r$)
   output : Updated $Input$ parameters
1  $E_r \leftarrow$ L $/P_w$;
2  if $E_r \leq A_r$ then
3      $P_w \leftarrow P_w - 1$;
4      if $P_w < 2$ then
5          $P_w \leftarrow 2$;
6          $Freq_{Thresh} \leftarrow Freq_{Thresh} + 1$;
7      end
8  else
9      $P_w \leftarrow P_w * 2$;
10     if $P_w > 30$ then
11         $P_w \leftarrow 30$;
12         $Freq_{Thresh} \leftarrow Freq_{Thresh} - 1$;
13     end
14 end
```

The servant election algorithm, at the beginning of each day imposes a goal that needs to be achieved during that day. This goal is the *Expected Ratio* (E_r), the number of elected nodes (servants) expected during one day. Since one day's activity might differ at some level from the others, the system tries to adapt the configuration parameters to the observed dynamics in order to achieve a better performance (Expected Ratio). Essentially, the algorithm tunes the configuration parameters using past history of observed encounters.

Initially the ratio is computed by the default configured values, no history seen before and at some point in time it is expected that the algorithm will reach an equilibrium where the configuration parameters (E_r, $Freq_Th$) will be stable or will not be subject to frequent change. To better explain how the algorithm works let us refer to the pseudo code shown above in 1:

At line 1, E_r is computed, except day 0, using the information gathered the day before; we impose a goal for todays activities based on data gathered the day before. The underlying assumption is that user has its own routine and habits which do not change radically from day to day. The Expected Ratio is computed by performing the division between the number of servant slots in the servant list (L) and the current probation window value (P_w). There might be users operating the software that have a high number of encountered devices per day and others whom have only few of them. By dividing with P_w the algorithm can tune the parameters *(P_w, $Freq_Th$)* imposing a sound goal for tomorrow's activity based on the user's capability of encountering other devices.

At line 2, E_r is achieved and we lower the monitoring period, decrementing it, imposing a higher goal for next time. A monitoring period P_w=*2* means that a node is considered periodic if it is seen $Freq_Th$ times in 2 days. In this case the device is frequently encountering nodes and electing them, so everything seems going well. Frequently encountering and electing servant/s does not necessary mean that the they are returning back the output of the delegated tasks and we do not have any instrument or criteria to determine whether this is the case or not. When the monitoring period goes below its minimum value (i.e. when it is less than 2) we increment the $Freq_Th$ and probe whether this high frequency of election is induced by a probable low threshold.

At line 9, we take a conservative approach, doubling the monitoring period and by doing so we lower the system expectations (ratio halved) for the next probation window. Ratio could not be achieved either because threshold is too high or effectively we are encountering a small number of devices (e.g. the device has been switched off). A monitoring period P_w=*30* means that a node is considered periodic if it is seen $Freq_Th$ times in 30 days. If the monitoring period exceeds its maximum value we decrement $Freq_Th$, imposing a lower threshold for election and probe whether a low frequency of election is induced by a probable high threshold.

4.1.3.3 Transport Level

At the transport level M2MShare exploits connected, multihop routes by following an approach similar as ORION (refer to Sec. 3.1). Connections in the connected portion of the network are established on demand and maintained as long as necessary. The transfer protocol utilizes the routes given by the content and response routing tables for transmission of control and data packets. The *content routing table* (CRT) may store several redundant paths for copies of the same content. Due to changing network conditions, a content producer might change during a transfer; thereby, control over the transfer is kept on the receiver-side, hence, we do not maintain an end-to-end semantic. Summarizing, at the transport level we exploit both synchronous (connected) multihop paths in the connected portions of the network as well as asynchronous (disconnected) multihop paths towards content producers.

For transfer, a content is split into several blocks of equal size. Since the maximum transfer unit of the mobile network is assumed to be equal between all neighboring nodes, the block size can be selected such that the data blocks fit into a single data packet. The receiver sends a *DATA-REQUEST* message for one of the blocks along the path given by the CRT. Once the DATA-REQUEST reaches a node storing the data content in the local repository, the node responds with a *DATA-REPLY* message, containing the requested block of the content. Later on Sec. 4.1.3.5, we will discuss the content division strategy aimed at augmenting the chances of retrieval.

To avoid ambiguities, we remember that our solution is not tailored to any specific network level protocol. The devised solution could be layered upon any existing best-effort transport level protocol. The naming scheme we adopt is flat (e.g., level-2 MAC addresses), hence, a node-based communication scheme.

4.1.3.4 Congestion Control and Request Handling

Requests are disseminated into the connected portions of the network by employing a controlled-flooding techniques as AODV. If a query request is pending on the local device and conditions are met, it is disseminated into the connected portion of the network. If a match is found transfer can initiate.

Otherwise, if no data matching the criteria was found the node defers query transmission until the request is either subject to delegation or the neighboring nodes have changed since the last issued query. Indeed, a query is issued again if the nearby neighbors of the consumer node have changed with respect to the last request. This capability is provided by the *Pending Query Table* (PQT) and helps avoid unnecessary redundant query messages. Obviously, this is an heuristic as changes might have occurred in the local connected network but are not visible by monitoring only 1-hop reachable nodes.

If delegation occurs, meaning a consumer is in the vicinity of a frequently encountered node, it is up to the receiver (servant) to accomplish the task. Delegation on the servant side is treated like a pending task which is on its turn subject to the same rules. In each case, the arrival of a query/retrieval request triggers a content matching event on the receivers side. If no match is found the query/retrieval is remembered and relayed to other neighboring nodes.

4.1.3.5 Content division strategy

Popular infrastructure-based P2P content sharing systems divide data into chunks (e.g., Gnutella [74]), which are the atomic transferable parts. In this context, a node has real time vision of what is happening, which data is being transferred and from whom. This is a good starting point but taking into consideration the possibility of delegations and in order to increase the chances of eventually receiving the requested content while reducing the number of transmissions, we require a more flexible partitioning strategy.

M2MShare provides a new content division strategy where data can be retrieved in pieces and a piece size varies. The data content is seen as map of non overlapping intervals of variable length that need to be retrieved (Fig.4.6). When a user chooses to initiate a data transfer, a task is created and scheduled for execution. Initially there is only one interval to be retrieved that is the entire content *[0, contentSize]* (Fig. 4.6) Once a servant becomes eligible, a *DATA-REQUEST* is issued containing the missing data interval, in this case *[0, contentSize]*. The following delegation are issued with overlapping data intervals so as to augment the changes of data transfer being completed while at the same time avoid exessive transmissions.

The starting point of the next interval to be requested is calculated by the following formula $d = (1+2p)/(2n)*contentSize$, where n denotes the current number of pieces the initial interval *[0, contentSize]* is composed of and p denotes the next interval on the current partitioning to be fetched.

To better illustrate how these intervals are computed, let's consider some potential scenarios, referring to Fig. 4.6. In each case the whole data content may be retrieved from a single servant, yet the starting point for the transfer varies as follows:

1. **Case Fig.4.6-A and case Fig.4.6-B**. Two successive delegations occur:
    (a) a task is delegated to servant 1 demanding the entire data content to be retrieved;
    (b) a task is delegated to servant 2 starting from the middle of the

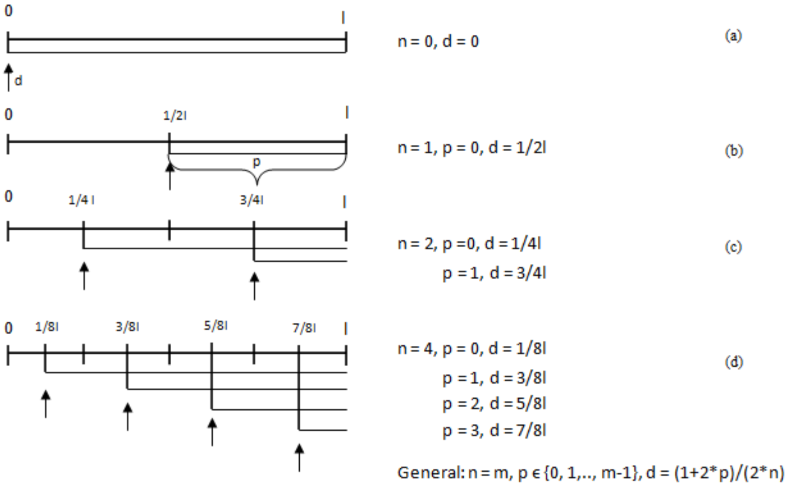


Figure 4.6: Content division strategy exemplified.

data content, i.e. point (1/2)*contentSize

2. **Case Figure Fig.4.6-C**. Four successive servants have been elected:
    (a) a task is delegated to servant 1 and 2 as described above;
    (b) a task is delegated from producer 3 starting from point (1/4)* contentSize of the data content;
    (c) a task delegated from producer 4 starting from the point (3/4)* contentSize of the data content.

In essence, the starting point of the requested interval is calculated so as to halve the largest interval left undivided on the original interval. Clearly, when the end of the data content is reached, the servants might continue retrieving the next pieces of data left to retrieve. However, in case transfers are prematurely interrupted by disconnection, they will all have retrieved different parts of the content thus maximizing the possibility to have cumulatively retrieved the whole content instead of having redundantly transferring the same limited part of it.

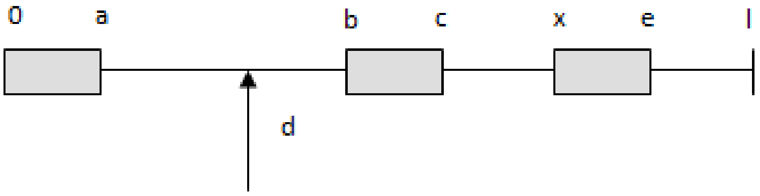


Figure 4.7: Transfer map example; some data pieces have been downloaded; the resulting map in this case is d;[a+1,d-1],[d,b-1],[c+1,x-1],[e+1,l].

Once the assigned data interval is retrieved from the producer we would desire to use this source to retrieve other missing intervals while in reach. On the other side, it is possible that a data request is not entirely satisfied (e.g., node out of reach) and between the next starting point (d) and interval end point (I) may exist retrieved parts interleaved with missing ones (Fig. 4.7). To represent this, retrieval map is used and provided to future servants, with the format specified in Fig. 4.8. This format includes the indication of the starting point d and missing intervals, those not yet retrieved.

As mentioned earlier, DTNs often use message replication techniques along the destination path in order to increase the probability of data reaching the destination. In our case the content division strategy might add redundancy during data transfer as it can happen that at concurring producers are requested overlapping data intervals. However, if two producers come at different points in time, e.g. a disconnection occurred and transfer was not concluded, the next requested data interval consists of only free, missing intervals.

4.1.4 A Named-data Approach for Disconnected Mobile Networks

The content-centric nature of the scenario under scrutiny lends itself to the NDN design introduced and discussed in Sec. 2.3.1. NDNs built-in behavior of decoupling the data from their location combined with the caching of data along the path make it a suitable candidate to tailor the design of our M2MShare. In this section we discuss the design of our system tailored to the NDN paradigm. Unless stated otherwise, the components discussed previously remain the same.

While NDN is proposed as a clean-slate approach to the current Internet, a lot of research work has been focused in devising proof of concept applicative scenario in the realm of vehicular networks. As discussed in Sec. 3.1.1, these solutions address mainly named-data communications in level-2 connected mobile networks. However, in our scenario producers and consumers might be spatially and temporally disconnected. Hence, they might not be connected at the same time in the same network. This is to say that a pure named-based solution is not feasible. Instead, the alternative is to employ a named-based, synchronous communication in connected portions of the network and a node-based, asynchronous communication between servant(s) and consumer(s) (e.g., task forwarding).

Multihop capability

Our solution is situated at the application layer of the NDN protocol stack and the multihop capability as discussed in Sec. 4.1.3.3 is guaranteed with the instruments provided by the NDN proposal. For this

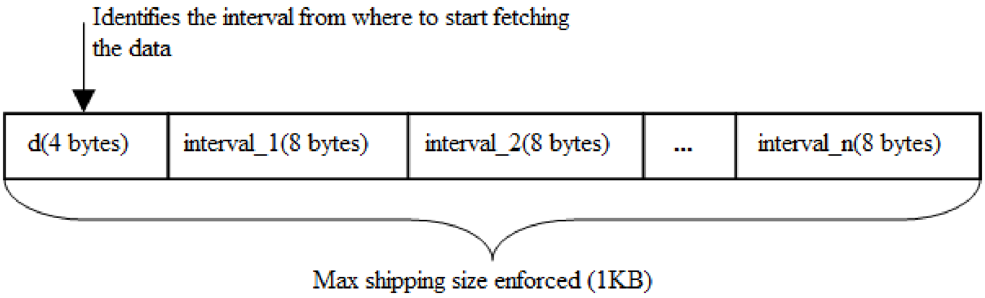


Figure 4.8: Transfer map format.

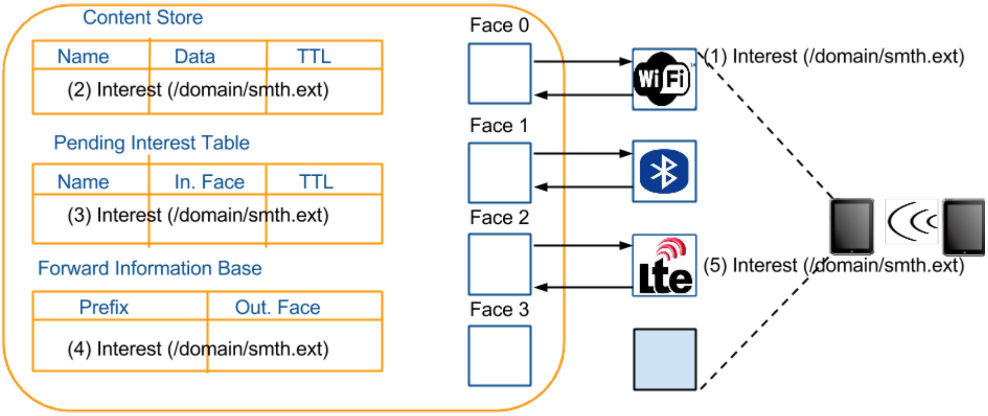


Figure 4.9: NDN Interest match flow.

section the DATA_REQUEST and DATA_REPLY packets are referred to as Interest and Data respectively, as denoted in the NDN architecture.

The NDN approach uses a single FIB (*/m2mshare*) entry denoting our application face (domain) and the forwarding strategy it implements. The FIB is similar in essence to the IP forwarding table where instead of IP addresses, named-data are used. Any issued application issued Interest triggers on the receiver side and NDN-flow where the data name is matched against the Content Store (CS) and Pending Interest Table (PIT) in this order (refer to Fig. 4.9). If no match is found the Interest is remembered in the PIT and relayed to other neighboring nodes using the FIB entry. PIT entries are expired after a predetermined TTL which is set by default at 30 msec. To be noted, that the underlying medium is accessed in broadcast rather than unicast.

The Interest while traversing the network nodes leaves a trail which is later on used to route the response back to the consumer node. A consumer initially issues a query Interest for the data content and for query dissemination we employ Bloogo [4], a gossiping algorithm proposed in the NDN context proved to employ the minimum number of transmissions while guaranteeing total network coverage. If a match is found transfer can initiate. It might be the case that the named content might not fit into a single packet. As stated in Sec. 4.1.3.3, it might be the case that the data might not fit into a single packet. In the same way, large files are broken up into multiple pieces, e.g., some data is broken up into a sequence of *contentIdentifier/1*, *contentIdentifier/2*, and so on. The next time a consumer issues an Interest for the next data sequence, a query is broadcasted following the same steps as previously described.

Naming and Packet Structure

We adopt the following general naming structure: */m2mshare/dataName*. The *dataName* denotes either a query about the data or directly identifies the data of interest. As discussed in Sec. 4.1.2 we do not tailor our design to a specific query mechanism instead for the experimentation purposes we address the content directly (e.g., the consumer already knows the data identifier after a previously issued Interest). The first naming component, *m2mshare*, serves as the application id, making the naming structure to be application dependent. As for node naming, we adopt an identifier guaranteed to be unique for the whole node population (e.g., its MAC address or an obfuscated MAC address computed with an appropriate hash function). Hence, the proposed naming structure is flat and it is to be attributed to the decentralized and dynamic nature of this network model.

In Tab. 4.1 is shown the header of an M2MShare packet.

1. *Interest*: When a node n running M2MShare issues a content Interest c, it generates an outgoing request packet placing c and n in the *dataName*, *reqName* field respectively. Depending whether the request is being delegated or disseminated into the network the delegation bit is set accordingly. If the Interest expresses a delegation (bit set to 1), the Interest is persisted by the application and undergoes the same process as until now described.

Table 4.1: M2MShare named-data approach packet header fields.

Bloogos header	the Bloogo neighbor table as defined in [4]
nonce	packet identifier
nodeName	node identifier
type	the packet type, either an Interest or a Data response
delegation	differentiates between an Interest and a delegation
delegation map	the retrieval map attributed to this delegation
dataName	Content identifier or query data
dataFound	denotes if a data was found in the Data packet

The *Bloogo header* is used to adopt a Bloogo-like forwarding strategy. To avoid requests being relayed indefinitely into the network, complementing and the mechanisms discussed previously, each node identifies duplicate packets using both *reqName* and *nonce* fields in the packet header. The latter is a random number generated by the consumer before issuing the request. When intermediate nodes receive a fresh data request an NDN-like flow is triggered. If no data matching the criteria is found and the request is a delegation, it is remembered in the PQT as previously discussed (refer to Sec. 4.1.3.4). Otherwise, a PIT entry is created or updated and the request is relayed to neighbor nodes.

2. *Data*: If a node receives a request for a data content being either a delegation or query request, it checks whether it is in cache. If data is available an outgoing relayed packet with the *dataFound* bit set to 1 is send in response. The data packet travels in the backward path following already build in the request dissemination phase. This packet is also used to invalidate active delegations (if any) for that content.

   It might be the case that the named content might not fit into a single packet. Large files are broken up into multiple pieces, e.g., some data is broken up into a sequence of *contentIdentifier/1*, *contentIdentifier/2*, and so on. The next time a consumer asks for the next piece of data in the sequence, the request travels directly towards the producer following the established path.

4.1.5 Simulation environment and Evaluation Strategy

The simulation environment chosen for our experimentation is the Opportunistic Network Environment (ONE) introduced in [86] an agent-based discrete event simulator. The simulator combines movement modeling, routing simulation, visualization and reporting in one program. Movement modeling can be done either on-demand using the integrated movement models or movement data can be imported from an external source. The node position data provided by the movement model is used for determining if two nodes can communicate and exchange messages, this depending on the communication radius of the technology in use. We point out, that the ONE does not provide any support for simulation of physical/MAC layer properties, instead it focuses only on the routing layer and performance of DTN algorithms. Transport level logic and the functionalities previously discussed were implemented extending the simulator and released as a contribution to the community [65].

Contact patterns among nodes can also be exported exported for routing simulation in external simulators or can be given to the internal routing modules which implement several different DTN routing algorithms. The internal routing modules perform operations on the messages on their own, but they can also be commanded using event generator modules or external traces. To make it suitable and efficient enough for simultaneous movement and routing simulation, the ONE adopts a time slicing approach, so the simulation time is advanced in fixed time steps. The time slicing can be complemented by scheduling update requests between the fixed time steps for higher simulation time resolution.

ONE is high modular, as shown in Fig. 4.10. The movement model and routing protocol are provided by independent modules, which can

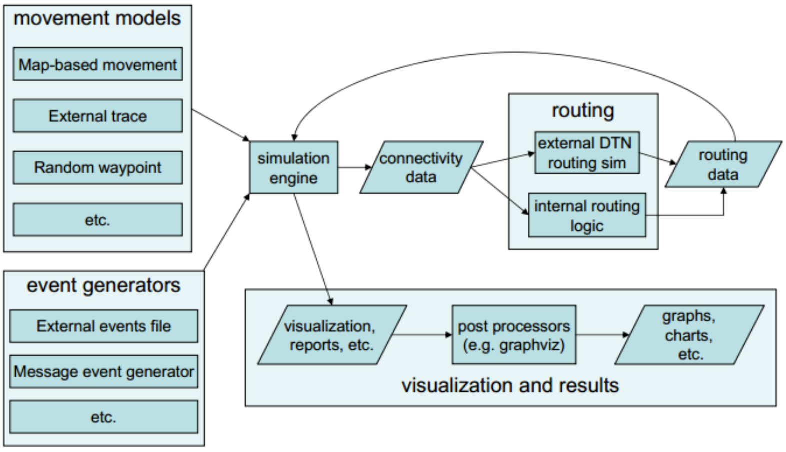


Figure 4.10: ONE architecture [86].

be dynamically loaded depending on the simulator settings. This allows a relatively easy implementation of new mobility models and routing protocols in the simulator.

4.1.5.1 Evaluation Strategy and Performance Metrics

Before discussing how the simulation of M2MShare was conducted, we start by introducing some terminology required to better comprehend the experimentation part:

1. *Population*: refers to the number of nodes in the simulation which emulate people operating M2MShare. This number does not include nodes which emulate public transportation entities, like buses or trams. People are distributed in districts of the map, as later on described;

2. *Content size*: refers to the size of the data content a consumer is interested in;

3. *Content popularity*: refers to how many copies of the data content are present at the beginning of the simulation world;

4. *Content distribution*: refers to how data content(s) is distributed among the population at the beginning of the simulation, while producers are chosen uniformly chosen from among all the active nodes or from subsets of them;

5. *Delegation Type*: refers to the type of delegation forwarding mechanism employed by the nodes. We study three scenarios:

   (a) No_delegation: strategy not employing the delegation forwarding strategy. Represents the scenario where content retrieval happens only if a producer is found in the connected portion of the network;

   (b) M2MShare: uses the delegation forwarding techniques employing the algorithm 1;

   (c) Delegation_to_all: strategy where consumer(s) delegate, eligible for delegation tasks to all encountered nodes in the network – represents an epidemic approach to data dissemination.

6. *Delegation Depth*: denotes the length of a DTN path;

7. *Multi-hop delegation probability* (MhDP): the probability that a servant node would delegate again a pending/incomplete task, used only for simulations where only multihop delegation is employed;

8. *Content Division Strategy*: refers to the type of content division strategy employed. It can be of three types:

   (a) M2MShare: uses the algorithm explained in Sec. 4.1.3.5 in choosing the initial transfer point for the requested data content;

   (b) iM: strategy where the entire data content is requested;

   (c) rM: the initial transfer point is chosen randomly from the available map.

Throught the experimentation part (refer to Sec. 5.1), we set on a trial to show the demonstrate the effectiveness of the devised delegation forwarding strategy studying key performance metrics. Crucial at achieving reliable performance indexes for the devised system and its components is the mobility model used to evaluate it. Our approach is to adopt state-of-the-art models mimicking human behavior as close to possible. To this end, we have adopted the map-based Working Day Mobility (WDM [25]) model supported by the simulator in use and a complementary data trace found in the CRAWDAD directory [83]. In the following section (Sec. 4.1.5.2) we briefly discuss both the approaches under use so as to keep the discuss contained, more details about the models can be found in the respective articles introducing them.

