UNIVERSITÉ DU LUXEMBOURG

PhD-FSTC-2011-07
The Faculty of Sciences, Technology and Communication

DISSERTATION

Defense held on 30/06/2011 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG
EN INFORMATIQUE

by

Gabriel SANDULESCU
Born on 06 April 1968 in Bucharest, Romania

RESOURCE-AWARE ROUTING IN DELAY- AND DISRUPTION-TOLERANT NETWORKS

Dissertation defense committee:

Dr. Simin Nadjm-Tehrani, Supervisor
Professor, Linköping University, Sweden

Dr. Steffen Rothkugel, Chairman
Associate Professor, University of Luxembourg

Dr. Thomas Engel, Vice-chairman
Professor, University of Luxembourg

Dr. Silvia Giordano, Member
Professor, University of Applied Science (SUPSI), Switzerland

Dr. Joan Borrell, Member
Associate Professor, Autonomous University of Barcelona, Spain

ABSTRACT

Over recent years, the popularity of mobile networks has grown as a consequence of the ubiquity of communication devices. In many mobile networks, nodes can move freely and connect to their neighbours as they move into each other's radio range. However, due to high node mobility, limited bandwidth or energy, radio obstruction or malicious attacks, the route between source node and destination node may be disrupted frequently and for relatively long periods of time. Traditional adhoc networking models that assume continuous connectivity are largely incapacitated in such challenged environments. In response to that, a new class of networks, known as delay- and disruption-tolerant networks (DTNs), have emerged with applications in vehicular communications, emergency response, the military, and wildlife monitoring, to mention a few. This thesis undertakes a study of the resource-aware routing problem in DTNs along four lines.

First, the thesis presents the design and comparative evaluation of a delay-tolerant routing protocol (ORWAR) which optimises message transmission and bandwidth usage during opportunistic encounters. This is done through the estimation of contact duration between nodes, the selection of the most suitable message to forward at any contact opportunity, and message differentiation.

Second, the thesis proposes a hybrid scheme whereby opportunistic and infrastructure-based communication can be combined to overcome network partitions and packet losses. Collaboration between two networks characterised by different capacities, costs, and performance levels has been shown to be not only cost-effective, but also capable of increasing network survivability in the combined network.

Third, it proposes and evaluates a mathematical model and an algorithm that can be used to compute the optimal level of redundancy and replication of a routing protocol as a function of message characteristics, such as size and time-to-live.

Fourth, a holistic approach to resources is proposed, where variations in the spatial and temporal distribution of various resources can feed strategies to reduce resource consumption. Using estimates of vicinity resources, a routing protocol may not only use up fewer resources overall, but may also consume resources preferentially from nodes with higher resource levels, sparing whenever possible those with limited supplies.

ACKNOWLEDGMENTS

This section is dedicated to all the people who have contributed in various ways to making these last four years one of the most interesting and rewarding times of my life. First and foremost, I would like to express my sincere gratitude to my supervisor, Professor Simin Nadjm-Tehrani, for her constant support, guidance, and collaboration over the years. Despite the considerable distance separating Linköping and Luxembourg, she managed to make supervision effective and collaboration fruitful. I would also like to thank Simin for her confidence and the freedom she has given me in pursuing my research interests; it was a real privilege for me to be able to do research in such stimulating circumstances.

I would also like to extend my thanks to the members of my evaluation committee, Professor Steffen Rothkugel and Professor Thomas Engel, for all the support and encouragement they have given me since 2007. Kind thanks also to Peter Schaffer for his useful comments and suggestions concerning the two papers on which we worked together.

There is a group of people who also deserve special mention: my colleagues at the European Fund Administration Luxembourg, the company where I was employed on a full-time contract in January 1999, and later switched to a part-time contract after September 2007, when I started my PhD. I have had many great colleagues over the years, far too many to mention individually here, so I would like to extend a collective thanks to them for all their support and cooperation. I would also like to thank the EFA management, in particular Directors Jean-Marc Verdure and Emile Lutgen for allowing me to maintain collaboration with the company over this period. This eventually gave me the opportunity to pursue my studies while also being employed and, more importantly, while being able to keep in touch with the industry and the corporate world.

My thanks also go to the University of Luxembourg for the part-time assistant contract I was offered. It not only provided me with the financial support I needed in order to be able to conduct my research in comfortable conditions, but it also allowed me to acquire some valuable teaching experience. In that context, I would like to extend my thanks to Professors Pierre Kelsen, Sjouke Mauw and Lothar Baum, with whom I have worked over the years.