4.1.5.2 Mobility Models and Characteristics

For our simulations we have adopted a both map-based mobility model that of Helsinki city center and the Haggle data trace available from the CRAWDAD repository [83]. Below we provide a brief overview of the traces under scrutiny, starting with the map-based model, discussing how the simulation world is orchestrated along with some configurations of the mobility model itself. The content patterns for this model are omitted as they might vary depending on the random seed used to generate the model, instead we provide the configuration parameters necessary to reproduce them. Concluding we discuss the Haggle data trace by enumerating the content pattern characteristics we found.

Map-based model

The map-based scenario has a size of about 8000 x 7000 m^2 (Fig. 4.11). To increase the trustworthiness of the experimental outcome we have included realistic day-by-day node movements through the Working Day Movement. The WDM is able to represent both the unpredictability of

Table 4.2: Map orchestration in regions.

District	Nodes	Offices	Meeting spots
A	150	30	4
B	50	10	1
C	100	20	2
D	100	20	2
E (A + B)	100	20	2
F (A + C)	150	30	4
G (A + D)	150	30	4
H (Whole map)	200	40	5

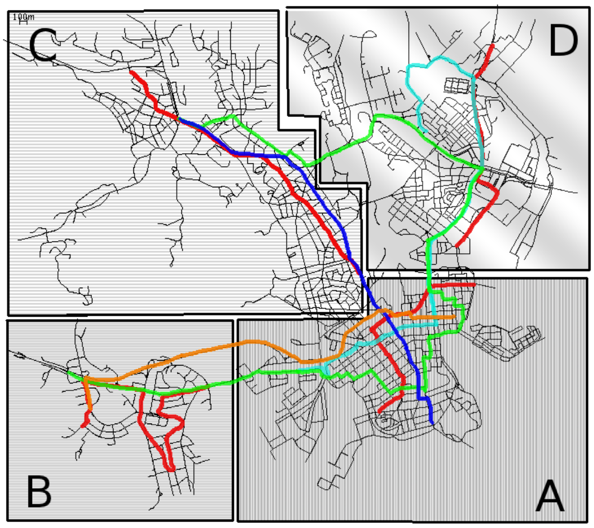


Figure 4.11: Simulation world map evidencing the map sectors and the bus routes interconnecting them.

certain movements of users and the routine of other movements such as, for instance, the daily trip from home to work. This provides a good approximation of inter-contact times and contact durations, providing the flexibility for configuring real life test scenarios.

We use the same configuration parameters described in [25]. Nodes population is variable during the simulations but the map is always divided in districts, with overlapping area between them. There are several bus routes, one for every district and every node can use a bus of the route belonging to the same district of the node. When a node is *walking*, the speed is set between 0.8 and 1.4 m/s and for *buses* between 7 and 10 m/s, with a 10 - 30 s waiting at each stop. Half of all the nodes were set to travel by car and the speed of *cars* is set to 20 m/s to make it a faster way to move between locations.

To emulate differences in peoples lifestyle, especially in morning wake up times, the differences in *schedules* of nodes were drawn from a normal distribution with a standard deviation of 7200 s emulating a situation where about 68% of the population leaves home between 7 and 11 in the morning. Every node has a probability of 0.5 of doing some *activity* in the evening, after work, with *groups size* variable between 1 and 3 nodes. The working day length is set to 28800 s (8 hours), which is a value common to a large number of jobs, and the pause times inside the office is drawn from a Pareto distribution with coefficient 0.5 and minimum value 10 s. The office size is set to a 100mx100m square, to compensate for the lack of floors, walls and other furniture (refer to Tab. 4.2). Finally the communication range of mobile devices is set to 10 m, which is common for most Bluetooth devices.

Haggle Contact Trace

Concerning the mobility model we have adopted the Haggle contact trace documented and reported in [83]. The dataset includes a number

Table 4.3: Characteristics of the Haggle data trace.

Total Devices Present	36
Total Trace Duration (days)	11.4
Network Type	Bluetooth
Mean # Encounter/node	591
Median # Encounter/node	632.5
Mean Unique # Encounter/node	30
Median Unique # Encounter/node	31
Mean # Encounter/node/day	886.75
Median # Encounter/node/day	1556.5
Encounter Duration Median (sec)	563.0
Encounter Duration Mean (sec)	1083

of traces of Bluetooth sightings by groups of users carrying small devices (iMotes) for a number of days varying from office environments, conference environments, and city environments. Mobile users in the experiment mainly consist of students from Cambridge University asked to carry iMotes with them at all times for the duration of the experiment. In addition to this, we deployed a number of stationary nodes in various locations that we expected many people to visit such as grocery stores, pubs, market places, and shopping centers in and around the city of Cambridge, UK. A stationary iMote was also placed at the reception of the Computer Lab, in which most of the experiment participants are students.

Through Sec. 5.1 we evaluate the networking techniques aimed at sustaining a content sharing service among mobile disconnected devices. We first start exhibiting the experimentation outcome for the node-based, connection-oriented approach, employing a split TCP in the connected portions of the network, than move toward the evaluation of the NDN design of the scenario.

4.2 Public Transportation System as a Service Delivery Platform

The carrier-based approach has proven more viable than its human counterpart, mainly due to the quasi-deterministic nodes mobility: buses move along pre-determined paths and follow an a priori known schedule. In these settings, routing algorithms can be devised on reasonable assumptions and probabilistic predictions of encounters [97, 7, 69, 24]. Despite their practical usage, even this scenario has a crucial and unsolved technical challenge: the scalability of network when applied to larger areas, with growing number of lines, and a potentially huge offered load. Since size and shape of bus lines are limited by human and organizational factors, network delivery delay may ramp up with the covered area due to the increasing number of hops each packet must traverse.

In this context, pursuing the scenario announced in Chap. 1 we propose and analyze the performance of Mobile Delay/Disruption Tolerant Network (MDTN): a delay tolerant application platform built on top of a public transportation system able to provide opportunistic service connectivity for elastic applications. We show by means of extensive simulations the performance trend of our MDTN. To this end, we have modeled two realistic deployment scenarios where carriers are public buses with routes corresponding to actual PTS lines in Milan, Italy, and Chicago, Illinois, and users are mobile entities owning handheld devices.

The remainder of this section is organized as follows: in Sec. 4.2.1 defines the scenario under consideration, providing some insights on the system *modus operandi* and architectural design. Following, through Sec.4.2.2 we discuss the simulation environment and parameters used to evaluate the performance of the devised solution. The results are later on discussed in Sec. 5.2.

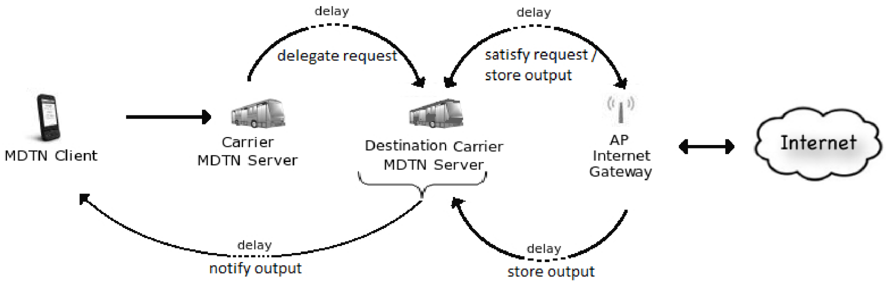


Figure 4.12: MDTN system orchestration.

4.2.1 Scenario Definition

In Fig. 4.12, are depicted the three main entities involved in the architecture: *(i)* the MDTN client or a mobile user operating a wireless-equipped handheld device, *(ii)* the MDTN server or the wireless data carrier which locally stores usersâ requests and tries to fulfill them when Internet connectivity is available, and *(iii)* the Internet Gateway (IG): an Internet-enabled wireless access point deployed at each bus end of line (e.g., bus terminals). We have considered positioning IGs at bus terminals since these are joint points between different bus lines and because the bus company may already have an infrastructure-based end point there (e.g., offices). However, we do not preclude the possibility of a future investigation of IG distribution in terms of a scientific optimization problem, even though less practical in real deployments.

The basic functionality of MDTN is as follows. Once the user gets on board of the carrier (e.g., the bus), it establishes a connection with the on-board hosted server (an base station hosting the MDTN server) and may issue a request. The request consists of a content identifier (e.g., an Uniform Resource Locator (URL) and a destination bus line where the user is going to pick up the response later on. The goal of MDTN is to provide the user with the desired content at the specified bus line. After the request has been issued, it is opportunistically forwarded by the carrier toward one of the IGs located along the line (or at line ends) and finally, the response has to reach the destination line where the content is expected.

The process described above involves a multi-hop opportunistic routing of both request and result from carriers toward one of the IGs and from the IG toward the destination bus line. The delivery process is independent from the underlying routing strategy and occurs as follows:

(i) If the carrier holding the request encounters during its trip a carrier belonging to a valid next-hop line, criterion based on the adopted routing strategy, the request is forwarded. It is now up to the new carrier to satisfy the request if it reaches its end of line;

(ii) If the carrier holding the request reaches its terminal, it is able to fulfill the request by itself; the response will be forwarded toward the destination line;

(iii) If the response reaches a destination line carrier, response dissemination among carriers of the same line can take place. In this phase, forwarding occurs during opportunistic contacts between destination line carriers (if any).

The rationale behind the last announced step of the delivery process is that there might be many carriers traveling on a destination line and outcome dissemination among them is needed, speeding up the delivery process.

Once in proximity with the wireless carrier, the client establishes a connection with the MDTN server to receive the request outcome when/if the request has been accomplished. Obviously, the MDTN client can disconnect/connect from/to the MDTN server at any time and the outcome will be forwarded the next time they pair with each other. The delivery process is entirely transparent to the client.

4.2.2 Simulation Environment and Evaluation Strategy

In this section we describe the simulation environment, and successively provide the details about the specific parameters used for evaluation.

4.2.2.1 The URBeS Simulation Framework

All simulations have been performed using the Urban Routing BackbonE Simulator (URBeS): an ad-hoc simulator presented and validated in [31]. URBeS acquires real city topological data and the relative PTS timetable to accurately recreate bus movements in a real urban environment in order to simulate data forwarding between buses as well as between buses and road-side network devices. This simulation platform is also able to support any external routing policy in order to compare the performance of various routing algorithms. The functional scheme of URBeS is reported in Fig. 4.13.

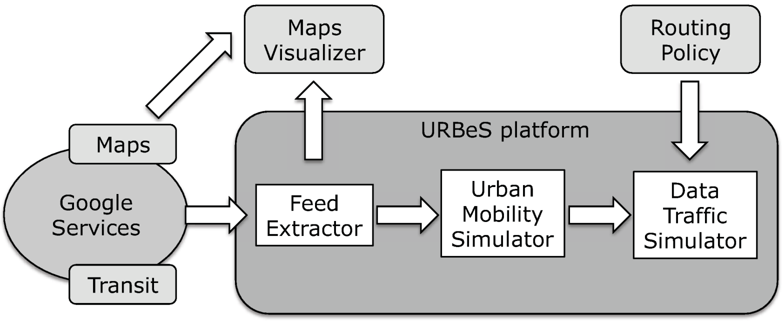


Figure 4.13: URBeS functional scheme.

The analysis of a city PTS starts from a Google Transit [35] feed, which is a database of planned trips, provided by a transit authority. The URBeS framework is then composed of three sequential modules.

In the first phase, URBeS parses information from the feed and produces a timetable of bus movements together with a topology of the PTS layout. GPS coordinates of every bus stop, also taken from the Google transit feed, are converted to Cartesian coordinates.

The output from the first phase is fed to an urban mobility simulation module which is in charge of generate mobility traces for all the buses based on the real PTS timetable. Nodes along each line move between coordinates accordingly to real timetables adopting a constant speed between stops. Pauses at stops are (when planned) also included in the transit feed and thus simulated accordingly. In this phase, URBeS also computes statistics about bus contacts. They are useful for predicting intra- and inter-contact times and for understanding city coverage of the PTS.

The last phase adds data traffic to the picture: network traffic is randomly generated by each bus and then delivered following the provided routing policy. URBeS logs detailed information to profile delivery rates, delays, and locations where forwarding takes place. Our simulator can provide a valid comparison between different routing policies while allowing easy testing on multiple urban environments.

The development of a new simulator has been required in order to:

- Overcome poor scalability, in term of total traffic and number of nodes, from existing products (e.g., GloMoSim [34] and ns-2 [63]).
- Introduce an highly optimized urban canyon model which is missing even in more modern simulation environments (e.g., the ONE simulator [86]).
- Better integrate external data sources such as Google Transit.

URBes has been positively evaluated against GloMoSim, for low traffic levels, using the same output coming from the second stage briefly introduced above.

4.2.2.2 Urban Environment

In order to evaluate our proposal we selected two cities: Milan (Italy) and Chicago (IL). These cities have been selected as indicative of all the

Table 4.4: Properties of PTS layouts for the two cities.

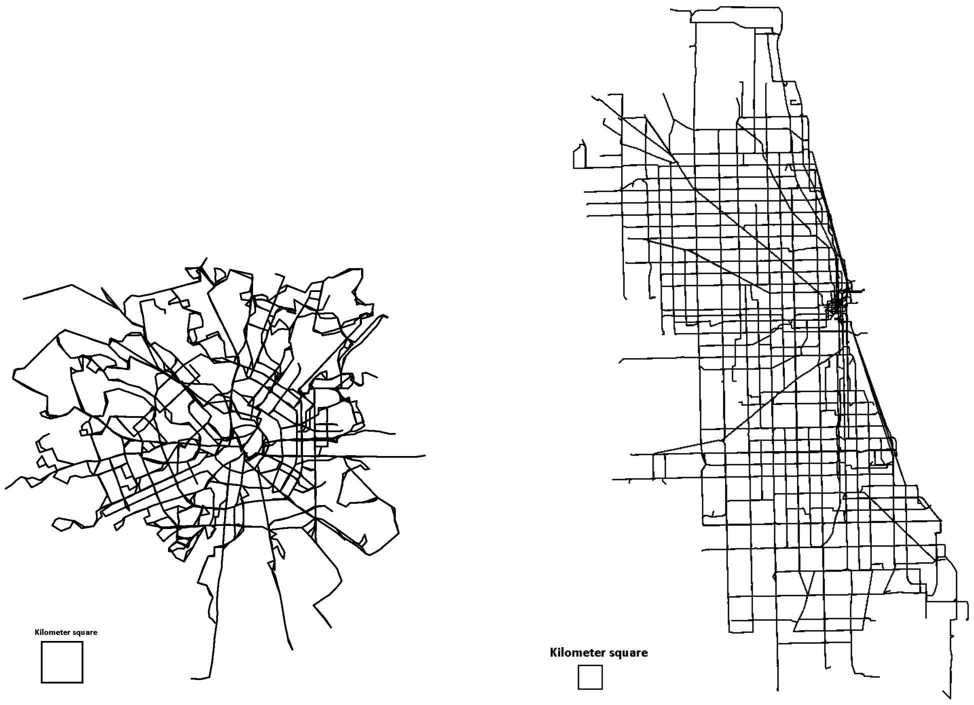
	Milan	**Chicago**
City size in Km^2	125	972
Number of lines	69	142
Line length (mean ± st. dev.) in Km	15.84 ± 5.76	24.48 ± 11.2
Line density	13.85	5.73

cities reviewed during the course of our research.

Milan, Italy represents our first case study. It is a medium size town (typical for many European cities), with a high bus density (69 bus lines) where routes wind around a circular city plan. The overall city structure is clearly not Manhattan-like due to the adaptation to the old Roman historical center and the progressive annexing of small peripheral towns in the main city body. With this kind of topology, crosses between bus lines occur at any time, there is no constant space between intersections, and streets do not run parallel one to each other for very long. In the aforementioned setting, with no free ways crossing the city, bus speed is generally low and foremost uniform.

The second city we consider is Chicago; chosen expressly to test scalability due to its immense size. Chicago has a great many bus lines (142) with routes running along the coast and then almost uniformly inland with a very low density. The PTS structure has a Manhattan-like plan, typical for many huge american cities. In this kind of settings we usually experience short contacts, mainly at intersections, at a low rate.

Figure 4.14 shows maps of the PTSs used in our simulations, while Tab. 4.4 presents a summary of the two cities and relative PTS layouts. In the table, bus line density is calculated as the mean number of kilometers covered by buses over a square kilometer.



(a) Milan, Italy (b) Chicago, Illinois

Figure 4.14: Graphical picture of public transportation systems used in the simulations.

4.2.2.3 Simulation Parameters

In our simulations each bus is equipped with an IEEE 802.11b network interface. Available bandwidth is 10 Mbps and radio range is 100 meters. Only line-of-sight contacts are considered; urban canyons created by buildings are taken into account.

Simulation starts at 4 A.M. (the first bus starts its trip at 5 A.M.) and ends at 8 A.M. of the following day (the last bus going out of service around 6 A.M.). All buses departing before 4 A.M. are considered as on duty during the night and not in the morning.

With regard to traffic generation, our aim is to simulate data exchange during an ordinary working day. Data traffic generation is a continuous process occurring during working hours: from 8 A.M. to 8 P.M. During simulations, requests are generated at a constant rate for every operating bus. Each bus receives 10 service requests per hour from users on board. These requests are accepted as long as the bus is in service, even when waiting for the next departure at the end of the line. For each request, the user specifies a destination line where she/he wants the response to be delivered. The destination line is chosen randomly on the PTS using a uniform distribution.

We assume that each bus is equipped with enough storage capacity, in line with modern equipment for industrial PCs. In order to simulate both typical web-based and generic data traffic (mail, forum posts, social network interaction, etc.), the size of request packets varies uniformly from 1 KB to 40 KB, whereas the size of response packets varies from 10 KB to 64 KB, thus representing classic emails, web pages, advertisement messages without heavy multimedia attachment so as to be able to actually provide useful services even with intermittent, opportunistic connectivity. When a request is generated, it is stored into the bus local storage until forwarding becomes possible.

When a bus reaches the end of the line, it may queue up and wait for another scheduled departure. If the bus stays in line it will hold all its data and will keep accepting requests from the surroundings. If, on the other hand, the bus leaves service, all content will be pushed to the first bus waiting in line. If there is no other bus around (i.e., there are no more scheduled departures) all the stored data, being them either requests or responses, are dropped and considered lost.

Forwarding can take place depending on the routing policy and distribution scheme between the base station on board to any other bus and between a base station and an IG deployed at each line end. In the following, we identify a routing policy as the set of rules used to identify a feasible next hop whereas a distribution scheme will be the set of rules to decide whether a packet will be forwarded. In our simulation we considered the following three distribution schemes that could be exploited by MDTN:

(i) *Pure Muling* (PM): each bus must keep all its requests until it reaches the end of line. When the end of line is reached, requests are satisfied and responses are immediately sent to the IG(s) of the destination line;

(ii) *Infrastructure-less Delivery* (ILD): requests are forwarded toward the destination line using a routing policy (later on). If, along the path between source and destination, an IG is encountered (i.e., the bus is in transmission range of an IG), requests are satisfied and changed into responses, which are forwarded toward their destination line adopting a specific routing scheme. Otherwise, requests are delivered to a destination line carrier, which will eventually meet an IG (e.g., at the end of the line), thus generating related responses;

(iii) *Infrastructure Aided Delivery* (IAD): requests are forwarded toward the destination line as in the previous case. When an end of the line is reached and requests are satisfied, responses are sent immediately to the IG(s) of the destination lines using a wired network.

With regard to the routing policy we need to test how MDTN can benefit from legacy or innovative routing approaches. For this reason in our simulations we compared three different routing algorithms that can be considered representative of a broader set of solutions; they are: Minimum hop (Min hop), MaxProp [13], and Op-HOP [73, 31].

Min hop follows a single-copy, link state routing approach, which should be able to exploit the specific design of a PTS, usually devised to bring people (as packets) across the metropolitan area minimizing the number of transits. Link state routing is extensively adopted in wired networks and we use it as a benchmark to understand how much MDTN can benefit from more sophisticated routing. Moreover, a link state approach is, in general, feasible for a PTS because all bus routes are known in advance and should not change without notice during the day.

MaxProp is considered the state of the art for DTN deployment over a PTS: it uses a multi-copy routing algorithm and implements a sophisticated buffer management based on messages priority. Being multi-copy, MaxProp uses more resources while generally providing good performances in terms of delivery delay.

The last protocol, Op-HOP, also uses a single-copy link state approach. Differently from the Min hop, Op-HOP adopts a metric based on probability: paths are calculated based on lines encounter probability to maximize packet delivery. Moreover, unlike other solutions, Op-HOP probability is estimated based on the number of encounters rather than their duration [73]. Op-HOP has already been shown to scale better than MaxProp in a purely ad hoc environment.

4.3 A Floating Data Network

Surprisingly, the more we expand our interconnecting possibilities, the more we are interested in local, context-related information. Think for instance of Facebook's success: even with the possibility to connect with anyone in the world, its main use is focused on our limited social context and people we already know. Another interesting example is represented by the revolutionary wave of demonstrations and protests that has recently flooded the Arab world, whose local coordination has massively relied on social networks such as Twitter.

In many scenarios data has temporal and spatial validity depending on their attributes or use. Once retrieved, data might be disseminated further, that is, a mechanism might be required to guarantee data availability (survivability) in the *Area of Interest* (AoI). The diameter of this network of data grows and shrinks and then vanishes meeting the user's demands. As an example, any instant advertising of goods, attractions, events and/or news can be consumed locally, disseminated and kept alive in a certain AoI, with data moving back and forth, from user to user: hence the name *floating data*.

Through the rest of this section we will discuss the proposed solution aimed at providing some guarantees for data availability in the anchor area. To this end, we start in Sec. 4.3.1 by further detailing the scenario under investigation and provide some use-case scenarios where it could be adopted. Following, we delve into the details of our proposal and its rationale. Concluding, in Sec 4.3.4 we discuss the simulation platform and the evaluation strategy. Finally, the experimentation outcome is shown and discussed later on through Sec. 5.3.

4.3.1 Scenario Definition and Use-Cases

The scenario under consideration is comprised of mobile entities producing and consuming data. The data is supplied and maintained by the users themselves and has geographical and temporal validity. Nodes in the anchor zone, exchange information in opportunistic fashion when they are in transmission range with each other. When leaving the anchor zone, nodes can discard the data acquired in the area of interest. In these settings, there are no guarantees of information availability due to the unpredictable nature of human mobility. However, we could augment the chances of data surviving in the AoI if a certain criteria is met. This criteria is bound to the node population available and to the mobility characteristics of nodes in the anchor area.

The concept of data anchoring can be applied in different kinds of scenarios, ranging from urban security, emergency to infotainment. An urban security application can be deployed in environments such as dense urban areas where each of us carries a smartphone running a service that provides data anchoring. Each user might disseminate different kinds of information concerning warnings or advices for the community; Blind persons could be interested to acquire information regarding holes in the road as well as an elderly can appreciate information regarding presence of obstacles on their walk. By far, the most appealing use-case scenario is that of employing such a solution toward the goal of people coordination in free-speech manifestations or uprising scenarios.

In this envisioned scenario, data is supplied and controlled by the people, in contraposition of todays infrastructure data model where data is made available globally and stays on forever. From this basis, we identify as the characteristics of data anchoring technique the survivability and accessibility of the data. Survivability means that the information is somewhere near the anchor zone while accessibility refers to the costs needed to access the information stored on nearby devices. In our work we focus on the survivability requirement. However, we also show how our proposal could be employed to facilitate data accessibility by controlling the spatial distribution of the data.

4.3.2 System Modus Operandi

The goal of AirCache (AC) is to keep alive specific content in determined locations until some conditions are met. The data might vanish as a consequence of node mobility and/or low node density or it might even expire after a while (e.g., the time attribute attached on the metadata has expired). The data might be produced locally or brought from elsewhere. When data needs to be cached in a specific location, a node responsible for that data (e.g., the producer itself or a node which is carrying this responsibility) searches around for an existing AC to which it can hook the data or, if no AC is present in the area, it creates a new one. Considering the case when an AC already exists, it is crucial to replicate the content so as to have it persisting in the area regardless of node swarming.

To achieve this, a minimum number (*Min*) of replicas should be always present in the system. This threshold can be achieved only in case the number of nodes required to fullfill this criteria is present is the anchroing zone. A good choice is to provide also an upper threshold number (*Max*) of replicas so as to limit the memory occupation and transmission costs among all the nodes in the area. More in detail, the system is designed so as to have an average (*Avg*) of replicas in the area and exploit hysteresis to decide when replicas have to be augmented ($Avg < Min$) or diminished ($Avg \geq Max$) to reach again a cardinality equal to *Avg*. In such a way a balance of replicas between the upper and the lower number are guaranteed to be found in the AC.

On the contrary, one could argue that a node once is in the AoI, it could replicate the data to all surrounding nodes if any available. However due to movement characteristics of nodes and resource constrained nature of the later, this indiscriminate/epidemic replication of data might not lead to an optimal performance. In our solution, the original node carrying the data broadcasts a message with the data inside. All nodes in transmission range receive the data, but only few eligible nodes memorize it. To this end, a criterion is needed to evaluate which nodes will or will not cache the data. To achieve this we could exploit sensed information about a nodes energy, mobility and memory space. Indeed, the nodes comprising the system periodically share with their neighboring nodes information with each other. Considering a network of this kind, formed by spontaneous aggregations of nodes and bereft of infrastructure or any central entity, a decentralized distributed algorithm establishing the grounds for node cooperation is required.

To not reinvent the wheel, we delved into scientific literature and found an interesting solution that could be adapted to enable the depicted scenario. The solution that inspired us is the Reliable R-Aloha (RR-ALOHA [11]) protocol, a distributed reservation protocol that uses a *Frame In-*

formation (FI) structure to avoid collisions between nodes, establishing the grounds for node cooperation. Through the rest of this section we detail our proposal built on top of the RR-Aloha protocol, discussing the added features and overall system *modus operandi*.

4.3.2.1 Joining the AC and Data Replication Policy

The RR-Aloha is a distributed MAC layer protocol implementing a reliable single-hop broadcast channel among all neighboring nodes proposed in the context of vehicular networks. This protocol is interesting as it does not suffer from the *hidden terminal problem*, as well as its distributed way of assigning slots to the nodes. In essence the protocol provides the means to build a TDMA (sloted) channel guaranteed to be accessed in mutual exclusion between nodes. Though the proposed protocol is particularly apt to the slotted physical layers it could be ported on top of any physical layer, in particular on top of the 802.11 stack.

When a node reaches an area of interest, a geographical location where data should be anchored, it enters the listening mode so as to verify the possible existence of a local AC. Each node has to wait for a whole Frame Information Time (FIT) before joining. Listening an entire FI length reduces the probability of collisions, augmenting the information that nodes trying to join acquire of the AC (if one already exists). In the current implementation the FI length is fixed and a system configuration parameter whose default value is set to 100. This means that there is a bound on the number of nodes that can contend the slots of the FI and by design the FI is shared by nodes found in 2-hop distance by each other. Following the original design we limit the number N of nodes contending for a FI to 50 while the rest of the slots inside the FI can be used for additional bandwidth reservation (later on).

After a listening period, if the node has not acquired any information and has data contents that require anchoring, it proceeds with the creation of the AC, otherwise it enters a contention period. After the contention period ends, the active node is able to broadcast its view of the surroundings along with its chosen BCH appended at the *end* of the FI. If the node does not have any data content to anchor and there is no local AC nearby it is considered as parked. Nodes in parking mode are not considered as part of the system but are eligible candidates at a later moment. Another situation where nodes enter into parking mode occurs when the FI does not have any additional free slots. This entire process is also evidenced by means of as state diagram depicted in Fig. 4.15.

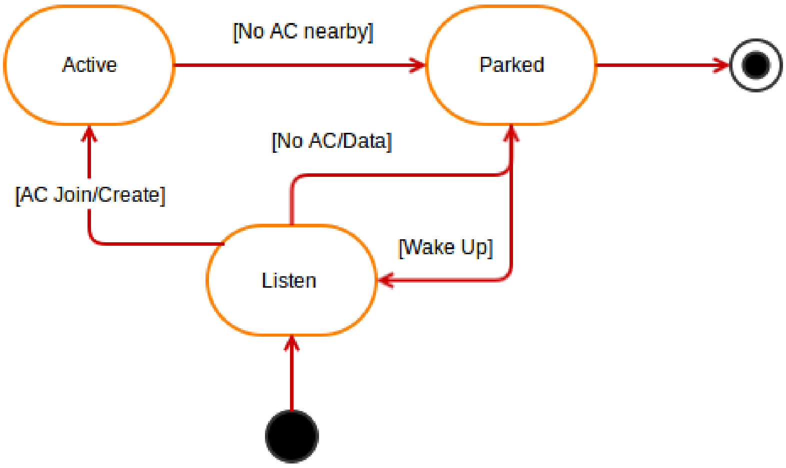


Figure 4.15: Node state transition diagram.

After a FIT each node has the available information about the data replication level inside a cluster through the sniffed control data packets (refer to 4.3.2.2). If the replica level is less than a preconfigured threshold ($Avg \leq Min$) and the system has the capabilities to further replicate the data, a replication event is triggered. The most capable node having a replica of the data content broadcast the data and the best elected node(s) are eligible to cache it. The election of the sender node is based on the battery index (↑) whereas the receiver node(s) are elected based on the buffer occupancy index (↓). If more than one replica is present in the cluster, the next candidate(s) register the content broadcast event and postpone their intervention in the next FI. If the event did not have any effect on the number of replicas the process repeats.

On the other side, when the maximum threshold for a given data content is exceeded ($Avg \geq Max$), the redundant replicas can be discarded. The node(s) eligible in this case are those with the lowest battery and buffer occupancy index and the resulting index is computed giving equal weight to both indexes.

4.3.2.2 Slot Reservation Mechanism

Before delving into the characteristic features of the devised protocol we provide a formal definition of cluster so as to better comprehend the terminology of the original proposal:

Definition: Cluster(C) = {Node(A) $\wedge$ Node(B) | Node(A) $\neq$ Node(B) $\wedge$ Node(A) $\in$ C $\wedge$ Node(B) $\in$ C $\Leftrightarrow$ Node(A) *hears* Node(B) $\wedge$ Node(B) *hears* Node(A) }

That is all nodes participating in the same cluster are nodes whom can all hear and be heard by other nodes on the same cluster. In other words clusters are strongly connected components of the graph in a certain temporal instant. When in dense environments and in presence of continous, periodic traffic exhanged among nodes, this scheme helps mitigate the well-known hidden terminal problem not completely solved by the 802.11 Request To Send/Clear To Send (RTS/CTS) MAC feature. At the same time, such a scheme provides the grounds for node cooperation in a disributed environment and is exploited by our protocol to augment the chances of data survivability in the anchor area.

The original protocol employs a sliding frame mechanism whereby each node autonomously verifies the status of the slots comprising the FI based on previous transmitted information from its single-hop neighboring nodes (cluster). Each node participates in a periodic contention for its slot (every N slots), each turn confirming its presence, broadcasting its view of the FI occupation based on its local overheard information. We employ a slightly different algorithm for processing the FI slot status occupancy as detailed by algorithm Alg. 2 and exemplified in Fig. 4.16.

In the original proposed algorithm each node uses only partial information of the overheard FIs, those parts of information falling inside the sliding frame of N slots. Instead, we exploit each nodes local view of its neighborhood in its fullness. The rationale of this design choice stands in the difference between the different mobility dynamics of vehicular and human comprised networks, the former being more dynamic, requiring the algorithm to be more reactive for slot status computation. Our *modus operandi* allows us to have a complete picture of the slot reservation status, information which is not propagated but is instead used locally to reserve additional bandthwidth whenever needed (later on). In this way, we are considering information which is at most *2*FIT* old. However, taking into account the human mobility in most scenarios and the 802.11 transmission range, this amount of time is neglectable.

In order to implement the aforementioned features, we have devised the required application level data structures as depicted in Fig. 4.17. The frame information is a data structure comprised of basic channels (BCHs). The *ReservationBoard* used in Alg. 2 has the same structure as the, instead its contents denote the slot status occupancy in a 2-hop coverage area, whereas the *FrameInformation* is the transmitted FI denoting the slot occupancy status of the 1-hop neghborhood. BCH on its turn is also a data structure made of the following fields:

1. An integer holding the node identifier that has reserved the BCH, otherwise -1 is the default value;
2. A flag denoting the slot occupancy status, set as true if the BCH is *Free* or false if it is *Reserved* by a node;

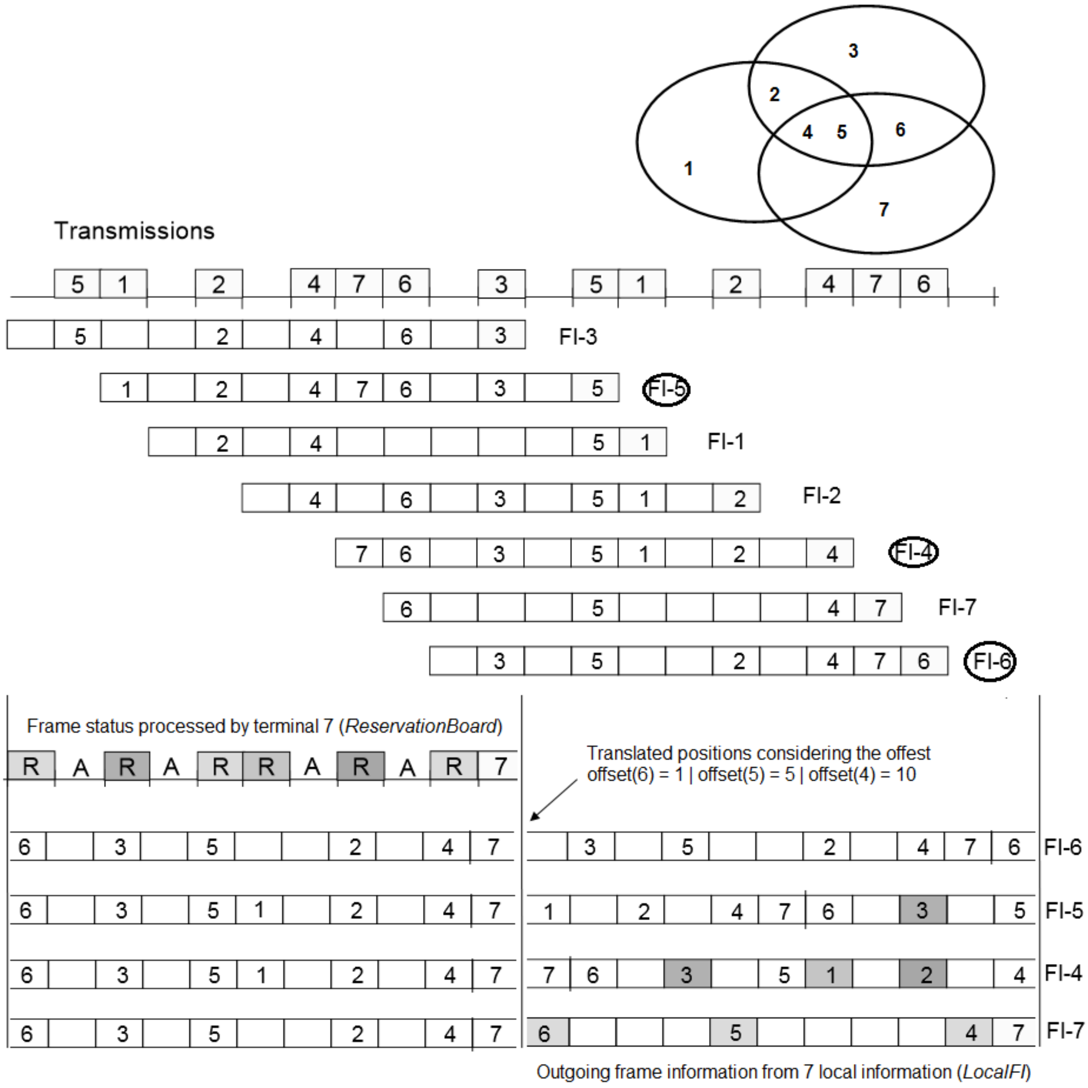


Figure 4.16: Slot occupancy computation. In the first part is shown the FI composition from each node perspective while in the second part are shown the steps involved in the computation of the ReservationBoard and of the outgoing FI from node 7 perspective.

Algorithm 2: Slot status computation.

```
input  : FrameInformation ($LocalFI$), ReservationBoard ($ReservationBoard$),
         Received FrameInformation(s) ($ReceivedFIs$), Local Node Identifier
         ($NodeId$)
output: Updated $ReservationBoard$ and $FrameInformation$
bool Collision ← False, Owned ← False ;
int Position ← -1; BCH Channel ← ∅;
for $i \leftarrow 1$ to Length(LocalFI) do
    BCH Heard ← BasicChannel(LocalFI, i) ;
    foreach $fi$ in ReceivedFIs do
        Position ← Shift($fi$, i) ;
        Channel ← BasicChannel($fi$, position) ;
        Collision ← False ;
        Owned ← Owner(Channel) ;
        if Owned and NotSameId(Channel, Heard) then
            Collision ← True;
            break ;
        end
    end
    if Collision then
        UpdateBoard($ReservationBoard$, i, Free) ;
        UpdateFI($LocalFI$, i, Free) ;
        if $NodeId$ = Id(Heard) then
            Collision(True) ;
            EnterContention() ;
        end
    else
        if (AdditionalAndExpired(Channel)) then
            UpdateBoard(board, i, Free) ;
        else
            UpdateBoard(board, i, Reserved) ;
        end
    end
end
```

3. A flag denoting if the BCH is *Owned* by the node or false if it is an *Additional* slot reserved by the nodes current needs.

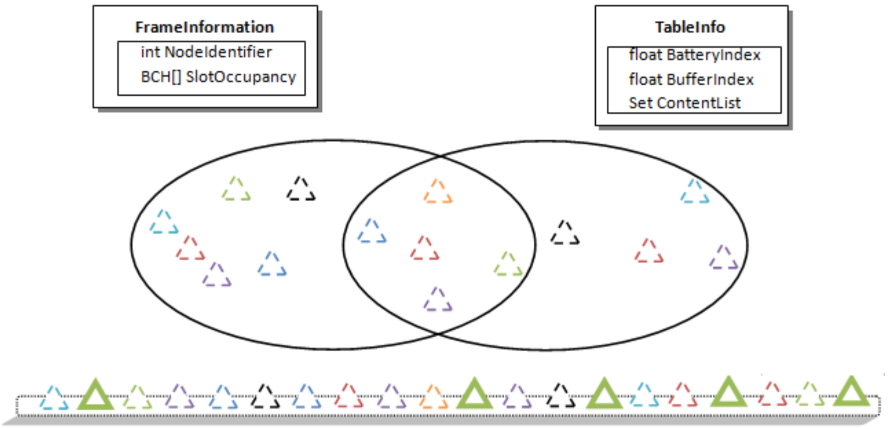


Figure 4.17: Devised data structures and respective data packets transmitted during the BCH. Slots evidenced in green (full) triangles denote free slots in the system.

The reason for holding a 2-hop slot occupancy status in the ReservationBoard is so as to implement the slot reservation mechanism whereby nodes reserve additional transmission bandwidth whenever the data to be transmitted does not fit inside a single BCH. In the current implementation, whenever required, each node unilateraly decides and reserves the additional slots after consulting its local reservation board, notifying the intention in his outgoing FI. The additional slot reservations expire as soon as they are not needed anymore and this control is contemplated through lines 23-26 of the algorithm Alg. 2. This information is broadcasted once and only by the node requiring the slots.

With reference to Fig.4.17, *InfoTable* is yet another data structure holding the battery percentage, buffer occupation level and the list of data content identifiers cached by the node. This information is broadcasted by each node in its reserved BCH in addition to the FI transmission and stands at the basis of the replication policy enforced by the protocol in place.

4.3.2.3 Control of Data Spatial Distribution

While data content survivability in the AoI was identified as a key feature in our envisioned scenario, data acessibility if not of less importance. Depending on the entry point of the data in the AoI it can happen that data gets confined and replicated only inside a single cluster e.g, a limited portion of the AoI. This happens because nodes start enforcing the replication policy only after having heard about the existence of the data. While this event can happen in our system, e.g., a node found in a cluster interesection area has the data and notifies its existence, it is accidental. Another possible situation of data being replicated elsewhere, except the initial entry point, is to be attributed to node mobility in the anchor area. Indeed a node caching the data might change its location while still reside inside the AoI and successively trigger a data replication event in the current cluster it resides on.

Hence, we deem necessary to provide a mechanism aimed at controlling the spatial distribution of the data. The rationale is that a more homogeneous data distribution reduces data access costs, easing data availability in the AoI. The responsibility to implement this feature is upon nodes found in the interesections between clusters. Each node after a FIT, autonomously identifies if it is positioned in a cluster interesection. This involves partitioning the set of confirmed slots in the outgoing FI into disjoint cluster as detailed by the algorithm Alg. 3.

After checking the slot occupancy status as detailed before in Alg. 2, nodes autonomously verify their position in their neighborhood so as to recognize if their are positioned at cluster intersections. The neighborhood connectivity is reconstructed exploiting the received FIs from other nodes. The algorithm starts by first filtering the heard local neighborhood nodes of X, checking in their respective sniffed FIs whether they have an entry with a slot assigned to X (line 1).

Algorithm 3: Cluster set computation.

```
input : FrameInformation ($LocalFI$), Set ($ReceivedFIs$)
        Set ($ConnectedComponents$)
output: Set of connected components $ConnectedComponents$
Set Connected ← Bidirectional($LocalFI$, ReceivedFIs) ;
Cluster P ;
P ← P ∪ FirstElement(Connected) ;
ConnectedComponents ← ConnectedComponents ∪ P ;
foreach Slot in Connected do
    bool InsideAnyCluster ← False ;
    foreach C in ConnectedComponents do
        if IsMember(NodeId(Slot), C, ReceivedFIs) then
            P ← P ∪ NodeId(Slot) ;
            InsideAnyCluster ← True ;
            break;
        end
    end
    if not InsideAnyCluster then
        Cluster P_New ;
        P_New ← P_New ∪ NodeId(Slot) ;
        ConnectedComponents ← ConnectedComponents ∪ P_New ;
    end
end
```

This means that the connection is bidirectional, both nodes hear each other and are both part of a cluster. After the filtering process, the algorithm starts partitioning the set of filtered nodes into disjoint clusters by enforcing the cluster property as defined above (line 8).

It is noteworthy to point out, that the computed clusters might not be complete or disjoint. Our algorithm does not account for common elements intentionally (line 11) and the incomplete information comes from the fact that each nodes uses only its local view rather than a global view of the network. The purpose for this is related to the replication policy enforced by nodes found in cluster intersections which we detail below.

Before transmission time and after having processed the sniffed slot occupancy status, each node verifies its position in his neighborhood according to the algorithm above. Nodes finding themselves in at most one connected cluster react as explained through Sec. 4.3.2.1. A different behavior is assumed by nodes finding themselves at cluster interesections, that is, nodes that participate in more than one connected cluster. These nodes have the responsibility at enforcing a replication policy on a per cluster basis, pulling the replicated data from one point (e.g., initial entry point) and causing its replication elsewhere.

To implement this feature, the protocol provides another data packet which we refer to as a *pull* data packet. This packet is broadcasted whenever the data replication level in a certain connected component is under a specified threshold and the node does not posses a replica of the data itself. The replication policy is enforced taking into account the number of nodes available in that particular cluster, that is, the minimum threshold is computed as the $Min(Min_{thresh},\ card(C_i))$. For this process to complete, causing the replication of the data in other connected components, a FIT has to pass. In the next period, if data is still under the replication level, a replication event is triggered causing data replication in other components.

4.3.3 Discussion and Current Limitations

A factual deployment of this protocol requires accommodating new nodes and data alimenting the system. However, as discussed above, the number of slots constituting the FI is fixed and decided *a priori*. Yet, another aspect of this *modus operandi* is related to the size of the FI: a high number of slots constituting the FI might hinder nodes reaction times. Symmetrically, having an insufficient number of slots might block new nodes from joining the system. To address this issue, additional changes have been proposed to the original protocol, introducing the possibility to dynamically adapt the frame information length [58]. The authors of this work, propose a FI adaptation algorithm which augments or shortens the length of the FI depending on its level of fragmentation. This metric is measured by exploiting the available information of free and busy slots. Our current implementation does not contemplate such a feature and we deem it as an interesting future work diserving further investigation.

Another practical implication of the protocol is related to the need to periodically reserve the communication slot. This recurrent process is payed in terms of energy consumption which is a precious resource for handheld portable devices. Additional logic could be put in place to save energy, so as for instance by alternating nodes from active to parked mode and viceversa according to certain criterias (e.g., data replication level is inside the defined thresholds).

Yet, another issue not handled in the current implementation, is presented by the scenario when more than one AC come to interfere with each other. This phenomenon occurs when nodes participating in different ACs come to communication range of each other. This interference is due to potentially different time synchronizations of nodes participating in the different ACs. Even worse, it might happen that a node caching data from AC-1 moves outside the reach area of AC-1 but has still some valid data to be cached inside an AoI comprising nodes from AC-1. This will trigger the node to create another AC oppened to other nodes to join in. Eventually the two ACs might end up interfering with each other.

This is an issue which we leave as a future work but in here we provide some insights as to how it might be addressed. Our protocol is able to distinguish and solve the hidden terminal problem whenever two nodes from different ACs talk inside the same basic channel. Simply letting the protocol do its job, forcing nodes to autonomously recontend for other available slots, does not solve the problem but rather poses it elsewhere for other contended slots. However, the presented situation is exacerbated when transmissions from nodes participating in different ACs interfere with each other.

The solution to both these sub-problems might be to add an identifier (e.g., a timestamp of creation) to each autonomous AC which is propagated within the FI. This way nodes hearing transmissions originating from a different AC as the one they are in, become temporarily parked. Inteference between nearby communications is solved by nodes entering a re-contention period for slots in their respective ACs. In these way after less then a FIT respective nodes can identify the presence of the other AC. Once interfering nodes are parked an AC merger procedure could be put in place to handle the gradual merger (sync.) of conflicting ACs with one another or nodes might even stay parked until one AC solely is present in their surroundings. We deem this issue an important one, and how this interacts with a dynamic frame adaptation algorithm deserves further investigation. The experimentation part of this scenario is orchestrated in such a way so as to avoid this problem ocurring (refer to 4.3.4).