Furthermore, I would like to thank the technical and administrative staff at the University of Luxembourg and the CSC team for their help and support, as well as the LASSY team and my former and current office colleagues, among whom I will only mention Alfredo, Moussa and Marwane, although I will not forget any of them. A big thanks goes to the RTSLAB colleagues at Linköping University for the lively discussions and debates, as well as the agreeable working environment they created during my visits to Sweden. I am indebted to many of them for their comments, criticism, or simply the good time we spent together during *fika*[1] time.

Of course I owe a lot of gratitude to my family. Pursuing PhD studies at a fairly advanced age while also being the father of two wonderful children has not always been easy. I am deeply grateful to my wife Florina, my son Robert, and daughter Laura for their love and constant support.

Gabriel Sandulescu

[1]coffee break, a kind of social institution in Sweden

Contents

List of Figures

Chapter 1

Introduction

A basic tenet of most Internet protocols is that network nodes are connected most of the time. In practice, however, there is a whole class of networks where nodes are connected only *intermittently*. These networks are characterised by episodic meetings, sometimes only involving two nodes, thus failing to provide end-to-end connectivity between source and destination.

Episodic connectivity can be a result of high node mobility, low node density, intermittent power from energy management schemes (on/off), short radio range, radio obstruction, or malicious attacks. For this reason, an alternative network model looking for new ways to deliver services in disconnected environments emerged in the early years of this century, and has become known under the name of *delay- and disruption-tolerant networking* (DTN).

DTN proposes new design principles governing data transmission in heterogeneous networks that may lack continuous connectivity. Moreover, DTN also deals with cases where some links suffer from large propagation delays or high error rates, so that a typical end-to-end transport protocol such as TCP would fail. The challenged character of delay-tolerant networks makes them ideal for novel scenarios such as cooperative mobile environments, planned networks in space, defense applications, and disaster management. In these challenged environments, novel routing, transport and application layer algorithms have to be established in order to achieve efficient and reliable communication between DTN nodes. Despite the growing number of proposals formulated in this context, there is still little consensus over a set of suitable algorithms that can be applied to DTNs in general.

1.1 Motivation

In the late 20^{th} century, computational expenditure was dominated by computer hardware costs. In the increasingly interconnected world of the 21^{st} century, weight has shifted to the cost of running the machines. More specifically, an important cost component is now the cost of maintaining connectivity between the machines, which thus act as network nodes. Engineers are concerned with the size of the resource footprint used in the networking process, knowing that the relative importance of this size in the general computation equation is high. It was estimated [LZYZ07] that in a wireless sensor network it takes about 1000 times more energy to move a data byte around than it does to do a computation with it once it arrives. Moreover, in emerging distributed architectures, such as grid or cloud computing, the time taken to complete a computation is currently limited by how long it takes to do the moving [Pal10]. Therefore, creating routing algorithms which allocate available resources efficiently in order to achieve delivery goals (such as given delivery latency and delivery ratio) remains a subject of research in the field of networking and is particularly topical in DTNs.

Over recent years, the popularity of mobile computing has increased also due to the steady decrease in the price of communication devices. The most convenient way for interconnecting these mobile devices is via wireless networks with or without fixed infrastructure. In some cases infrastructure networks such as cellular or access points cannot be used either because of the high cost (networks in rural areas in development countries), because of security issues (military networks), or because this infrastructure simply does not exist at a particular location (disaster response networks, wildlife tracking networks). Thus, mobile nodes may be required to communicate directly with each other without using the infrastructure, and together build what has become known as *mobile ad-hoc networks* (MANETs).

Mobile ad hoc networks are composed of several mobile nodes that can communicate together without centralised control or an established infrastructure, as long as nodes are within each other's radio ranges. MANETs have been studied extensively as fully connected networks; there are cases, however, where node density in these networks is insufficient to provide an end-to-end route; in these cases a meaningful approach would be to model them as delay-tolerant networks. Potential situations range from public safety forces in a natural disaster to military forces on a battlefield, or simple city scenarios where people with different mobilities (either on buses, in cars or as pedestrians) communicate with each other. This thesis will focus on a subclass of DTNs, more exactly mobile adhoc networks with intermittent connectivity. Although the latter are only a subclass of DTNs, the two terms will be used interchangeably throughout this thesis.