4.3.4 Simulation Environment and Evaluation Strategy

The targeted simulation environment used to implement and evalute the proposed protocol is the Network Simulator 3 (NS3 [85]). The need to

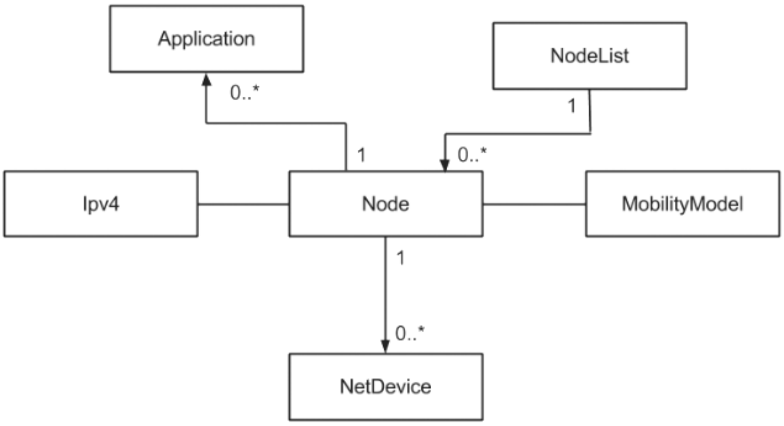


Figure 4.18: Node class diagramm.

adopt a different simulation environment was because the NS3, opposed to the previous adopted simulation environments, is capable of simulating common features of the 802.11 Phy/MAC. Hence, it provides a more realistic simulation environment for this targeted scenario. Before discussing on how the experimentation is orchestrated, we now briefly survey the programming components of the software environment under scrutiny, oulining some design choices of our implementation.

Depicted in Fig. 4.18 is the *Node*, the basic computing device abstraction, providing the necessary methods to manage the representations of computing devices (nodes) during the simulation. It represents a device having communication capabilities through the presence of one or many network interfaces. In our experimentation each nodes is equipped with a wireless 802.11g interface. We adopt a flat addressing scheme where each node in the simulation is assigned a unique IPv4 address. This is done for simulation purposes so as to attribute unique identifiers to nodes whereas in a real deployment address assignment in a dynamic, decentralized environment is still an open research issue; Level-2, MAC addresses are used in practice.

1. **Channel and NetDevice**: Represent the wireless channel where

the data flows and the network device respectively. In our experimentation we adopt the WiFi channel specification by [52] supported on the NS3 platform. Below is shown a code snippet providing the necessary configuration parameters regarding the communication aspect between nodes in our simulations;

2. **Mobility Model**: To mimic human behavior in aggregation areas, we chose to evaluate our proposal using the Random Way Point (RWP) mobility model, generating the contact traces through the BonnMotion tool [5]. More details on how the traces are generated and the actual parameters used to generate them are provided later on the experimentation part of this scenario (refer to Sec. 5.3);

3. **Application**: Is the application running on a node where we have implemented the logic described throughout this section;

4. **Energy Model**: Not evidenced in the figure above, however, this component is used and attached to a node in order to account for the energy dispenditure of wireless transmission/reception costs. In this way, we can account for a dynamic energy resource even when computing the sender/receiver nodes whenever a replication process is triggered.

Our aim is to show that the proposed protocol does indeed provide some guarantees of data survivability in the anchor area. Hence, in our experimentation we study the data TTL, that is the time interval elapsed from data being brought into the AC and the time it ceases to exist. Another feature we deemed interesting to provide through our protocol, is the control of the spatial data distribution inside the AC. The rationale was to enforce a more homogeneous distribution of the data, easing data availability in the AoI. To this end, we have performed different simulations with varying node population and mobility characteristics which are reported through Sec. 5.3 evidencing that our proposal does indeed serve its purpose.

```
Config::SetDefault ("ns3::WifiRemoteStationManager::Frag-
  mentationThreshold",
  StringValue ("2200"));
Config::SetDefault ("ns3::WifiRemoteStationManager::RtsCts-
  Threshold",
  StringValue ("2200"));
std::string phyMode ("OfdmRate54Mbps");
Config::SetDefault ("ns3::WifiRemoteStationManager::Non-
  UnicastMode",
  StringValue (phyMode));

YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default ();
YansWifiChannelHelper wifiChannel;
wifiChannel.AddPropagationLoss ("ns3::RangePropagation-
        LossModel",
        "MaxRange",
        DoubleValue (100.0));
wifiPhy.SetChannel (wifiChannel.Create ());
WifiHelper wifi = WifiHelper::Default ();
wifi.SetStandard (WIFI_PHY_STANDARD_80211g);
wifi.SetRemoteStationManager ("ns3::MinstrelWifiManager");

NqosWifiMacHelper wifiMac = NqosWifiMacHelper::Default ();
wifiMac.SetType("ns3::AdhocWifiMac");
```

Listing 4.1: Wireless medium configuration.

Few considerations are due before concluding this section. As stated in Sec. 4.3.2, our protocol does not contemplate the scenario of conflicting ACs. In our experimentation we avoid this case by considering a single AoI, the same for all data contents present in our simulation environment. We also force nodes in the simulation area to to only join and not create the AC, while the node creating (bootstraping) the AC is chosen randomly at the begining of the simulation time from the entire node population.

5

Results

In this chapter we evaluate the performance of the devised solutions and discuss their outcomes. In particular, in Sec. 5.1, we discuss the evaluation strategy and outcome of the content sharing solution detailed through Sec. 4.1. First, we start by evaluating a single-hop delegation forwarding strategy, that is the DTN forward chain is comprised of only one servant and later on extend the study by showing the feasibility of a multi-hop DTN chain. Concluding, we argue about the experimentation outcome and the draw the conclusions regarding this scenario. Following, in Sec. 5.2 we study the feasibility of MDTN, a delay tolerant solution employing the public transportation system as a routing backbone for elastic, non-real time service delivery. In this context, we study the performance trend of an urban-wide deployment architecture under realistic topological data taken from Milan, Italy and Chicago, Illinois. Concluding this section, is a discussion about the delivery profiles of our platform and the tradeoffs that arise from state-of-the-art routing algorithms. Finally, in Sec. 5.3, we provide the experimentation outcome of AirCache, a network of floating data, discussed and detailed through Sec. 4.3.

5.1 Mobile-to-Mobile Content Sharing Solution

In this section we show the experimentation part of the devised content sharing solution, explained and detailed in Sec. 4.1, with emphasis on the delegation forwarding strategy. We start by showing the evaluation of our solution under the map-based mobility model in the single hop delegation scenario, that is the DTN chain is composed of a single hop. Next, in Sec. 5.1.2 we study the multihop delegation forwarding scenario where we provide some statistics regarding the tradeoffs that emerge. Finally in Sec. 5.1.3, we discuss the evaluation of the named-data approach under the Haggle contact trace detailed in Sec. 4.1.5.2. To avoid confusion, we point out that both system designs were evaluated under both mobility models, however, their outcome has a similar trend and therefore we present only one set of results. Intuitively, one could expect NDN outperforming the connection-oriented design, due to its behavior of caching data-caching the request path, but the mobility traces under consideration did not evidence this benefit. A possible explanation to this outcome might reside on the nature of the mobility traces under consideration and the sparse connected environment they present. This is an issue we need to further investigate and consider it as a future work.

Concluding, while in the real data trace the movement is fixed, in the map-based one we repeat each simulation scenarios several times in order to achieve more accurate results, independent of the initial positioning of the consumer node in search for that particular data content and independent of the initial positioning of the content in the simulation world. Each scenario is run using different random seeds to initialize the movement model and for every seed the simulation is repeated using the compared protocols.

5.1.1 Single-hop Delegation Efficiency

In this analysis we evaluate the efficiency of the delegation forwarding scheme. Table 5.1 shows the settings used for these set of simulations, comparing M2MShare against the following schemes:

1. **No_delegation**: strategy not employing the delegation forwarding strategy, representing the scenario where content retrieval ocurrs when a producer is found in the connected portion of the network with the consumer;
2. **Delegation_to_all**: strategy where consumer(s) delegate, eligible for delegation tasks to all encountered nodes in the network. It represents an epidemic approach to data dissemination.

Table 5.1: Simulations settings for evaluation of the delegation forwarding scheme.

Population	1000
Content size	3.0 MB
Content popularity	5%
Content distribution	Uniformly distributed
Content Division Strategy	M2MShare
Beaconing Frequency	5 min
Delegation TTL	Probation Window (P_w)
Nr. of simulation runs	40 x 3
Simulated time	7 days

The key performance metrics we are interested in are as follows:

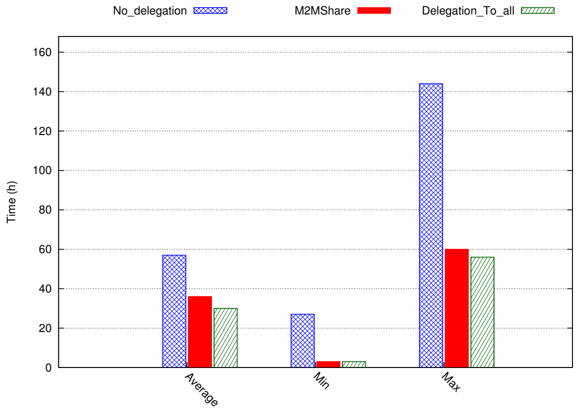
1. *Found time* (F_t): denotes the time interval between the first delegation is issued and the time the data content is retrieved;
2. *Number of delegations employed*: representing the number of tasks used by a consumer to find the required data content;

3. *Return Index*: the number of returned tasks over the total number of delegations employed by the consumer for that particular data content. This index is complementary to the above, quantifying the soundness of the devised mechanism;

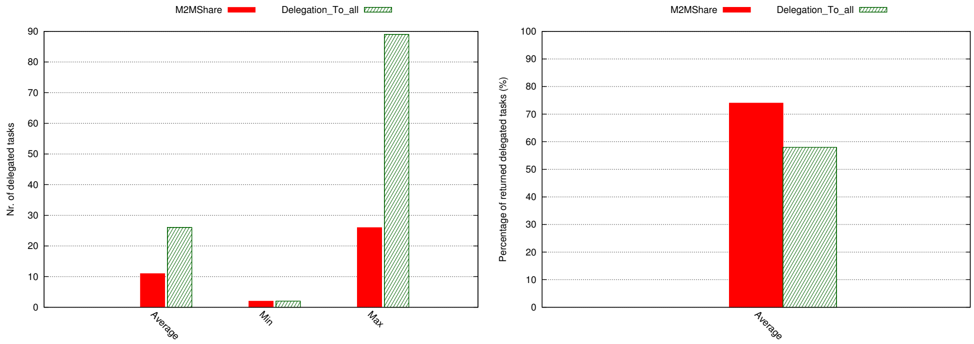
4. *Redundancy*: refers to the quantity of data transfer exchanged during the simulation for a particular consumer request. This is an index of the overall network redundancy introduced in order to retrieve the required content.

In Fig. 5.1a it is possible to see the advantage, in terms of found time, in using the delegation technique instead of not using it. The two systems employing delegations find the required content in less time in each simulation run at the expense of higher overhead in terms of bandwidth due to delegations. The system employing delegations to all encountered nodes obtains a better result on average, but at the cost of a higher number of delegated tasks (Figure 5.1b). A higher number of delegated tasks imply the used of more resources translated in terms of storage and energy-costs.

Figure 5.1b makes a comparison between the two systems employing delegations by showing the number of overall employed task delegations till content is retrieved or simulation time expires. It is easy to see that M2MShare uses fewer delegations while achieving a higher percentage of completed delegated tasks (Fig. 5.1c). This outcome is due to a conservative delegation strategy employed by M2MShare in delegating unsatisfied, unaccomplished tasks only to frequently encountered nodes (servants). Since we do not have any means for evaluating the ability of one servant to satisfy a content request what we do is delegate to the encountered peers that can be expected to be encountered again in the future. The Delegation_to_all strategy also contributes to higher overhead due to completed tasks, ready to be returned towards the consumer



(a) Time employed to retrieve the content



(b) Delegations employed

(c) Return index

Figure 5.1: Comparison between the different schemes.

that unfortunately expire along the path and are discarded before having the chance of encountering the consumer.

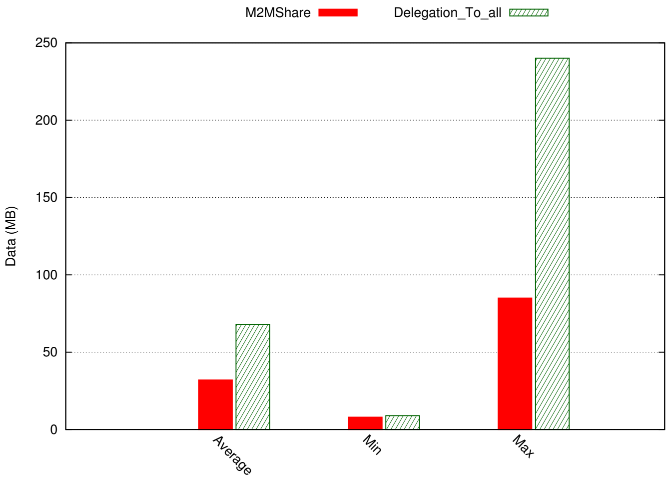


Figure 5.2: Average, min, max transferred data amount employed in each delegations technique.

Fig. 5.2 shows the comparison between M2MShare and Delegation_to_all in terms of transferred data quantities until content download or simulation time ends. It is straightforward to notice the higher overhead in terms of data transmissions introduced by delegating to all encountered peers whether M2MShare reduces the exchanged data quantity while still achieving the goal of acquiring the requested content. From the above results it is obvious that the delegation strategy serves its purpose by extending a peer reach area to other mobile disconnected networks where data content might be available therefore reducing the found time of a desired content. Although this strategy introduces an overhead in terms of bandwidth usage, computation and power consumption we control these side effects by delegating only to frequently encountered peers whom are expected to be encountered again in the future.

5.1.1.1 Content Division Strategy

In previous analysis we evaluated the efficiency of M2MShare delegation technique, but in the protocol the delegation forwarding mechanism is not the only new innovation described. M2MShare provides a new content division strategy, described in Section 4.1.3.5, where a content can be retrieved in pieces and a piece size varies. The content division strategy might add redundancy during data transfer as it can happen that overlapping data intervals are simultaneously retrieved by different servants. However, the fact that each servant is asked to retrieve the data content starting from different points allows reconstructing the whole data content even if both retrieved only parts of it.

Table 5.2: Simulations settings for evaluation of content division strategy efficiency.

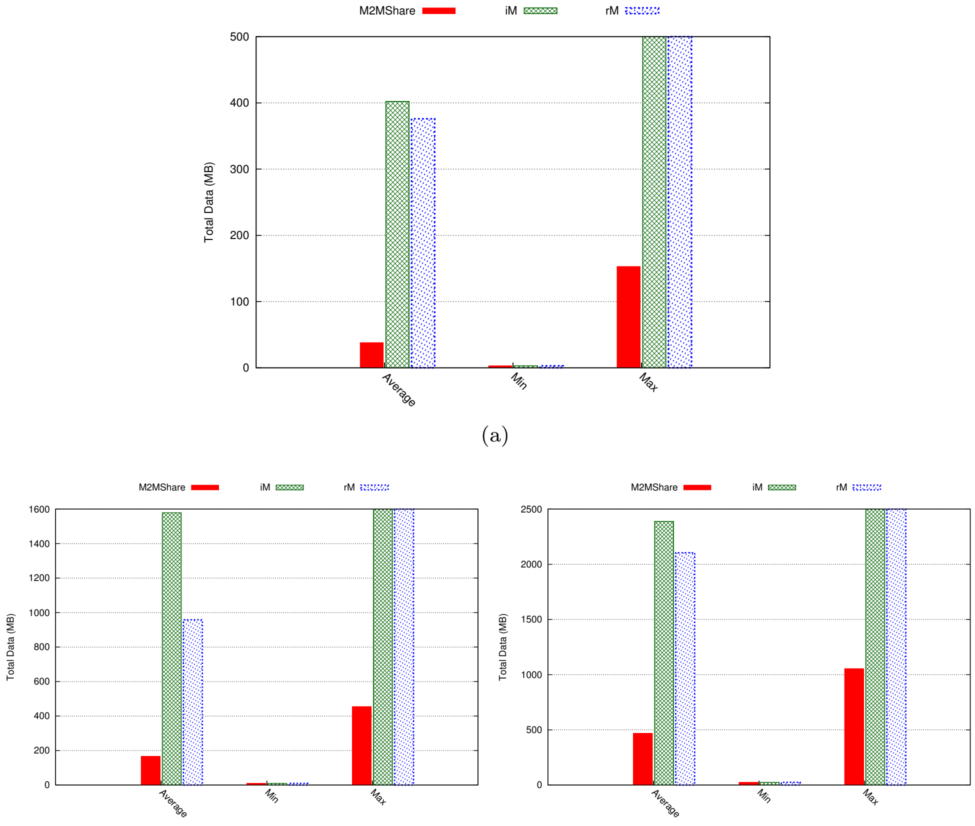
Population	1000
Content size	3.0 MB, 10.0 MB, 25.0 MB
Content popularity	5%
Content distribution	Uniformly distributed
Delegation type	M2MShare
Delegation depth	1
Content Division Strategy	M2MShare, iM, rM
Nr. of simulations	40 x 3 x 3
Simulated time	7 days

As shown in Tab. 5.2 we compare our content division strategy with two other division strategies:

- iM: a strategy where delegations are issued at each servant for the entire data content.
- rM: a strategy that randomly chooses the initial transfer point in the content.

We considered the average, min and max transferred data amount employed in each content division strategy. We repeated the simulations

40 times, changing random seeds, for every content division strategy in order to achieve more accurate results.



(a)

(b)

(c)

Figure 5.3: Comparison of the different content division schemes for content size: (a) 3.0 MB (b) 10.0 MB and (c) 25.0 MB.

In Fig. 5.3a, Fig. 5.3b and Fig. 5.3c we can see that our division strategy has the least redundancy during data transfer, especially considering big-sized data contents.

5.1.1.2 Variable Content Popularity

In previous analysis we have shown the advantage, in terms of found time of the searched content, using M2MShare delegation strategy against avoiding using delegation or delegating unaccomplished tasks to every node met. The confrontation was made using a constant number of copies of the searched data content uniformly distributed between all nodes in the simulation, i.e. 50 copies. The current analysis wants to show the difference in performance using the three delegation strategies changing the initial content popularity of the searched content. To this aim, in these simulations we change the Content popularity (F_p) of the data content keeping constant the number of total nodes (N). Simulations settings are shown in Tab. 5.3.

Table 5.3: Simulations settings for evaluation of delegation efficiency with varying content popularity.

Population	1000
Content size	3.0 MB
Content popularity	5%, to 40%, 5% step
Content distribution	Uniformly distributed
Comparison with	No_delegation, Delegation_to_all
Delegation depth	1
Content Division Strategy	M2MShare
Nr. of simulations	40 x 8 x 3
Simulated time	7 days

We change the Content Popularity (F_p) value, from 5% (50 copies) to 40% (400 copies). When the F_p is low (with $F_p \geq 5\%$) the system which not employing delegation is not able to find any piece of the data content during the entire simulation time. We have indicated this in the chart by assigning to Ft_{avg} a value of 48 h. With higher values of F_p, the first protocol is able to find the content thanks to the higher popularity of the requested content, but it employs more time than M2MShare and Delegation_to_all. Finally, with the highest values of F_p the performances of the three compared systems becomes very similar.

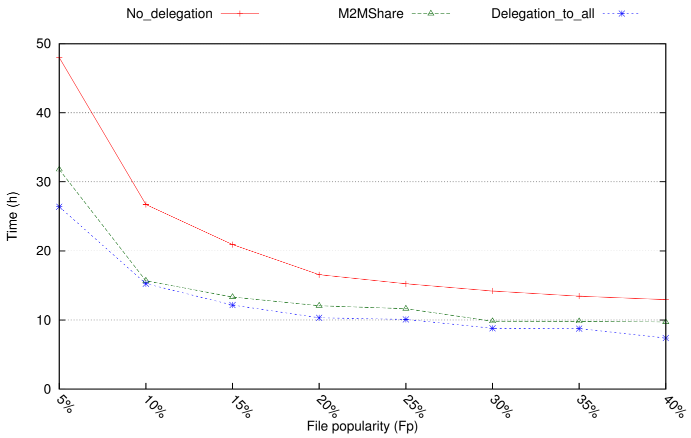


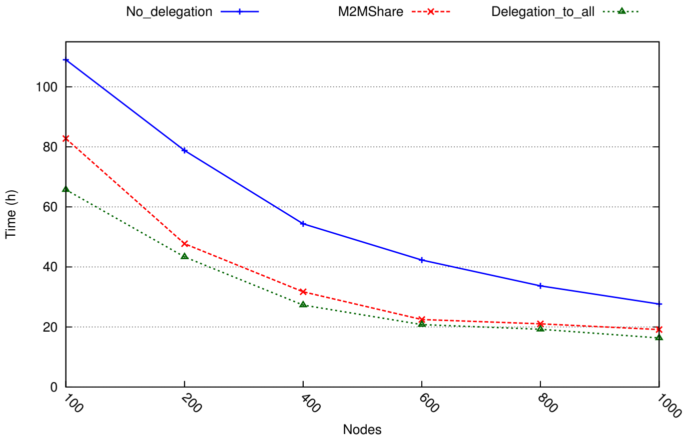
Figure 5.4: Average retrieval time with varying content popularity.

5.1.1.3 Variable Node Population

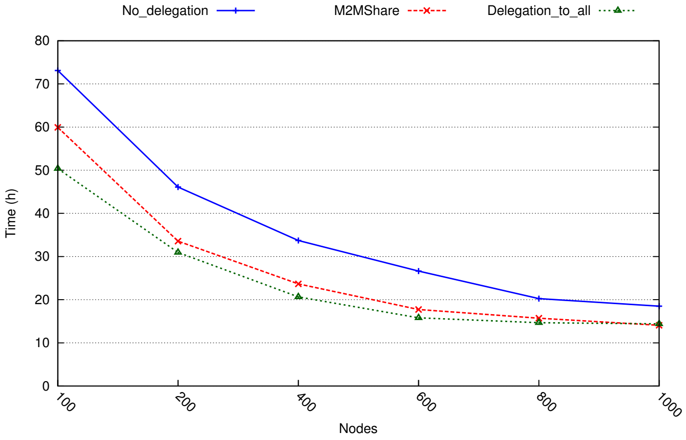
In previous analysis we considered as constant the number of nodes emulating people using M2MShare on their devices. As we can see in Tab. 5.4, in the current analysis we vary the total nodes population of the simulations and the searched content popularity, observing how this affects the performance of compared systems.

In the first scenario (Fig. 5.5a) we consider Fp = 5%. The protocol not employing delegations (black line in the chart) is not able to find any data piece during the simulation time when the considered nodes in the scenario are equal or less than 400. We have indicated this in the chart by assigning to Ft_{avg} a value of 48 h. This is due to the trivial strategy employed by the protocol and to the sparse environment. Even when able to find a node possesing the data content (with N $\geq$ 600), the time needed results bigger than using a strategy employing delegations.