The scenarios developed as part of this thesis consider DTN nodes to be mobile and to have wireless networking capabilities. Nodes are heterogeneous in type, and are capable of communicating with each other only as long as they are within each other's transmission range. The time during which nodes are within each other's transmission range is known as a *contact window*.

In most practical scenarios, and indeed in the case considered in this thesis, nodes have no prior knowledge about their future contact opportunities, and thus they form a so-called *opportunistic network* with nodes that are unaware about changes in network topology and message schedule. Such networks suffer from frequent connectivity disruptions, making the topology only intermittently and partially connected. Due to these disruptions, regular mobile ad-hoc networking approaches to routing and transport do not work, and new solutions need to be developed.

In a fully connected network such as the Internet, or in fully connected MANETs, *routing* means finding a path across the network along which packets will travel from source to destination. This does not apply however to delay- and disruption-tolerant networks, where nodes are only intermittently connected, therefore end-to-end connectivity cannot be guaranteed. In these cases even the most reliable and the most adaptive traditional routing protocols fail to deliver, the reason being the impossibility of identifying a contemporaneous end-to-end route. Therefore, in DTNs, nodes forward messages to custodians which use their mobility to deliver hop-by-hop messages towards their destination. This mechanism is known in the literature as the *store-carry-forward* mechanism. Moreover, messages can also be routed across the network using some redundancy or by splitting them into fragments, these mechanisms having a considerable impact on how system resources are used. Starting from these assumptions, this thesis will investigate the topics of routing, fragmentation and redundancy schemes, as well as resource availability in node vicinity.

1.2 Defining the problem

The aim of this thesis is to understand how resources available for intermittently connected DTN nodes may be leveraged in order to improve routing performance in DTNs. To begin with, it is important to mention that DTNs are considered to provide best-effort services, which means quality of service (QoS) is not inherently supported [Yan04]. There are no contemporary end-to-end paths in DTNs, network topology changes unpredictably [DH10], and control protocols deployed can have huge delays, making it impractical to control end-to-end flows [BJS06]. However, considering that DTN nodes usually have limited capacities available in terms of batteries, computation power, and storage, connectivity policies and strategies need to be defined in order to achieve objectives similar to what admission control, buffer management or packet scheduling do in traditional networks. In a DTN context, the mechanisms commonly applied in traditional networks are replaced to a certain extent by custodian election and queue management policies. *Custodian election* refers to a node's decision to choose one or more custodians from all those available, i.e. nodes with the highest probability to carry a message towards its destination. *Queue management* policies refer to a node's decision to prioritise the transmission of one particular message based on the knowledge available to that node. Queue management may also be responsible for deleting some messages from a queue when storage space for a node becomes short. The processes implementing these functions usually do not have access to global network knowledge and, because the environment is intermittently connected, they cannot rely on knowledge being quickly spread to nodes across the network.

Routing is still an open issue in DTNs. Routing strategies that make full use of both node properties and message properties are needed to ensure efficient message delivery. This task is inherently difficult, given the frequent and unpredictable changes in network topology, and the freedom for nodes to move and organise themselves in an arbitrary fashion. Based on analytical and experimental work, this thesis formulates routing solutions relying on information that nodes may collect autonomously and in a distributed fashion. The following issues are addressed:

- What is the geometry of node contact?
- How can link duration between nodes be modelled and what impact does it have on the volume of data delivered?
- When should a message be fragmented, how big should the resulting fragments be, and how much redundancy should be added to the protocol?
- In case there is also some infrastructure available, how can most of the routing be done over a DTN, and only a small residual part over the infrastructure, if that proves to be necessary in order to complete message transmission successfully?
- Is there a simple model that can be used for optimising message replication and fragment redundancy together?
- How can resources in the vicinity of a node be characterised from a quantitative and qualitative perspective, and what is their influence on routing performance?

1.3 Research challenges

This research, like some other studies in the field that will be described below, has been confronted with the following challenges:

1) **Limited connectivity**
Under the very limited connectivity that affects delay- and disruption-tolerant networks, node connections may appear rarely and they may be limited to connected pairs. This places a severe limitation on the extent to which distributed schemes can be deployed over this kind of network.

2) **Dynamic and unpredictable topology**
Network topology is highly dynamic, given that changes in node position are fast and unpredictable. Moreover, because of the limited connectivity, topology information between nodes cannot be updated in due time. This means that the majority of routing decisions must be determined locally or based on historic information.