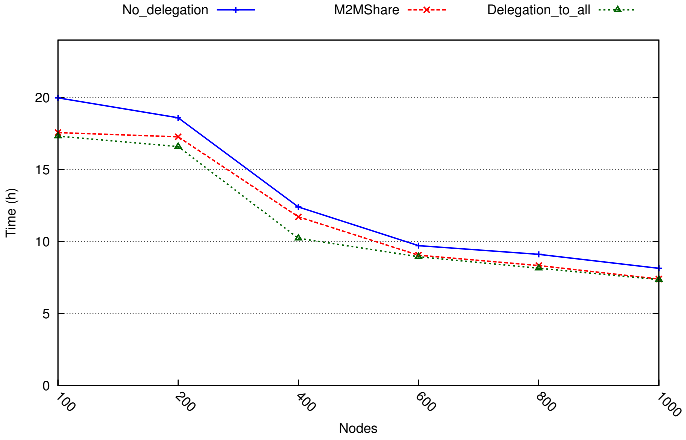
Clearly, when increasing the content popularity (F_p), even the number of nodes in the population that posses the data content increases; as a result, the time to retrieve the data content decreases in all scenarios.



(a) F_p=5%



(b) F_p=10%



(c) F_p=50%

Figure 5.5: Average retrieval time with varying node population and content popularity.

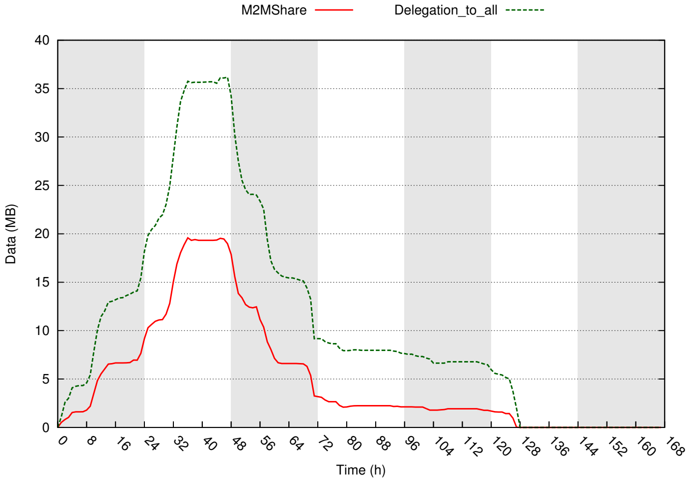
Table 5.4: Simulations settings for evaluation of delegation efficiency with varying node population and content popularity.

Population	100, 200, 400, 600, 800, 1000
Content size	3.0 MB
Content popularity	5%, 10%, 50%
Content distribution	Uniformly distributed
Comparison with	No_delegation, Delegation_to_all
Delegation depth	1
Content Division Strategy	M2MShare
Nr. of simulations	40 x 6 x 3 x 3
Simulated time	7 days

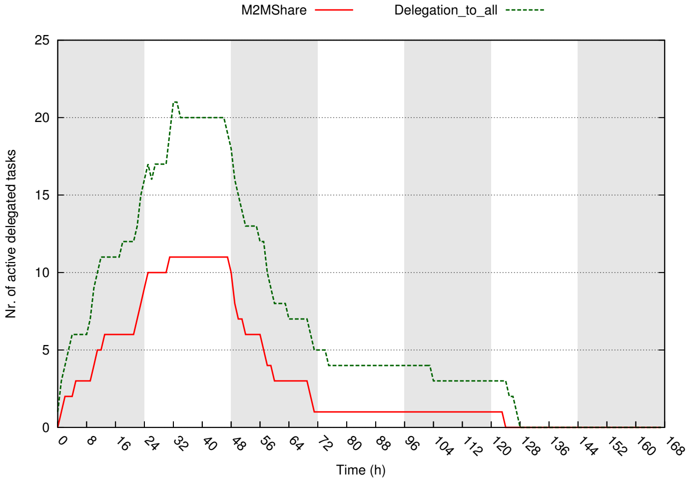
A similar result is achieved also when considering a wider popularity for the required content (Fp = 10%, in Fig. 5.5b). However, in this case, the high popularity of the requested content helps both solutions in finding a producer in a smaller Ft_{avg} than in the previous scenario. Finally, in Fig. 5.5c, the performances of the compared solutions are very similar. This is due to the high content popularity among nodes (Fp = 50%): the chances of eventually encountering a producer in a short time are clearly much higher.

5.1.1.4 Data Redundancy

In the analysis in Sec.5.1.1, considering the results in Fig. 5.2, we show that our system is the most efficient with respect to data transmissions. Although using delegations introduces an overhead in terms of bandwidth usage we control this side effects by delegating only to frequently encountered peers which are expected to be encountered again in the future. Another side effect caused by delegating tasks is the increasing of data redundancy in the whole network. For *redundancy*, in this case, we focus on storage space used in nodes involved in delegation system, not neglecting in any case the effect on transmission costs paid in terms of energy consumption.



(a) Data redundancy in the network



(b) Simultaneously active delegations

Figure 5.6: Redundancy comparison between delegation forwarding schemes.

Table 5.5: Simulations settings for evaluation of data redundancy in the entire network.

Population	1000
Content size	3.0 MB
Content popularity	5%
Content distribution	Uniformly distributed
Delegation type	M2MShare, Delegation_to_all
Delegation depth	1
Content Division Strategy	M2MShare
Nr. of simulations	40 x 2
Simulated time	7 days

Redundancy in the scheme not employing delegations is always zero. Therefore, for this study we compare only the two systems which use task delegations and settings used for these simulations are shown in Tab. 5.5. In Fig. 5.6a we show how the average data redundancy changes during the progress of simulations. It is straightforward to notice the higher value introduced by delegating to all encountered peers whether M2MShare reduces the data redundancy quantity while still achieving the goal of acquiring the requested content. This is due to the number of contemporary active delegated tasks, shown in Figure 5.6b, which is higher in the system which delegates tasks to all encounter nodes.

The trend of this graph is related to the number of simultaneously active delegated tasks, shown in Fig. 5.6b. Whenever a task is delegated, a new node looking for the content is introduced, and if it is found, the node will copy some data interval in its own data storage, and by so doing increasing the total data redundancy. On the other hand, when a delegated task expires, or a servant returns the output back to the consumer node, the cached data retrieved for the task is deleted, freeing space in servant data storage and decreasing the total data redundancy.

5.1.2 Multi-hop Delegation Efficiency

In previous analysis we considered only delegation strategies using single-hop delegation, i.e. once a servant receives a task, delegated from another node, it will not delegate it again to a further-level servant. There are some situations in which single-hop delegations are not enough, due to other factors, like a low popularity of the searched content or its distance from the searching node. It is also possible that all the nodes holding the requested content have different behaviors from those of the consumer and his direct servants. In these cases, we extended M2MShare by giving a servant node the ability to delegate a task on its own. To avoid creating an excessively large number of delegations, we allow it to delegate again only after a trial period, i.e. one day, in which the servant tries to complete the task by itself. At the end of this period, if the task is still incomplete it is delegated again to a new set of upper-level servants.

We simulate the behavior of our protocol in a similar situation by tuning the distribution of the searched content at the beginning of the simulation: we distribute the initial 25 copies only between nodes in a map district on the other side of town from the searching node. As usual we repeated the simulations several times to obtain more accurate results, independently from the initial location of the searching node. In every simulation we then compare then the effectiveness of single- two- and three-hop delegation-forwarding.

A higher number of servant nodes involved in the delegation system results in a higher data redundancy added to the network. To limit the number of delegations used, we implement the multihop delegation system using different Multi-hop Delegation Probability (MhDP). This value indicates the probability that a node would re-delegate an incomplete task in a multi-hop system. As shown in Tab. 5.6, we repeated our simulations with MHDP values of 10%, 25%, 50%, 75% and 100%.

In Fig. 5.7 it is possible to see the average found time employed by

Table 5.6: Simulation settings for evaluation of multi-hop delegation efficiency.

Population	1000
Content size	3.0 MB
Content popularity	2,5%
Content distribution	Distributed in a single district
Delegation type	M2MShare
Delegation depth	1, 2, 3
Delegation probability (DP)	10%, 25%, 50%, 75%, 100%
Content Division Strategy	M2MShare
Nr. of simulations	40 x 3 x 5
Simulated time	7 days

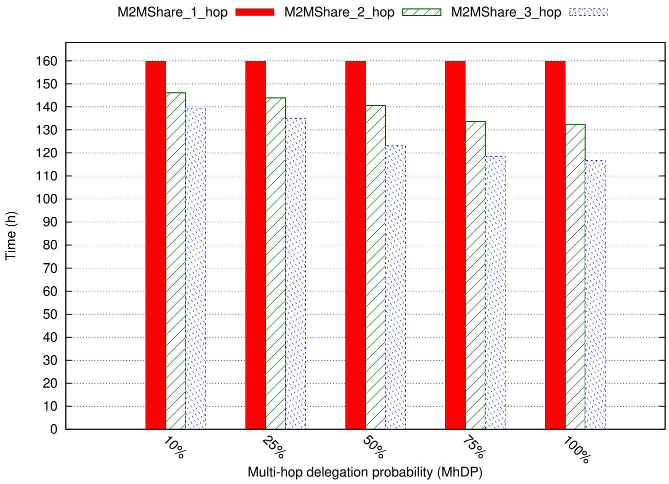


Figure 5.7: Average retrieval time employed by M2MShare with different multi-hop versions in finding the required data content.

the three strategies to return the requested content towards the consumer node. M2MShare with 1-hop delegations can achieve some success, but it is not comparable with results of M2MShare versions employing multi-hop delegations. The two systems using multi-hop delegations find the required content in less time in each simulation run at the expense of higher overhead in terms of number of servant nodes involved, as we will see in Section 5.1.2.2.

5.1.2.1 Map Coverage Statistics

The effect of extending the delegation forwarding mechanism extends the total explored area. A single node can explore only a small area, covered only by its own connectivity range. With M2MShare 1-hop version there is an increase of the explored area, due to delegations to some servants with different movement behavior than the requester, but limited to 1-hop delegation. With multi-hop delegation there is a maximum extension of the coverage area. In this analysis we evaluate the average explored area using different values of delegation depth and MhDP.

First of all we create a control set to evaluate the maximum area that can be explored by nodes during the simulations. To do so we execute 40 simulations with nodes involved in their daily activities, but with no contents distributed. Settings for these simulations are shown in Tab. 5.7. We evaluate the average area explored by all the population nodes during the entire simulation. The related map is shown in Fig. 5.8. We adopt this value to the maximum area that can be explored (100%) and we repeat a set of simulations using the values from Tab. 5.6: we compare M2MShare with 1-hop, 2-hop and 3-hop delegations and using different Multi-hop Delegation Probability (MhDP). The average area explored using different MhDP values is shown in Fig. 5.9.

Table 5.7: Simulations settings for map coverage control set.

Population	1000
Content distribution	No content is available
Delegation type	No delegation employed
Nr. of simulations	40
Simulated time	7 days

As it is possible to see, M2MShare expands the explored area of a single due to delegations to some servants with different movement behavior than the requester, but limited to 1-hop delegation. The maximum area extension is when using up to 3-hop delegations in which almost the entire city is covered by searching nodes.

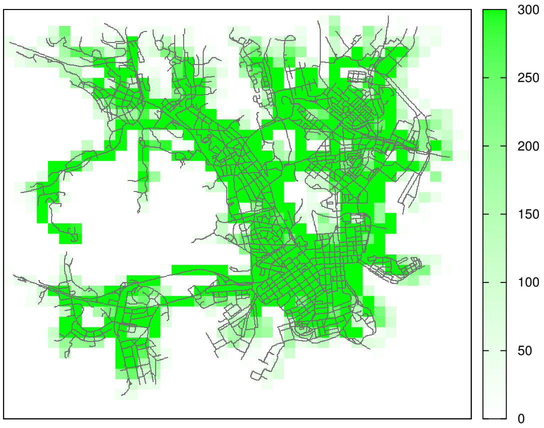


Figure 5.8: Total reachable simulation area.

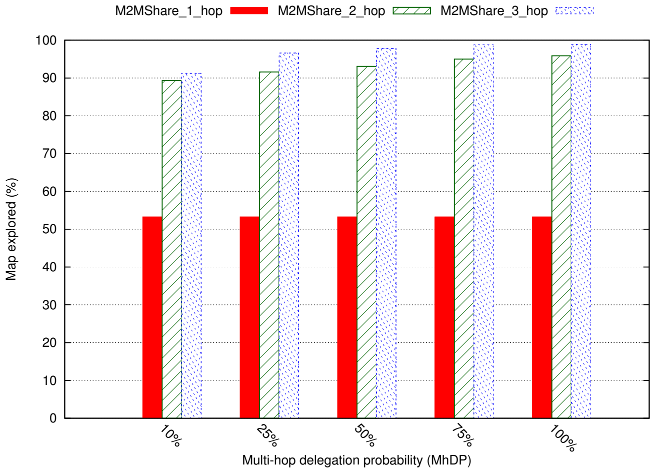


Figure 5.9: Average percentage of explored area employing M2MShare with different multi-hop versions.

For each MhDP value we show the differences in explored area using 1-hop, 2-hop or 3-hop delegations. In Fig. 5.10 we show the tree maps related to the average explored area using the three versions of M2MShare with MhDP = 10%. Raising the MhDP value to 25% (Fig. 5.11) it is possible to see an increment of the explored area using multi-hop delegations. This increment is less visible increasing the MhDP again to 50% (Fig. 5.12), 75% (Fig. 5.13) or 100% (Fig. 5.14).

5.1.2.2 Data Redundancy

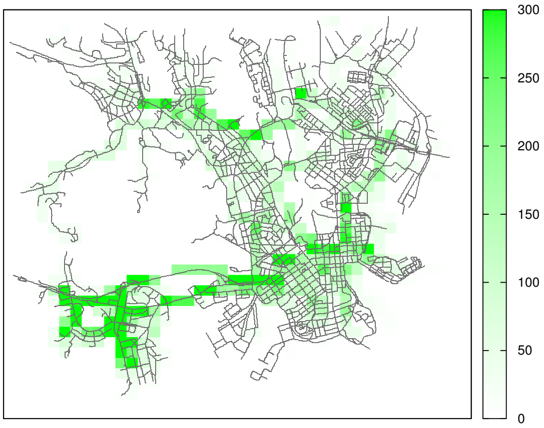
Through the analysis in Section 5.1.1.4 we evidenced the redundancy added in the network by using delegation and showed how our system outperforms the other trivial system which uses delegation toward all encountered nodes. In current section we show the impact of using multi-hop delegation respect to added redundancy in the network. For these simulations we use the settings shown in Tab. 5.6. We compared three versions of M2MShare with different maximum number of delegation hops. We also changed the probability that a servant node would delegate again a received pending task.

Figure 5.15 to 5.19 show the difference in number of delegation used and data redundancy added by the three versions of M2MShare. Both of them use the same number of delegations for the first day, then the 2-hop and the 3-hop versions start to delegate again the incomplete task. The next difference can be seen after another day, when nodes adopting 3-hop delegations again delegate the unaccomplished task. It is straightforward to notice the increment of servant nodes using 3-hop delegations versus 2-hop or 1-hop systems.

A higher number of servant nodes involved in the delegation system results in a higher data redundancy added to the network. To limit the number of delegations used and related data redundancy added, we implement the multi-hop delegation system using different Multi-hop Delegation Probability (MhDP). This value indicates the probability that a node would delegate again an incomplete task in a multi-hop system.

(a) Explored area with 1-hop delegations

(b) Explored area with 2-hop delegations

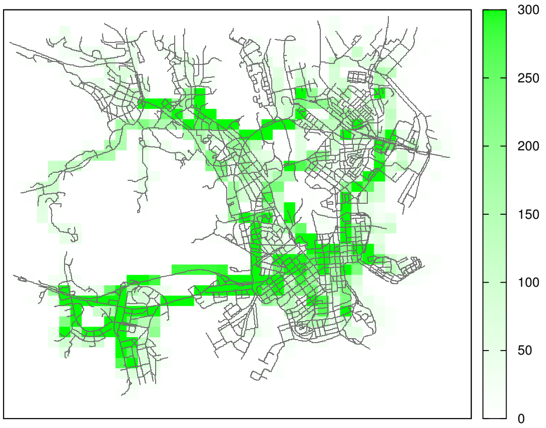


(c) Explored area with 3-hop delegations

Figure 5.10: Average explored area with MhDP = 10%

(a) Explored area with 1-hop delegations

(b) Explored area with 2-hop delegations

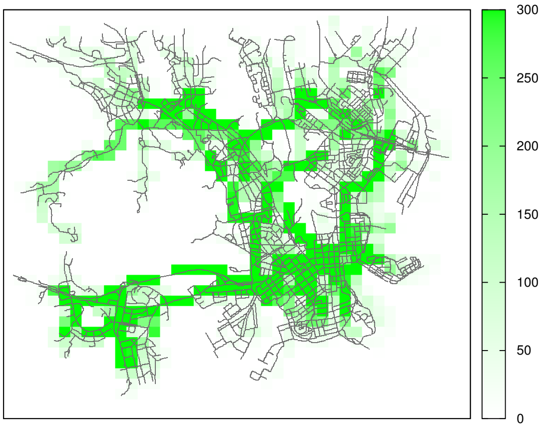


(c) Explored area with 3-hop delegations

Figure 5.11: Average explored area with MhDP = 25%

(a) Explored area with 1-hop delegations

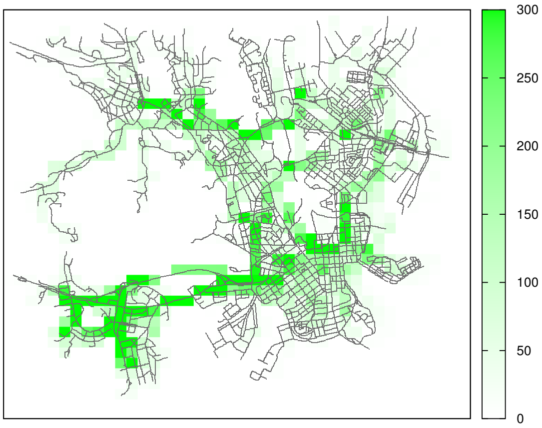
(b) Explored area with 2-hop delegations



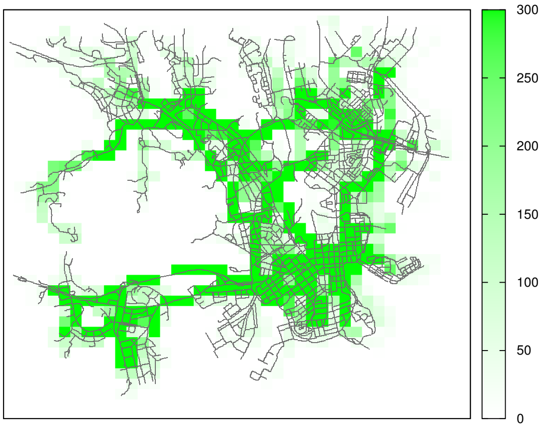
(c) Explored area with 3-hop delegations

Figure 5.12: Average explored area with MhDP = 50%

(a) Explored area with 1-hop delegations



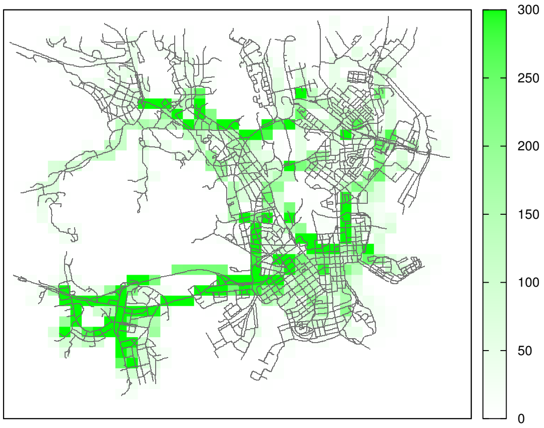
(b) Explored area with 2-hop delegations



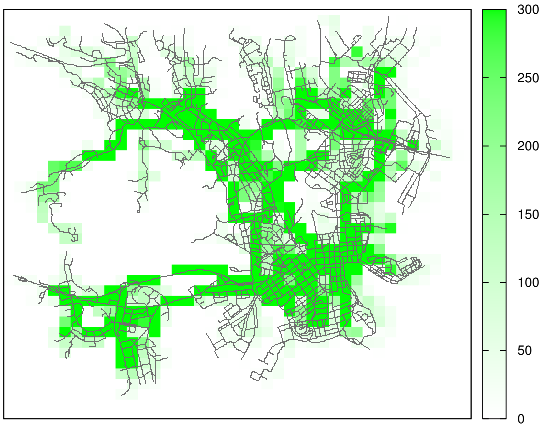
(c) Explored area with 3-hop delegations

Figure 5.13: Average explored area with MhDP = 75%

(a) Explored area with 1-hop delegations

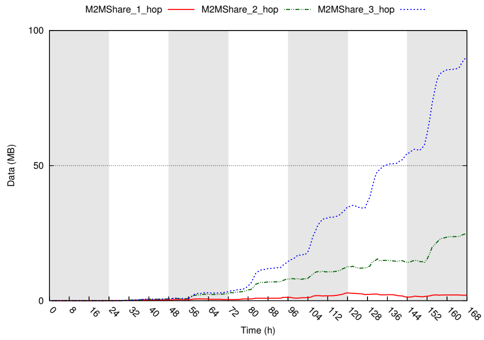


(b) Explored area with 2-hop delegations



(c) Explored area with 3-hop delegations

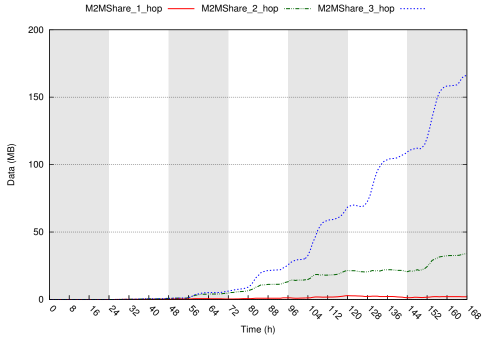
Figure 5.14: Average explored area with MhDP = 100%



(a) Data redundancy in the network

(b) Number of simultaneously active delegated tasks

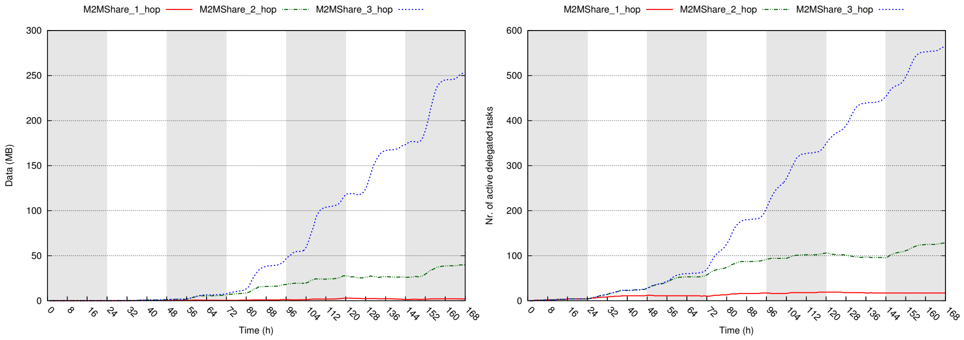
Figure 5.15: Average data redundancy and number of active delegated tasks with MhDP = 10%



(a) Data redundancy in the network

(b) Number of simultaneously active delegated tasks

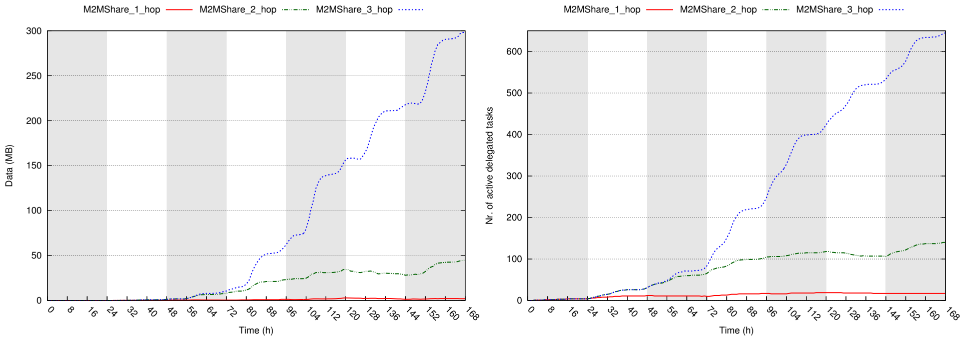
Figure 5.16: Average data redundancy and number of active delegated tasks with MhDP = 25%



(a) Data redundancy in the network

(b) Number of simultaneously active delegated tasks

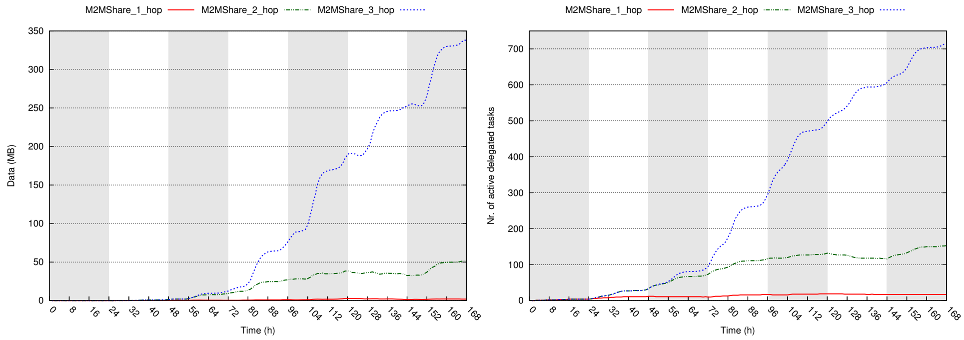
Figure 5.17: Average data redundancy and number of active delegated tasks with MhDP = 50%



(a) Data redundancy in the network

(b) Number of simultaneously active delegated tasks

Figure 5.18: Average data redundancy and number of active delegated tasks with MhDP = 75%



(a) Data redundancy in the network

(b) Number of simultaneously active delegated tasks

Figure 5.19: Average data redundancy and number of active delegated tasks with MhDP = 100%

We repeated our simulations with MhDP values of 10% (Fig. 5.15), 25% (Fig. 5.16), 50% (Fig. 5.17), 75% (Fig. 5.18) and 100% (Fig. 5.19). With a small MhDP value a small number of delegations are used, reducing the overall data redundancy. Raising MhDP value, more delegations are used, increasing the data redundancy introduced into the network. With MhDP 100% a servant peer delegates a unaccomplished task to every encountered node which exceeds Frequency Threshold value, after the one-day trial period. With a such high MhDP value we can see that over 70% of nodes in the simulation are involved in delegations system.

5.1.3 A Named-data Approach

We now consider a named-data approach to M2MShare discussed through Sec. 4.1.4. In our simulations each node is equipped with a Bluetooth network interface as in the original experiment from which the dataset originates. The data content to be retrieved has a fixed size of *100KB* and delegations on the servant side have a predefined *Time-To-Live* (TTL) specified through a configuration variable whose default value is set to *24* hours; the value is based on the rationale that human behavior exhibits periodic behavior on a daily basis. Further, each device continually probes its surroundings taking note of neighboring nodes and the beconing frequency is as in the original experiment and entirely dependent on the trace. On the other side query requests are periodically issued every *5* minutes and are subject to the mechanism explained in Sec. 4.1.3.4.

We compare the delegation forwarding strategy against the following approaches: *(i) No_Delegation* scheme which does not employ delegation forwarding and retrieval is done only when a node holding the requested data content is found in the local connected networks visited by the consumer node; *(ii) Delegate_to_all* scheme employing employing a trivial strategy where unaccomplished retrieval tasks are assigned to all encountered nodes.

For the evaluation, we have partitioned the trace in *seven* distinct classes of consumer/producer pairs where each class denotes the day when the first encounter between the nodes took place. Formally each class is defined as shown in Def. 5.1. The upper axis of Fig. 5.20 shows each class configuration, the number of consumer/producer pairs that reside in it. The reason for partitioning the trace is done so as to better evidence the differences between the forwarding strategies and the tradeoffs that arise in space/time due to employing one approach or the other.

$$class_i = \{pair(x, y) \mid x \neq y \ \wedge \ encounter\_day(x, y) = i\} \tag{5.1}$$

We report that the majority of encounters between the nodes takes place within the first hour of the contact trace and all the strategies find the data by employing the same amount of time, while the flooding scheme incurs unnecessary overhead to be attributed to the epidemic approach it employs. We took a next step at filtering the data residing in *class 1* and the reported number of pairs correspond to pairs whose encounter is past the first hour of the contact trace. The pairs of con-

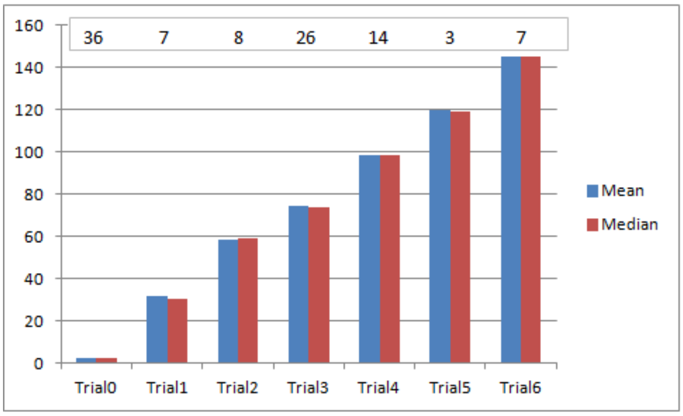


Figure 5.20: Retrieval time in hours employed by the *No_ delegation* scheme along with class membership distribution shown above.

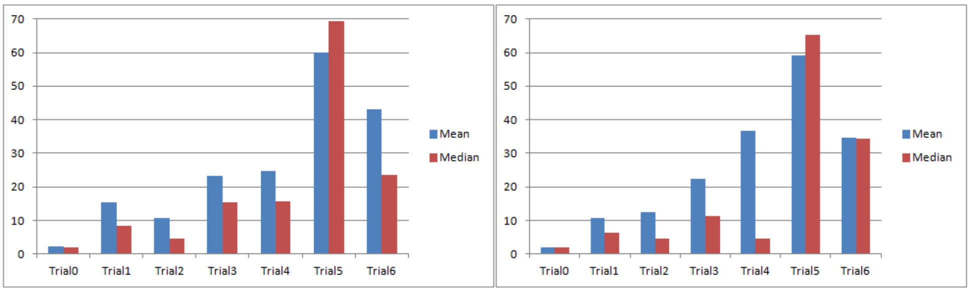
sumer/producers residing in *class 7* correspond to nodes which never come in direct contact with each other during the entire trace duration. We argue that in a real scenario this class might indeed be the rule rather than the exception.

The metrics we study throughout the next section is the *(i)* content retrieval time, *(ii)* amount of data transferred and *(iii)* the delegation return index, which is the fraction of returned delegations against the number of overall employed delegations (if any).

5.1.3.1 Results

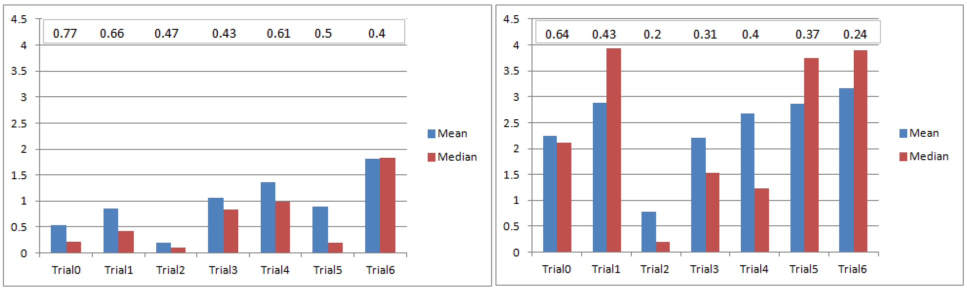
In Fig. 5.20 are shown the aggregate retrieval times employed by the *No_ delega-tion* scheme for each of the classes of consumer/producer nodes. Comparing these results from those exhibited in Fig. 5.21, representing *M2MShare* and *Delegate_ to_ all* scheme respectively, the benefit of employing delegations is evident. The retrieval process in each class is noticeably reduced as compared to the benchmark scenario. Hence, the delegation forwarding strategy helps nodes expand their reach area in other disconnected local networks speeding retrieval times. It is noteworthy pointing out, that the monotonic trend exhibited by the *No_ delegation* scheme is due to the chosen class partitioning rather then some underlying property of the contact trace. Indeed, in the latter scenarios this trend is broken, showing that the result outcome depends upon node encounter pattern rather than time.

Confining the analysis on the delegation schemes, the epidemic approach to delegations drastically reduces retrieval times, outperforming the threshold election scheme in all the scenarios expect those residing in *class 5*. This is explained by a lower, aggregate contact encounter rate of consumer nodes residing in this class when compared to the other classes.



(a) M2MShare (b) Delegate_to_all

Figure 5.21: Retrieval times in hours employed by each delegation scheme.



(a) M2MShare (b) Delegate_to_all

Figure 5.22: Incurred redundancy in MB for each delegation scheme.

Now let us take a look at the complete picture by discussing the other metrics of relevance. Both the delegation forwarding schemes have another common feature, that of exploiting redundancy to increase the chances of content delivery. Indeed a node during his journey might employ multiple delegations for the same content retrieval task, which if accomplished more than once, translates into redundancy introduced into the network paid in terms of storage and transmission costs. Comparing the redundancy data shown in Fig. 5.22, it is clear that the threshold election (M2MShare) scheme incurs lower redundancy into the network, several orders of magnitude comparing to the epidemic approach. The delegation forwarding technique is content-agnostic, in the sense that no information about data content is used to compute the forwarding utility while it provides the means to expand the node reach area by properly selected encountered nodes.

The upper axis in Fig. 5.22 shows the average percentage of returned delegation tasks, computed as the number of overall delegated tasks against the number of those which are returned before the delegation expires. In this metric, a returned task does not necessarily mean an accomplished task as this process is accidental and orthogonal to the forwarding strategy. As it is evidenced, the threshold scheme achieves an higher return index in all scenarios when compared to the epidemic approach. To be noted, that the delegation expiration time is a static system parameter based on the rationale of daily human patterns.

So far, we have exposed the results concerning pairs of nodes whom have an encounter during the contact trace duration. However, as previously anticipated there is another class of consumer/producer denoted by *class 7* comprised by pair of nodes which do not come in contact with each other in any level-2 connected network for the whole trace duration. Below, there is a summary of the above metrics for this class.

In this class of encounters a similar trend and tradeoff in time and space arises whereby the Delegate_to_all scheme prevails in terms of

Table 5.8: Aggregate retrieval times (hours) for Class 7.

	M2MShare	Delegate_to_all
Average	33.6630	12.0192
Median	3.1198	1.9277

Table 5.9: Aggregate redundancy (MB) for Class 7.

	M2MShare	Delegate_to_all
Average	1.269	2.3034
Median	0.254	1.9

employed retrieval times while the threshold election scheme in terms of incurred redundancy. We now propose a way to quantify this tradeoff, providing an evaluation index computed according to equation 5.2 and shown in Fig. 5.23.

$$comparison_i = \alpha \times avg\_ret_i + (1-\alpha) \times avg\_red_i \tag{5.2}$$

where avg_ret_i and avg_red_i are the average retrieval time and average incurred redundancy for the particular class i and α is $\frac{1}{2}$. An α of $\frac{1}{2}$ gives the same importance to both studied quantities. We recall that the redundancy metric does not simply account for introduced storage redundancy but most importantly for transmission costs translated into battery consumption. As shown from Fig. 5.23 the threshold scheme convincely outperforms the flooding scheme in all scenarios.

5.1.4 Discussion

Through this section we studied the performance of Mobile-to-Mobile (M2M-Share): a content sharing solution tailored for the characteristics of mobile disconnected networks. Addressing communication in sparse networks we proposed *delegation-forwarding*, a mechanism whereby nodes delegate unsatisfied/unaccomplished retrieval tasks to other nodes in the system. Our protocol employs node mobility to retrieve data content

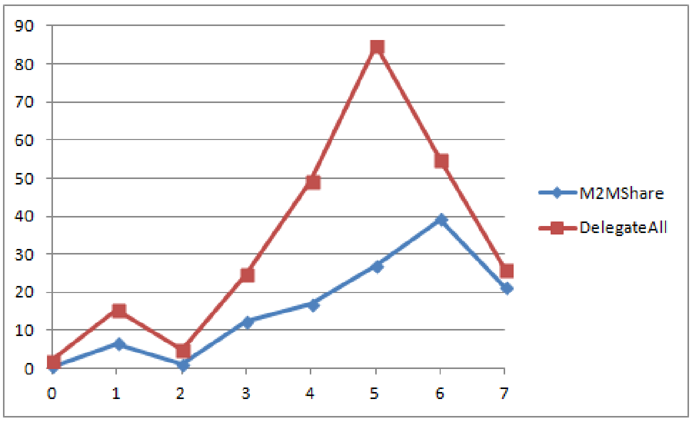


Figure 5.23: Comparison index for the delegation forwarding strategies.

available elsewhere, in other connected portions of the network. Delegating tasks to all encountered nodes in the system proves resource-consuming, and at times with no added benefits. The delegation mechanism exploits node contact history to control system redundancy, delegating tasks to frequently encountered nodes whom are expected to be met again in the future.

Along with the node-based, connection-oriented system design we proposed a name-based design tailored to the characteristics of the NDN architecture. Both designs were evaluated employing a synthetic, map-based mobility model and a field gathered contact trace. A due observation regards the similar trends that the different designs exhibited. Intuitively, the NDN paradigm given its *modus operandi* should have out-performed the other. This similar behavior is to be attributed to the nature of contact patterns exhibited by the traces, in particular while referring to the synthetic data trace. The sparse environments we considered do not give much space to the caching strategy to show its benefits in time.

Our focus was at devising the networking techniques aimed at sustaining a content sharing solution without infrastructure reliance. We showed the effectiveness of our delegation forwarding strategy when compared to other approaches and the tradeoffs that arise. The feasibility of an urban-wide, real deployment for this applicative scenario as is, is undermined by its retrieval times, directly influenced by the number of participants. Addressing this issue, in the following section we set on a trial to study the delivery profiles of MDTN, a service delivery platform built on top of the PTS which could be used in synergy with M2MShare to help boost retrieval times.

5.2 Public Transportation System as a Service Delivery Platform

We now consider MDTN, a delay tolerant platform deployed on top of the PTS aimed at supporting elastic applications. The metrics we used to evaluate system performance are delivery time and rate, and the total resources used by the system. From these metrics we are able to understand which kinds of services are meaningful to be provided from the MDTN platform. At the end of the evaluation part, we provide some discussion about the achieved results.

5.2.1 Delivery Delay

A prospect of average delivery delays under the various combinations of routing policies and distributions schemes is reported in Tab. 5.10 and Tab. 5.11.

As we can see from Tab. 5.10, in Milan with the ILD scheme, responses will be delivered with an average delay varying between less than 2 hours to almost 4 hours, being the Min hop policy the worst case. Under these lenses, the system could be actually used for news retrieval, delay tolerant web browsing, and distribution of information regarding local events. In Chicago (Tab. 5.11), the situation is a little worst: in the ILD scheme

Table 5.10: Summary of delivery delay in Milan (values in hours).

	mean	**median**	**std. dev.**
pure muling	1.56	1.41	0.86
infrastructure-less delivery			
Min hop	3.89	3.10	2.63
Op-HOP	2.70	2.49	1.25
MaxProp	1.59	1.51	0.61
infrastructure aided delivery			
Min hop	1.59	1.41	0.65
Op-HOP	1.82	1.59	1.13
MaxProp	0.71	0.65	0.36

Table 5.11: Summary of delivery delay in Chicago (values in hours).

	mean	**median**	**std. dev.**
pure muling	1.72	1.27	1.62
infrastructure-less delivery			
Min hop	5.25	4.58	2.85
Op-HOP	5.14	4.69	2.44
MaxProp	3.11	2.69	1.77
infrastructure aided delivery			
Min hop	2.24	1.63	1.92
Op-HOP	3.16	2.37	2.57
MaxProp	1.28	0.85	1.43

delay ranges between about 3 and 5 hours. News retrieval is still feasible but other services might become unrealistic.

A different scenario comes forward when the distribution scheme changes and a request is routed only up to the first IG, while its response is transmitted to the IG of the destination line via wireline. In addition to the fact that the average delay time is reduced, Min hop starts performing better than Op-HOP. The explanation of this sits in the optimization of Op-HOP against Min hop: the latter is more likely to miss contacts and carry the request toward the IG of the first line, thus reducing the average delay more than Op-HOP. In all cases, delay is adeguate for urban-wide, non real-time services.

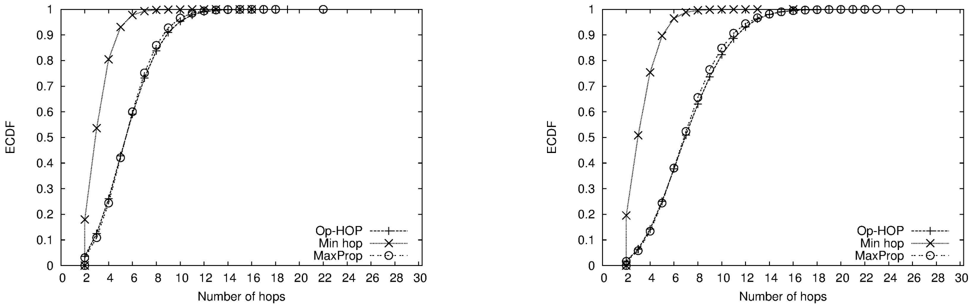
If we combine Tab. 5.10 with Tab. 5.11, it is possible to draw some more general conclusions. First of all, the Min hop routing policy albeit designed for wired networks performs in a way comparable with the other routing policies. Second, MaxProp seems to be constantly outperforming the other two routing policies. In particular, regarding Op-HOP, in [73] we demonstrated that it scales better than MaxProp in terms of network load, but it seems to follow the same trend when scalability comes in terms of PTS extension. This can be explained thanks to the multi-copy forwarding approach of MaxProp.

5.2.1.1 Number of Hops

A performance metric closely connected to the delivery delay is the number of traversed hops. This is because every hop is the result of a contact opportunity or a (sometimes long) travel toward an IG.

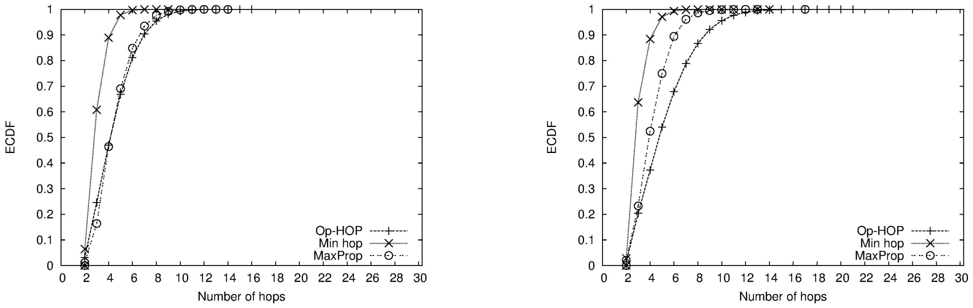
Through Fig. 5.24 to Fig. 5.25 is reported the ECDF of the number of traversed hops for each of the studied topologies under the different distribution schemes.

In the charts, the *pure muling* policy is not reported since it would not bring any meaningful information (the number of hops is fixed) and all



(a) Milan (b) Chicago

Figure 5.24: ECDF of number of hops using the ILD distribution scheme.



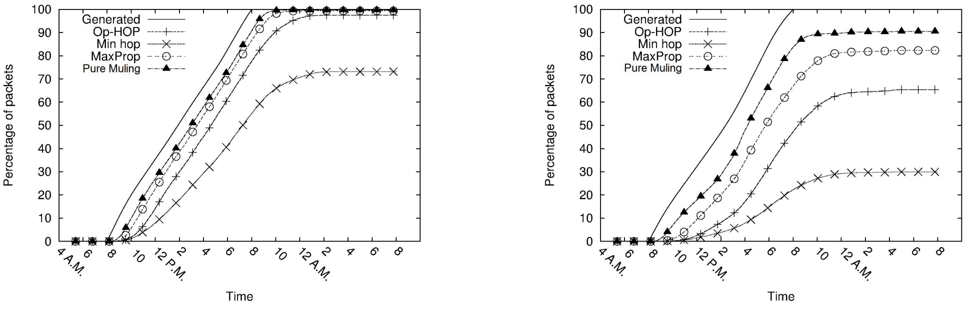
(a) Milan (b) Chicago

Figure 5.25: ECDF of number of hops using the IAD distribution scheme.

profiles have a minimum of two hops because the data have to traverse at least an IG before reaching the destination. Analyzing the charts, we can observe that, despite the lower delivery delay, MaxProp does not outperform Op-HOP in a sensible way. The only exception is Chicago with the IAD scheme (Fig. 5.25b), but this is more likely due to a scalability problem of Op-HOP.

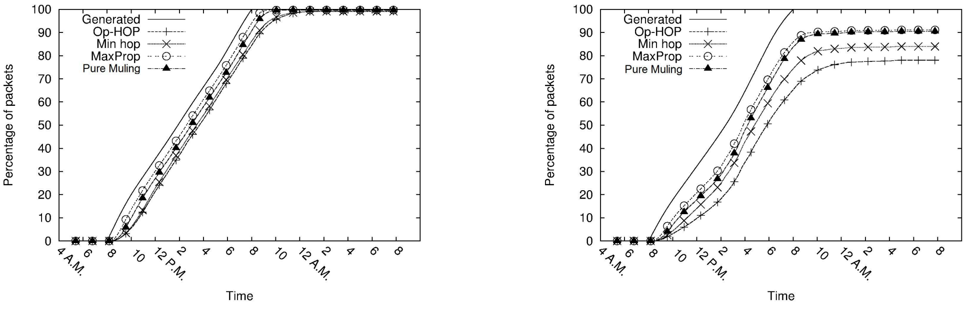
5.2.2 Delivery Rates

Figures 5.26 and 5.27 show the traffic delivery profiles in the various cases.



(a) Milan (b) Chicago

Figure 5.26: Request delivery rates with the ILD distribution scheme.



(a) Milan (b) Chicago

Figure 5.27: Request delivery rates with the ILD distribution scheme.

First of all, Min hop always generates the worst performance (sometimes to an unacceptable level such as in Fig. 5.26b) unless the system is extremely connected (Fig. 5.27a). This can be explained considering the extremely long delivery delay: packets will accumulate during the evening while buses going out of service will make the network less connected, increasing the number of packet drops. Op-HOP is not suffering to the same degree because paths are built based on encounter probability: a smaller fraction of the total traffic will stay in the system for a much longer time and will be dropped at the end of the day, but a more considerable number of packets will be able to find its way to destination.

Second, we have to observe that not even the PM distribution scheme is able to guarantee a 100% delivery of packets. This is particularly true for Chicago (Fig. 5.26b and Fig. 5.27b) and depends on the PTS being scarcely connected: buses are less likely to be able to spool packets from the internal buffer when going out of service. Nevertheless, MaxProp is able to slightly outperform the PM scheme but it needs assistance from the IGs.

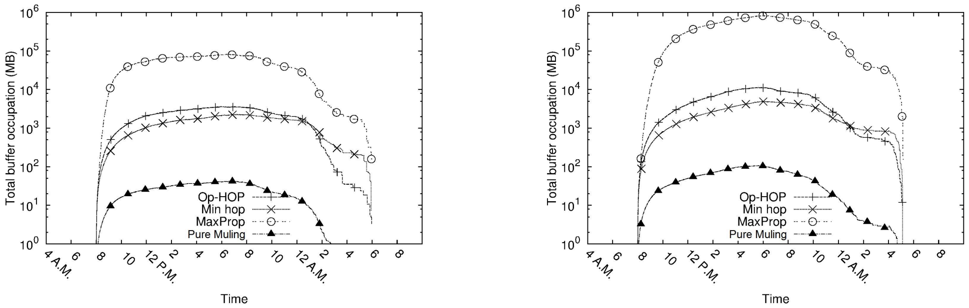
5.2.3 Resource Usage

So far, MaxProp seems to be the best option for MDTN: lower delivery delay and acceptable delivery rate. Nevertheless, we have to remember, many of this advantages comes due to its multi-copy approach. More likely, it is going to use more resources than the other routing policies.

In this section we will analyze the usage of system resources under the various routing conditions.

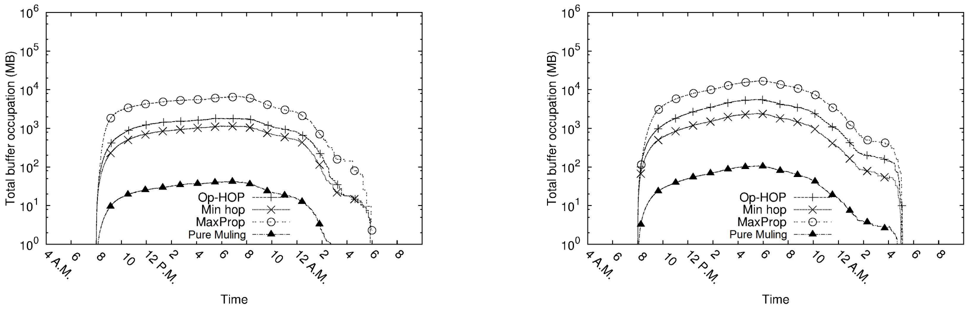
5.2.3.1 Buffers Usage

The first metric we are going to analyze is the global buffer usage. Figures 5.28 and 5.29 depict the evolution of the total amount of traffic in transit from source to destination.



(a) Milan (b) Chicago

Figure 5.28: Buffers usage in the ILD distribution scenario.



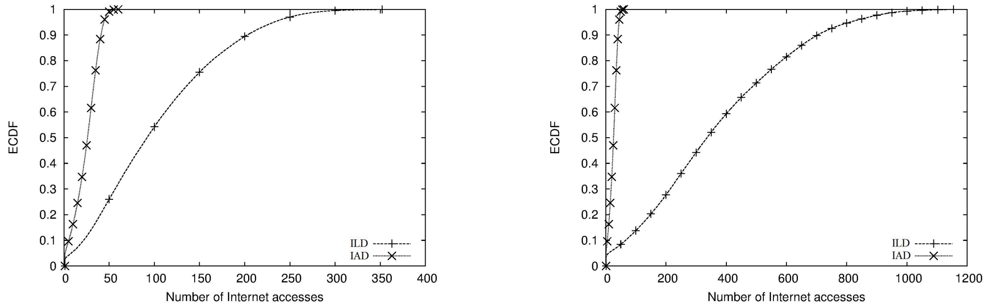
(a) Milan (b) Chicago

Figure 5.29: Buffers usage in the IAD distribution scenario.

As we can see from all figures, MaxProp is always one order of magnitude more demanding than the minimum hop routing policy. Despite the fact, for this specific simulation, the absolute value is still reasonable for modern technology it will be amplified many times in a real deployment, where hundred of thousands of users will be streaming load from and to the infrastructure. The effect is sensibly reduced (to one order of magnitude) only when transmission between IGs is used. Interestingly enough, these are the cases where performances are less prominent when compared to the other policies. With regard to the difference between the profiles of minimum hop and Op-HOP, it is due to different delivery delay: the added request from Op-HOP is the space required by packets which are taking a longer to reach destination.

5.2.3.2 Wasted Internet Access

Another resource for MDTN is the Internet access: an IG, upon receiving a packet must use its access network to fulfill the request an produce a response. In the case of multi-copy routing, multiple copies of the same request will require multiple accesses to the Internet with a consequent waste of resources. Figure 5.30 shows the ECDF of the probability to have a given number of Internet accesses for a given packet.



(a) Milan (b) Chicago

Figure 5.30: ECDF of the number of unrequired Internet access for MaxProp.

As we can see from the curves, with ad hoc routing, only 50% of packets will generate less than 100 internet accesses in Milan and less than 350 in Chicago. These values drop considerably when transmission between IGs in in place, lowering down to 25 in both cases.

While memory is cheap and we can sustain a certain degree of overallocation, Internet access may be expensive and increasing usage by 20 times may have a severe impact on the system. This situation calls for a carefully planned content delivery network tailored for urban services accessed via a PTS.

5.2.4 Discussion

Through this section we studied the performance of Mobile Delay/Disruption Tolerant Network (MDTN): a delay tolerant application platform built on top of a Public Transportation System (PTS) and able to provide service opportunistic connectivity. Differently from previous work, we applied our solution to two real city environment: Milan (Italy) and Chicago (Illinois), extending the protocol *modus operandi* to a more complex and realistic usage scenario. The system have been tested using different routing strategies: one single copy typical of wired network (Min hop), one single copy for DTN based on encounter probability (Op-HOP), and one multi-copy for DTN (MaxProp). Moreover, data transfer between Internet Gateways located at line ends has also been taken into consideration. By means of simulation we demonstrated that MDTN could be a viable solution for pull/push-based advertisements dissemination and non-critical data retrieval.

Nevertheless, performance indexes of the considered routing policies have shown that there is no golden rule. Forwarding between IGs will actually make a difference in the service provisioning. Single-copy routing approaches such as Op-HOP are more favorable in highly connected environments due to their reduced resource usage, good delivery rate, and reasonable delay. On the other hand, MaxProp and multi-copy routing approaches have to be preferred in loosely connected environment despite the fact we still need to understand if the added resources usage is worth the performance improvement.

5.3 AirCache: A Floating Data Network

We now consider AirCache, a distributed protocol aimed at sustaining a floating data network. The metrics we used to evaluate the protocol performance is the time interval the data survives in the AoI after it has been injected. Another, not of less importance feature of our protocol, is that of controlling the spatial distribution of the data. We quantify its relevance by using the distance metric as formulated below:

$$Distance_{data} = \frac{1}{N} \times \sum_{n \in AoI} dist(n, hops - min(n, datum)) \tag{5.3}$$

That is the distance is the minimum number of hops required to reach the data averaged over the population of nodes sustaining the AC. The denotation n∈AoI serves to indicate that we consider inside this metric also nodes that are in listening mode, ready to join the AC, during the sampling process. Nodes caching a replica of the data contribute to zero. As anticipated, the mobility model used to evaluate our proposal is the RWP mobility model generated with the parameters shown in Tab. 5.12 and a cut off time of 3600 s.

Before simulation starts each node in the population is attributed an initial energy source taken uniformly at random from the range [500, 1.750] mAh. The application buffer size is not taken into consideration given the nature of the experimentation. That is sender/receiver of the data are chosen by exploiting the battery level only. For each simulation scenario we perform several runs per configuration using different

Table 5.12: Parameters for the RWP mobility model used for evaluation.

Simulation Area	400 × 400
Mobility	[0.5, 1.5] m/s
Pause Time	30 s
Trace Duration	60 min
Attraction Point	200 × 200
AoI radius	150 m

mobility seeds. As explained in Sec. 4.3.4, we avoid AC creation and interference by controlling the simulation scenario. This issue is avoided by delegating the responsibility of AC creation to a random node within the AoI and restrict others only to join it. Nodes might still move away from the original AC, giving rise to different ACs which are synchronized with each other. The merger in this scenario is handled by the protocol without additional intervention; conflicting nodes (if any) enter a re-contention period for new available slots.

5.3.1 Data Survivability in the AoI

Concerning the study of data survivability in the AoI we generate different contact traces with the parameters as shown in Tab. 5.12. The AoI is a concentric circle within the simulation world. Nodes having a replica of the data once outside the AoI discard the data along with the AC related information and are marked as inactive. We study the system behavior by controlling the in/out flow of nodes inside the AoI.

To achieve this, the nodes falling outside the AoI before simulation starts are marked as inactive. Inactive nodes become active when inside the AoI and their number depends upon the flow of nodes that have left the AC. To enforce this in/out flow inside the AoI we take a snapshot of the simulation world each 5 s and control the in-flow depending on the out-flow when compared to the last taken snapshot. Active nodes according to the state transition diagram shown in Sec. 4.3.2.1 with a parking time of 5 s.

We study the data survivability with varying population size and different in/out policies. For each configuration we perform 10 runs with different mobility seeds so as to increase the confidence of the obtained results. The considered data size is that it fits inside a single BCH without triggering the need for extra slot reservation. The results are shown in Tab. 5.13.

The *Trace* scenario represents the benchmark solution where no in/out flow policy is applied and this flow depends entirely on the mobility behavior exhibited by the original trace. The last column (*Departure*) denotes the average time interval the producer nodes left the AoI for each configuration run.

Table 5.13: Average AC survival times in minutes with varying population size and different in/out flow policy.

	Trace	**1:1**	**1:2**	**1:3**	**1:4**	**1:5**	**Departure**
Population Size 10							
Average	48.53	39.05	22.11	20.11	10.1	7.24	6.2
Median	42.12	30.55	18.42	17.23	6.56	4.54	4.3
Population Size 15							
Average	52.12	49.11	40.2	30.23	25.23	20.53	6.3
Median	43.5	40.57	33.47	23.47	18.34	15.45	4.5
Population Size 20							
Average	55.21	51.56	48.45	33.45	27.57	25.16	6.55
Median	53.32	49.06	46.45	32.29	25.54	23.46	5.5
Population Size 25							
Average	60	59.51	54.11	50.01	45.55	38.51	7.02
Median	60	56.53	50.37	44.4	39.5	30.36	4.5
Population Size 30							
Average	60	60	58.11	55.12	51.34	48.54	6.76
Median	60	60	55.45	50.7	49.23	40.32	7.3
Population Size 35							
Average	60	60	60	60	58.12	49.11	7.54
Median	60	60	60	60	52.18	44.5	8.12

In all scenarios the system is able to provide a margin of profit which decreases when the control flow policy becomes more aggressive. In overall the obtained results show that node availability inside the AoI is of crucial importance. A higher node population in the AoI translates into an AC being alimented and data being replicated. In the lower end, when a *1:6* policy is enforced is where the protocol is not as resilient to counteract the effects of nodes departing the AoI. This can be observed by the similar trend exhibited for this policy in all the contact traces. However even in this scenario the system is able to provide a margin of profit when compared to the scenario where no cooperation is involved.

5.3.2 Spatial Data Distribution

As argued through Sec. 4.3, a homogeneous data distribution could facilitate data accessibility in the AoI. We delegate this control to cluster intersection nodes, enforcing this feature whenever circumstances require it. We have performed a set of simulations, 10 for each configuration run, with the settings as shown in Tab. 5.14 to confront both the scenarios where no control and when control is enforced.

Table 5.14: Settings for the spatial control distribution scenario.

Simulation Area	500 × 500
Mobility	[0.5, 1.5] m/s
Pause Time	30 s
Trace Duration	10 min
Population Size	35
Attraction Point	250 × 250
AoI radius	200 m

In the scenario where no control is enforced, the data does not reach the end of the trace in 3 different runs, with a data survivability of 2.45, 5.45 and 6.34 min. These configurations are not considered when

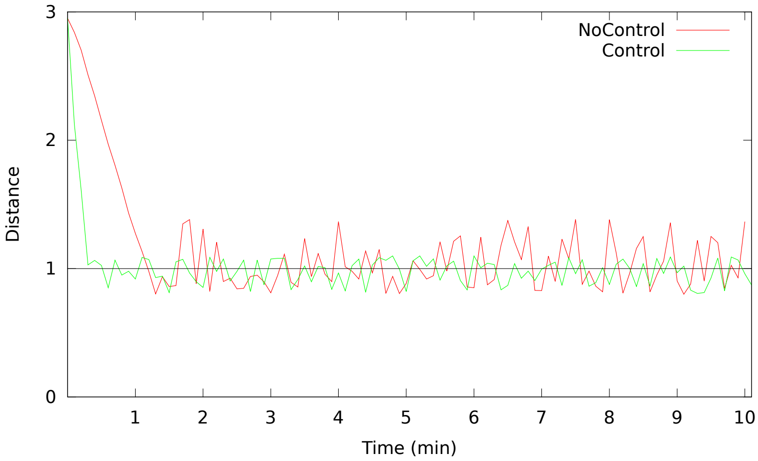


Figure 5.31: Distance metric comparison of the strategy employing control distribution versus the one not employing it with a replication policy of Min(1)/Max(2).

computing the distance metric reported in Fig. 5.31. An important fact to evidence is that the scenario enforcing control on the data dissemination is more resilient to mobility when compared to other one. This is to be attributed to the data being disseminated into the network more rapidly, creating a number of replicas distributed over the AoI rendering the AC data less subject to mobility and nodes departing the AC in a certain area. This trend is shown in the upper part of graph where in the control scenario the distance metric converges to 1 more rapidly whether the no control scenario exhibits a higher convergence time. The process of data dissemination in the no control scenario is to some extent left to chance, data being advertised by cluster heads in other connected parts of the AoI, or disseminated to other parts exploiting node mobility. When the data has reached the target replication level in the AC, the no control scenario is subject to more fluctuations while the other exhibits a more stable trend. This difference in this fluctuating behavior stands in their *modus operandi* where in the control scenario data the replication policy is enforced in a timely fashion while in the no control policy it depends upon node mobility physically carrying the data to newly arrived nodes.

5.3.3 Discussion

Through this section we presented our preliminary analysis of AirCache: a distributed, collision-free protocol aimed at sustaining a floating data network. In this envisioned scenario the data are supplied and maintained by the users themselves. Data survivability and accessibility are two important features we identified. To this end, we performed a set of simulations studying protocol behavior in presence of mobility and variable node density. The mobility characteristics considered, represent a scenario which is arguably realistic when compared to human behavior in gathering areas. However, this was intentional as it represents a dynamic scenario serving as a benchmark for our solution.

In these settings, we showed that our devised solution, enforcing node cooperation does provide guarantees of data availability. Control of data distribution makes the system more resilient to mobility at the same time pushes data near the users, effectively easing data availability in the AC. Node density in the AoI is an important factors impacting both the studied metrics. Yet, the protocol *modus operandi*, requiring periodic transmissions to reserve the communication slot proves energy-consuming. Toward this issue we incorporate into the protocol the logic of electing sender-nodes amongst those with a higher energy resource partially addressing this issue. An adaptable slot length mechanism, adapting to the observed mobility dynamics could aid into this direction.

6

Conclusions and Future Work

In this work, we aimed to go beyond infrastructure supported communication platforms and produce technology that will enable the Internet of Everything, allowing sharing and distribution of content among mobile disconnected devices without strict infrastructure reliance. Indeed, with interconnected devices that have recently surpassed in number the people on Earth, our lives are currently immersed in a digital fountain with information being produced everywhere around us. A user with a handheld or wearable device, equipped with sensing and communication capabilities can now be both producer and consumer of information and services. We pursued our investigation toward a provider-less, infrastructures-less communication platform by means of three applicative scenarios, with emphasis on the networking techniques aimed at supporting them. This approach gave us the opportunity to continuously improve our comprehension of the actual situation, producing technology that could be used in synergy to help sustain a urban-wide opportunistic communication platform.

6.1 Summary of Results

The distributed and dynamic nature of opportunistic communications demands for networking techniques able to cope with mobility and the unpredictability of contact opportunities. Yet, content producers and consumer might be spatially and temporally decoupled, hence never connected at the same time in the same network. Redundancy and context-related information are important building blocks that could help tackle the issues, exploring multiple paths toward a destination while at the same time control system redundancy.

In this context, we started by exploring a content sharing solution among mobile devices without any infrastructure reliance. How search and retrieval could be orchestrated in this human-comprised network is what we investigated. Confining the search and retrieval to a users immediate vicinity or context he/she resides in, relates to a constrained data horizon they could explore. Instead, we depart from a synchronous to an asynchronous, store-and-forward communication model resembling that pioneered by the original outer space DTNs. Our retrieval scheme, coined *delegation-forwarding*, gives a node the capability to explore and reach data content available elsewhere, in other connected portions of the network. Our proposal exploits redundancy and contact frequency to help address the unpredictability of human behavior. We evaluated our proposal under different mobility scenarios, employing synthetic and field gathered mobility traces, showing the efficiency of our delegation scheme when compared to other approaches. However, the feasibility of an urban-wide, real deployment for this applicative scenario as is, is undermined by its retrieval times, directly influenced by the number of participants.

Departing from a node-based communication, we proposed a named-data oriented approach of the former solution tailoring its design to the NDN architecture. Compared to classical approaches in the NDN domain we depart from a name-based data communication to hybrid communication scheme, whereby node-based communication is employed when the forwarding mechanism is employed. In the evaluation phase, both system designs showed similar performance trends. Intuitively, the NDN paradigm given its *modus operandi* should have out-performed the other. This similar behavior is to be attributed to the nature of contact patterns exhibited by the traces, in particular while referring to the synthetic data trace. The sparse environments we considered do not give much space to the caching strategy to show its benefits in time.

Helping boost the performances of the former scenario, we set on a trial to investigate and include into the picture more predictable actors in the forwarding process. We proposed Mobile Delay/Disruption Tolerant Network: a delay tolerant application platform built on top of a Public Transportation System (PTS) and able to provide service opportunistic connectivity. Differently from previous work, we applied our solution to two real city environment: Milan (Italy) and Chicago (Illinois), extending the protocol *modus operandi* to a more complex and realistic usage scenario. We evaluated our proposal with different routing strategies and by means of simulation we demonstrated that MDTN could be a viable solution for push-based advertisements dissemination and non-critical data retrieval. Nevertheless, performance indexes of the considered routing policies showed that there is no golden rule. Forwarding between infrastructure deployed gateways does actually make a difference in the service provisioning. Single-copy routing approaches are more favorable in highly connected environments due to their reduced resource usage, good delivery rate, and reasonable delay. On the other hand, multi-copy routing approaches have to be preferred in loosely connected environments despite the fact we still need to understand if the added resources usage is worth the performance improvement.

In the previous scenarios, content producers are spatio/temporally decoupled from consumers and multi-hop routing/forwarding is required. However, in some scenarios data is of local relevance, confined to the context they are produced and could be consumed just by being in proximity with it. One could imagine the data moving back and forth, from user to user, confined in the interest area, hence the coined name of floating data. We identified data survivability and access as key features at enabling this scenario. To this end, we proposed a mechanism built upon existing solutions, providing the grounds for node cooperation in a dynamic, volatile environment. Preliminary evaluation of the devised protocol shows that the mechanisms put in place do indeed provide guarantees of data survivability in the anchor area. The algorithm has the capability to control the spatial distribution of data effectively addressing data access concerns. This said, we stated that there is indeed another issue which needs to be tackled, that of AC interference which in our controlled simulation environment was prevented from occurring.

6.2 Future Work

Possible directions and future works are planned for several solutions presented in this thesis.

Concerning the content sharing scenario our current proposal exploits contact history so as to to help guide the forwarding process but this process is agnostic about the data. An informed forwarding criteria including data into the picture could help improve its efficiency. A common approach in unstructured overlay networks is to employ gossiping algorithms and bloom filters to help prune the search graph. A similar approach could be used for our purposes to push data advertisements along the path toward the consumer. However, we do not deem these instruments as sufficient to help contain the unpractical retrieval profiles in our current network model. We plan to investigate in this direction in a more contained environment, studying instances of the initial problem such as in the vehicular environment.

Another direction we plan to investigate is the synergistic use of the proposals delineated in this work to help move toward an urban-wide, opportunistic communication platform. In the current study, changes in the PTS timetable are not contemplated. The floating data concept could be used to introduce resilience, counteracting unpredictable changes in the PTS timetable, causing request (responses) to queue up and thereafter be dropped. A more general scenario is presented when we start from a single requester point of view requesting data being delivered/anchored to some destination, and delivery is done by exploiting an urban backbone comprised of human-carried devices in synergy with the PTS entities.

In the MDTN scenario, the use of gateways deployed at end stations did prove improve performance and helped lower delivery times. In this direction, we would like to investigate a gateway positioning strategy and study the tradeoffs that emerge when considering the different routing policies. In addition, the results of our study showed that a hybrid forwarding strategy adapting to the specifics of the PTS under investigation is necessary.

Regarding the floating data scenario, in the current implementation we avoid ACs interference by controlling the simulation scenario. We proposed a potential solution whereby nodes are capable to notice the moment this phenomena occurs and deal with it by resorting to a decentralized merger procedure whereby the two ACs synchronize with one another. Another straight forward alternative is to park the conflicting nodes. One approach does not exclude the other and depending on the situation one might be most suitable and/or effortless when compared to the other.

Concluding, another interesting research direction, related to the broader domain of opportunistic communications, deemed worth pursuing is the study of the tradeoffs that arise in dissemination and transport of the data while employing routing and opportunistic forwarding versus approaches with no routing at all; which are effective under what conditions.

Bibliography

[1] Ahlgren, B., D'Ambrosio, M., Marchisio, M., Marsh, I., Dannewitz, C., Ohlman, B., Pentikousis, K., Strandberg, O., Rembarz, R., and Vercellone, V. Design considerations for a network of information. In *Proc. of ACM CoNEXT* (New York, NY, USA, 2008), ACM, pp. 66:1–66:6.

[2] Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D., and Ohlman, B. A survey of information-centric networking. *IEEE Communications Magazine 50*, 7 (2012), 26–36.

[3] Ahmed, S., and Kanhere, S. S. Cluster-based forwarding in delay tolerant public transport networks. In *Proc. of IEEE Conference on Local Computer Networks* (Washington, DC, USA, 2007), IEEE Computer Society, pp. 625–634.

[4] Angius, F., Gerla, M., and Pau, G. Bloogo: Bloom filter based gossip algorithm for wireless ndn. In *Proc. of ACM Workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications* (New York, NY, USA, 2012), ACM, pp. 25–30.

[5] Aschenbruck, N., Ernst, R., Gerhards-Padilla, E., and Schwamborn, M. Bonnmotion: a mobility scenario generation and

analysis tool. In *Proc. of the International Conference on Simulation Tools and Techniques* (2010), pp. 51:1–51:10.

[6] Baev, I., Rajaraman, R., and Swamy, C. Approximation algorithms for data placement in arbitrary networks. In *Proc. of ACM-SIAM Symposium on Discrete Algorithms* (2001), pp. 661–670.

[7] Balasubramanian, A., Levine, B., and Venkataramani, A. Dtn routing as a resource allocation problem. *SIGCOMM Computer Communication Review 37* (August 2007), 373–384.

[8] Banerjee, N., Corner, M. D., and Levine, B. N. Design and field experimentation of an energy-efficient architecture for dtn throwboxes. *IEEE/ACM Transactions on Networking 18* (April 2010), 554–567.

[9] Biswas, S., and Morris, R. Opportunistic routing in multi-hop wireless networks. *SIGCOMM Comput. Commun. Rev. 34*, 1 (Jan. 2004), 69–74.

[10] Boldrini, C., Conti, M., and Passarella, A. Contentplace: social-aware data dissemination in opportunistic networks. In *Proc. of International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems* (New York, NY, USA, 2008), ACM, pp. 203–210.

[11] Borgonovo, F., Capone, A., Cesana, M., and Fratta, L. Adhoc: a new, flexible and reliable mac architecture for ad-hoc networks. In *Proc. of IEEE Wireless Communications and Networking.* (2003), vol. 2, pp. 965–970 vol.2.

[12] Bruno, R., Conti, M., and Passarella, A. Opportunistic networking overlays for ict services in crisis management. In *Proc of International Conference on Information Systems for Crisis Response and Management* (Washington, DC, USA, 2008).

[13] Burgess, J., Gallagher, B., Jensen, D., and Levine, B. N. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *Proc. of INFOCOM* (April 2006), pp. 1–11.

[14] Burleigh, S., Hooke, A., Torgerson, L., Fall, K., Cerf, V., Durst, B., Scott, K., and Weiss, H. Delay-tolerant networking: an approach to interplanetary internet. *IEEE Communications Magazine 41*, 6 (June 2003), 128–136.

[15] Campbell, A. T., Eisenman, S. B., Lane, N. D., Miluzzo, E., and Peterson, R. A. People-centric urban sensing. In *Proc. of the Annual International Workshop on Wireless Internet* (New York, NY, USA, 2006), ACM.

[16] Carzaniga, A., Papalini, M., and Wolf, A. L. Content-based publish/subscribe networking and information-centric networking. In *Proc. of the ACM SIGCOMM Workshop on Information-Centric Networking* (New York, NY, USA, 2011), ACM, pp. 56–61.

[17] Carzaniga, A., and Wolf, A. Content-based networking: A new communication infrastructure. In *Developing an Infrastructure for Mobile and Wireless Systems*, B. König-Ries, K. Makki, N. Pissinou, S. Makki, and P. Scheuermann, Eds., vol. 2538 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2002, pp. 59–68.

[18] Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., and Scott, J. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing 6*, 6 (2007), 606–620.

[19] Conti, M., Giordano, S., May, M., and Passarella, A. From opportunistic networks to opportunistic computing. *IEEE Communications Magazine 48*, 9 (2010), 126–139.

[20] Conti, M., and Kumar, M. Opportunities in opportunistic computing. *Computer 43*, 1 (2010), 42–50.

[21] Corson, S., and J., M. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. RFC 2501, January 1999.

[22] Costa, P., Mascolo, C., Musolesi, M., and Picco, G. Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications 26*, 5 (2008), 748–760.

[23] Demers, A., Keshav, S., and Shenker, S. Analysis and simulation of a fair queueing algorithm. *SIGCOMM Comput. Commun. Rev. 19*, 4 (August 1989), 1–12.

[24] Demmer, M., and Fall, K. Dtlsr: delay tolerant routing for developing regions. In *Proc. of Workshop on Networked systems for developing regions* (New York, NY, USA, 2007), ACM, pp. 1–6.

[25] Ekman, F., Keränen, A., Karvo, J., and Ott, J. Working day movement model. In *Proc. of ACM SIGMOBILE workshop on Mobility models* (New York, NY, USA, 2008), ACM, pp. 33–40.

[26] et al., V. C. Delay-tolerant networking architecture. RFC 2501, January 1999.

[27] et al., V. C. Interplanetary internet (ipn): Architectural definition. Technical Report NDN-0005, IPNSig, October 2012.

[28] Fall, K. A delay-tolerant network architecture for challenged internets. In *Proc. of SIGCOMM* (New York, NY, USA, 2003), ACM, pp. 27–34.

[29] http://www.wirelessweek.com/articles/2010/02/femtocell-vs-wi-fi-da\delimiter"026B30Dta-offloading-indoor-coverage. Last visited: 28 October, 2013.

[30] Fiore, M., Mininni, F., Casetti, C., and Chiasserini, C. To cache or not to cache? In *Proc. of INFOCOM* (2009), pp. 235–243.

[31] Gaito, S., Maggiorini, D., and Rossi, G. P. Bus Switched Networks: an Ad Hoc Mobile Platform Enabling Urban-Wide Communications. *Ad Hoc Networks Journal* (to appear 2012).

[32] Gerla, M., Lindemann, C., and Rowstron, A. P2P MANET's – New Research Issues. In *Perspectives Workshop: Peer-to-Peer Mobile Ad Hoc Networks - New Research Issues* (Dagstuhl, Germany, 2005), M. Gerla, C. Lindemann, and A. Rowstron, Eds., no. 05152 in Dagstuhl Seminar Proceedings, Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany.

[33] Ghodsi, A., Shenker, S., Koponen, T., Singla, A., Raghavan, B., and Wilcox, J. Information-centric networking: seeing the forest for the trees. In *Proc. of ACM Workshop on Hot Topics in Networks* (New York, NY, USA, 2011), ACM, pp. 1:1–1:6.

[34] http://pcl.cs.ucla.edu/projects/glomosim/. Last visited: 11 December, 2013.

[35] http://transit.google.com/. Last visited: 11 December, 2013.

[36] Grassi, G., Pesavento, D., Wang, L., Pau, G., Vuyyuru, R., Wakikawa, R., and Zhang, L. Acm hotmobile 2013 poster: Vehicular inter-networking via named data. *SIGMOBILE Mob. Comput. Commun. Rev. 17*, 3 (Novemeber 2013), 23–24.

[37] Grossglauser, M., and Tse, D. Mobility increases the capacity of ad-hoc wireless networks. In *Proc. of IEEE INFOCOM* (2001), vol. 3, pp. 1360–1369 vol.3.

[38] Hossmann, T., Spyropoulos, T., and Legendre, F. Know thy neighbor: towards optimal mapping of contacts to social graphs for dtn routing. In *Proc. of the Conference on Information Communications* (Piscataway, NJ, USA, 2010), IEEE Press, pp. 866–874.

[39] Hui, P., and Crowcroft, J. How small labels create big improvements. In *Proc. of IEEE International Conference on Pervasive Computing and Communications Workshops* (2007), pp. 65–70.

[40] Hui, P., Crowcroft, J., and Yoneki, E. Bubble rap: social-based forwarding in delay tolerant networks. In *Proc. of ACM International Symposium on Mobile Ad Hoc Networking and Computing* (New York, NY, USA, 2008), ACM, pp. 241–250.

[41] `http://www.internetworldstats.com/`. Last visited: 28 October, 2013.

[42] Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H., and Braynard, R. L. Networking named content. In *Proc. of the ACM CoNEXT* (New York, NY, USA, 2009), pp. 1–12.

[43] Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H., and Braynard, R. L. Networking named content. In *Proc. of the ACM CoNEXT* (New York, NY, USA, 2009), pp. 1–12.

[44] Jain, S., Fall, K., and Patra, R. Routing in a delay tolerant network. *SIGCOMM Computer Communication Review 34* (August 2004), 145–158.

[45] jen Hsu, W., Dutta, D., and Helmy, A. Profile-cast: Behavior-aware mobile networking. In *Proc. of IEEE Wireless Communications and Networking Conference* (2008), pp. 3033–3038.

[46] Jetcheva, J., Hu, Y.-C., PalChaudhuri, S., Saha, A., and Johnson, D. Design and evaluation of a metropolitan area multitier wireless ad hoc network architecture. In *Proc. of IEEE Mobile Computing Systems and Applications* (October 2003), pp. 32–43.

[47] Johnson, D. The dynamic source routing protocol (dsr)for mobile ad hoc networks for ipv4. Technical Report NDN-0005, UCLA, October 2012.

[48] Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L. S., and Rubenstein, D. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. *SIGOPS Oper. Syst. Rev. 36*, 5 (Oct. 2002), 96–107.

[49] Jung, S., Lee, U., Chang, A., Cho, D.-K., and Gerla, M. Bluetorrent: Cooperative content sharing for bluetooth users. *Pervasive and Mobile Computing 3*, 6 (2007), 609 – 634.

[50] Klemm, A., Lindemann, C., and Waldhorst, O. A special-purpose peer-to-peer file sharing system for mobile ad hoc networks. In *Proc. of IEEE Vehicular Technology Conference* (2003), vol. 4, pp. 2758–2763 Vol.4.

[51] Koponen, T., Chawla, M., Chun, B.-G., Ermolinskiy, A., Kim, K. H., Shenker, S., and Stoica, I. A data-oriented (and beyond) network architecture. *SIGCOMM Comput. Commun. Rev. 37*, 4 (August 2007), 181–192.

[52] Lacage, M., and Henderson, T. R. Yet another network simulator. In *Proc. of the Workshop on NS-2: the IP network simulator* (New York, NY, USA, 2006), ACM.

[53] LEE, S.-B., WONG, S., LEE, K.-W., AND LU, S. Content management in a mobile ad hoc network: Beyond opportunistic strategy. In *Proc. of INFOCOM* (2011), pp. 266–270.

[54] LI, Y., SU, G., HUI, P., JIN, D., SU, L., AND ZENG, L. Multiple mobile data offloading through delay tolerant networks. In *Proc. of the 6th ACM workshop on Challenged networks* (New York, NY, USA, 2011), ACM, pp. 43–48.

[55] LINDGREN, A., DORIA, A., AND SCHELEN, O. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mobile Computer Communication Review 7* (July 2003), 19–20.

[56] LINDGREN, A., AND HUI, P. The quest for a killer app for opportunistic and delay tolerant networks: (invited paper). In *Proc. of ACM Workshop on Challenged networks* (New York, NY, USA, 2009), ACM, pp. 59–66.

[57] LIU, C., AND WU, J. An optimal probabilistic forwarding protocolin delay tolerant networks. In *Proc. of ACM International Symposium on Mobile Ad Hoc Networking and Computing* (New York, NY, USA, 2009), ACM, pp. 105–114.

[58] LIU, J., REN, F., MIAO, L., AND LIN, C. A-adhoc: An adaptive real-time distributed mac protocol for vehicular ad hoc networks. *Mobile Networks and Applications 16*, 5 (2011), 576–585.

[59] MAY, M., KARLSSON, G., HELGASON, R. O., AND LENDERS, V. A system architecture for delay-tolerant content distribution. In *Proc. of IEEE Conference on Wireless Rural and Emergency Communications*, October 2007.

[60] `http://www.msoon.com/LabEquipment/PowerMonitor/`. Last visited: 11 November, 2013.

[61] Mtibaa, A., May, M., and Ammar, M. On the relevance of social information to opportunistic forwarding. In *Proc of IEEE International Symposium on Modeling, Analysis Simulation of Computer and Telecommunication Systems* (2010), pp. 141–150.

[62] Nelson, S., Bakht, M., and Kravets, R. Encounter-based routing in dtns. In *Proc. of INFOCOM* (April 2009), pp. 846–854.

[63] `http://nsnam.isi.edu/nsnam/`. Last visited: 11 December, 2013.

[64] Oliveira, C. T. D., Braga, R. B., Taveira, D. M., Fern, N. C., and Duarte, O. C. M. B. A predicted-contact routing scheme for brazilian rural networks, 2008.

[65] `http://www.netlab.tkk.fi/tutkimus/dtn/theone/`. Last visited: 11 January, 2013.

[66] `http://www.o3bnetworks.com/`. Last visited: 28 October, 2013.

[67] Passarella, A., and Conti, M. Characterising aggregate inter-contact times in heterogeneous opportunistic networks. In *Proc. of International IFIP Conference on Networking* (Berlin, Heidelberg, 2011), Springer-Verlag, pp. 301–313.

[68] Pelusi, L., Passarella, A., and Conti, M. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *IEEE Communications Magazine 44*, 11 (November 2006), 134–141.

[69] Pentland, A., Fletcher, R., and Hasson, A. Daknet: rethinking connectivity in developing nations. *Computer 37*, 1 (January 2004), 78–83.

[70] Perkins, C. Ad hoc on-demand distance vector (aodv) routing. Technical Report NDN-0005, UCLA, October 2012.

[71] `http://www.google.com/loon/`. Last visited: 28 October, 2013.

[72] Qin, S., Feng, G., and Zhang, Y. How the contact-probing mechanism affects the transmission capacity of delay-tolerant networks. *IEEE Transactions on Vehicular Technology 60*, 4 (2011), 1825–1834.

[73] Quadri, C., Maggiorini, D., Gaito, S., and G.P., R. On the scalability of delay-tolerant routing protocols in urban environment. In *Proc. of Wireless Days WD* (October 2011).

[74] Ripeanu, M. Peer-to-peer architecture case study: Gnutella network. In *Proc. of International Conference of Peer-to-Peer Computing* (2001), pp. 99–100.

[75] Schollmeier, R., Gruber, I., and Niethammer, F. Protocol for peer-to-peer networking in mobile environments. In *Proc. of Internation Conference of Computer Communications and Networks* (2003), pp. 121–127.

[76] Scott, K., and Burleigh, S. Bundle Protocol Specification. RFC 5050 (Experimental), November 2007.

[77] Sede, M., Li, X., Li, D., Wu, M.-Y., Li, M., and Shu, W. Routing in large-scale buses ad hoc networks. In *Proc. of IEEE Wireless Communications and Networking Conference* (April 2008), pp. 2711–2716.

[78] Sede, M., Xu, L., Da, L., Min-You, W., Minglu, L., and Wei, S. Routing in large-scale buses ad hoc networks. In *Proc. of Wireless Communications and Networking Conference* (2008), pp. 2711 –2716.

[79] Silvis, J., Niemeier, D., and D'Souza, R. Social Networks and Travel Behavior: Report From an Integrated Travel Diary. In *Proc. of International Conference on Travel Behavior Research* (2006).

[80] Solis, I., and Garcia-Luna-Aceves, J. J. Robust content dissemination in disrupted environments. In *Proc. of ACM Workshop on Challenged networks* (New York, NY, USA, 2008), ACM, pp. 3–10.

[81] Sollazzo, G., Musolesi, M., and Mascolo, C. Taco-dtn: a time-aware content-based dissemination system for delay tolerant networks. In *Proc. of International MobiSys workshop on Mobile opportunistic networking* (New York, NY, USA, 2007), ACM, pp. 83–90.

[82] Spyropoulos, T., Psounis, K., and Raghavendra, C. Single-copy routing in intermittently connected mobile networks. In *Proc. of IEEE Sensor and Ad Hoc Communications and Networks* (2004), pp. 235–244.

[83] Su, J., Scott, J., Hui, P., Crowcroft, J., De Lara, E., Diot, C., Goel, A., Lim, M. H., and Upton, E. Haggle: seamless networking for mobile applications. In *Proc. of the International Conference on Ubiquitous Computing* (Berlin, Heidelberg, 2007), Springer-Verlag, pp. 391–408.

[84] Tang, B., Zhou, Z., Kashyap, A., and cker Chiueh, T. An integrated approach for p2p file sharing on multi-hop wireless networks. In *Proc. of IEEE Conference on Wireless And Mobile Computing, Networking And Communications* (2005), vol. 3, pp. 268–274 Vol. 3.

[85] `http://www.nsnam.org/docs/release/3.10/manual/singlehtml/index.html`. Last visited: 11 January, 2013.

[86] `http://www.netlab.tkk.fi/tutkimus/dtn/theone/`. Last visited: 11 December, 2013.

[87] `http://www.goingwimax.com/the-role-of-wimax-in-enabling-wi-fi-data-offload-networks-9406/`. Last visited: 28 October, 2013.

[88] Tseng, Y.-C., Ni, S.-Y., Chen, Y.-S., and Sheu, J.-P. The broadcast storm problem in a mobile ad hoc network. *Wireless Networks 8*, 2-3 (2002), 153–167.

[89] Vahdat, A., and Becker, D. Epidemic routing for partially-connected ad hoc networks. Tech. rep., 2000.

[90] Wang, J., Wakikawa, R., and Zhang, L. Dmnd: Collecting data from mobiles using named data. In *Proc. of Vehicular Networking Conference* (2010), pp. 49–56.

[91] Wang, L., Afanasyev, A., Kuntz, R., Vuyyuru, R., Wakikawa, R., and Zhang, L. Rapid traffic information dissemination using named data. In *Proc. of ACM Workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications* (New York, NY, USA, 2012), ACM, pp. 7–12.

[92] Wang, L., Wakikawa, R., Kuntz, R., Vuyyuru, R., and Zhang, L. Data naming in vehicle-to-vehicle communications. In *Proc. of IEEE Conference on Computer Communications Workshops* (2012), pp. 328–333.

[93] Wood, S. Mathewson, J. J. J. S. M.-O. K. M. Iceman: A practical architecture for situational awareness at the network edge. Technical Report, 2013.

[94] Xu, Q., Huang, J., Wang, Z., Qian, F., Gerber, A., and Mao, Z. M. Cellular data network infrastructure characterization and implication on mobile content placement. In *Proc. of ACM SIGMETRICS* (New York, NY, USA, 2011), ACM, pp. 317–328.

[95] Yin, L., and Cao, G. Supporting cooperative caching in ad hoc networks. *IEEE Transactions on Mobile Computing 5*, 1 (2006), 77–89.

[96] Yuan, Q., Cardei, I., and Wu, J. Predict and relay: an efficient routing in disruption-tolerant networks. In *Proc. of ACM International Symposium on Mobile Ad Hoc Networking and Computing* (New York, NY, USA, 2009), ACM, pp. 95–104.

[97] Zhang, X., Kurose, J., Levine, B. N., Towsley, D., and Zhang, H. Study of a bus-based disruption-tolerant network: mobility modeling and impact on routing. In *Proc. of ACM International Conference on Mobile Computing and Networking* (New York, NY, USA, 2007), ACM, pp. 195–206.

[98] Zhao, Q., Tong, L., Swami, A., and Chen, Y. Decentralized cognitive mac for opportunistic spectrum access in ad hoc networks: A pomdp framework. *IEEE Journal on Selected Areas in Communications 25*, 3 (2007), 589–600.

A

Appendix

Below are enumerated the scientific publications of the candidate related to the research activities discussed throughout this dissertation.

International Conferences and Workshops

(C1) M. Ambrosin, **A. Bujari**, M. Conti, F.D. Gaspari, C.E. Palazzi, “Smartphone and Laptop Frameworks for Vehicular Networking Experimentation”, in Proc. of *IFIP/IEEE Wireless Days*, to appear, Valencia, Spain, November 13-15, 2013.

(C2) **A. Bujari**, D. Maggiorini, C. E. Palazzi, L. Ripamonti, H. Sadushi, “Geo-Anchored Floating Data for Mobile Users”, in Proc. of *IEEE International Conference on Multimedia and Expo, Networking Issues on Multimedia Entertainment*, San Jose, CA, USA, Jul 2013.

(C3) D. Maggiorini, L. Ripamonti, **A. Bujari**, C. E. Palazzi, “Evaluating Design Constraints for Proximity-Based Games on a Real Urban

Topology", in Proc. of *IEEE International Conference on Multimedia and Expo, Networking Issues on Multimedia Entertainment*, San Jose, CA, USA, Jul 2013.

(C4) **A. Bujari**, "A Survey of Opportunistic Data Gathering and Dissemination Techniques", in Proc. of *IEEE International Conference on Computer Communications and Networks*, Munich, Germany, Jul. 2012.

(C5) **A. Bujari**, C. E. Palazzi, D. Maggiorini, C. Quadri, G. P. Rossi, "A Solution for Mobile DTN in a Real Urban Scenario", in Proc. of *IEEE WCNC Workshop on Wireless Vehicular Communications and Networks*, Paris, France, Apr 2012.

(C6) **A. Bujari**, B. Licar, C. E. Palazzi, "Movement Pattern Recognition through Smartphones Accelerometer", in Proc. of *IEEE Communications and Networking Conference*, Las Vegas, NV, USA, Jan 2012.

(C7) **A. Bujari**, C. E. Palazzi, D. Bonaldo, "Performance Evaluation of a File Sharing DTN Protocol with Realistic Mobility", in Proc. of *IFIP/IEEE Wireless Days*, Niagara Falls, ON, Canada, Oct 2011.

(C8) **A. Bujari**, B. Licar, C. E. Palazzi, "Road Crossing Recognition through Smartphone's Accelerometer", in Proc. of *IFIP/IEEE Wireless Days*, Niagara Falls, ON, Canada, Oct 2011.

(C9) C. E. Palazzi, **A. Bujari**, S. Bonetta, G. Marfia, M. Roccetti, A. Amoroso, "MDTN: Mobile Delay/Disruption Tolerant Network", in Proc. of *IEEE International Conference on Computer Communications and Networks, Workshop on Networking Issues in Multimedia Entertainment*, Maui, HI, USA, Aug 2011.

(C10) C. E. Palazzi, **A. Bujari**, "A Delay/Disruption Tolerant Solution for Mobile-to-Mobile File Sharing", in Proc. of *IFIP/IEEE Wireless Days*, Venice, Italy, Oct 2010.

(C11) C. E. Palazzi, **A. Bujari**, E. Cervi, "P2P File Sharing on Mobile Phones: Design and Implementation of a Prototype", in Proc. of *IEEE International Conference on Computer Science and Information Technology*, Beijing, China, Aug 2009.

International Journals

(J1) M. Gerla, D. Maggiorini, C. E. Palazzi, **A. Bujari**, "A Survey on Interactive Games over Mobile Networks", *Wiley Wireless Communications and Mobile Computing* , vol. 13, no. 3, Feb 2013.

(J2) **A. Bujari**, C. E. Palazzi, M. Roccetti, G. Marfia, "DTN Content Sharing among Commuters", *International Journal of Satellite Communications Policy and Management*, Inderscience Pub., vol. 1, no. 2/3, Oct 2012.

(J3) C. E. Palazzi, **A. Bujari**, "Social-Aware Delay Tolerant Networking for Mobile-to-Mobile File Sharing", *Special Issue on Interdisciplinary and Cross-layer Design of Mobile Social Networks and Wireless Networks*, International Journal of Communication Systems, Wiley, 2011.

Articles in Books

(B1) **A. Bujari**, C. E. Palazzi, "Intersection Collision: Causes and Avoidance Techniques", *Wireless Networks for Car Collision Avoidance*, Springer Ed., 2013, pp 189-227.