3) **Heterogeneity of devices**
Nodes participating in DTN networks will differ in terms of their levels of storage, energy or communication resources. Routing algorithms need to make the most efficient use possible of these resources while also making sure that those in limited supply are not exhausted.

4) **Variable node population**
In many realistic scenarios nodes are expected to freely join and leave the network. This is usually associated with a more dynamic topology, a potential loss of messages due to custodian nodes leaving the network, and a need for nodes entering the network to be informed about available resources and network conventions.

5) **Variable and limited contact window**
Contact window or link duration is the time during which the connection between two mobile nodes is possible, this being a function of the nodes' wireless and mobility properties. As nodes move freely, contact windows will vary across a wide range of values.

6) **Evaluation**
Due to the difficulty and cost of deploying real DTN networks, most evaluations of routing protocols have been done through simulation. This has been the case for this thesis, but also for the majority of related work. Simulation has the advantage of allowing for the creation of hypothetical scenarios and replicable experiments, as well as the

ability to quickly explore a large design space. In addition to simulation, some of the algorithms presented in this thesis have been implemented on an Android platform, but the number of devices available for this real-life implementation has been a significant constraint in the evaluation performed subsequently.

1.4 Contributions

The contributions of this thesis are the following:

I) **A new opportunistic routing algorithm** using Window-Aware Adaptive Replication (ORWAR). This routing protocol exploits the store-carry-forward mechanism in delay-tolerant networks and combines selected replication and delivery acknowledgements with two novel features in a DTN context: (1) the use of message differentiation in the selection of replicated messages as well as deletion from the message queue, and (2) the use of estimated contact window for selecting the optimal message to forward at any contact opportunity. This routing scheme is well-suited for a wide range of message sizes and heterogeneous relative speeds between nodes and it can contribute to reducing the number of partial transmissions.

II) **A combined fragmentation/redundancy scheme** that has been added to ORWAR. It not only improves delivery ratios but, if residual infrastructure is available, it also allows messages to be completed by pulling dropped fragments.

III) **An optimisation framework** for optimising network cost when using message replication jointly with message erasure coding. There are two alternatives proposed for optimisation: 1) a purely analytical method, using fragment delivery distribution when delivery times can be accurately estimated, or 2) a distributed method based on acknowledgements, using message delivery history to feed the optimising framework when the protocol is too complex to be studied analytically or when heterogeneous mobility patterns are considered.

IV) **A collaboration scheme** for calculating the level of resources available in the vicinity of a node. A generic model for resources is proposed, then applied to individual network assets: energy, buffer space, and bandwidth. Knowledge about the resources available in the vicinity allows nodes to implement meaningful custodian election and queue management strategies. These strategies can be approached from a holistic perspective based on availability of the three resources under consideration in node proximity. Based on this information, a routing protocol may not only use up fewer resources overall, but may also consume resources preferentially from nodes with higher resource levels, sparing nodes with limited supplies as much as possible. As a result, disparities in available resources across the node population are significantly reduced, and nodes are less likely to leave the network as a consequence of resource depletion.

1.5 Thesis outline

The thesis is organised as follows. This introductory chapter is followed by Chapter 2, which contains an overview of DTN routing and discusses some common techniques applied. Chapter 3 presents ORWAR - a contact window aware routing algorithm designed for heterogeneous message sizes in opportunistic networks. Chapter 4 presents and evaluates a combined fragmentation/redundancy scheme which not only improves delivery ratios but, if infrastructure is available, also allows messages to be completed by pulling dropped fragments. Chapter 5 introduces an optimised model allowing parameters to be selected either analytically or based on delivery history. Chapter 6 proposes a distributed scheme for calculating the level of resources available in the vicinity of a node, as well as a set of policies that can rely on the level of resources estimated by using this scheme for selecting a suitable custodian (custodian election) and for choosing the right message to transfer (queue management). Finally Chapter 7 summarises the findings and outlines future work.

1.6 List of publications

Some of the contributions presented in this thesis have appeared in the following publications:

(I) Gabriel Sandulescu and Simin Nadjm-Tehrani, "Adding Redundancy to Replication in Window-aware Delay-tolerant Routing," Journal of Communications, *Special Issue on Delay Tolerant Networks, Architecture, and Applications*, vol. 5, no. 2, pp: 117–129, Academy Publisher, March 2010

(II) Gabriel Sandulescu and Simin Nadjm-Tehrani, "Opportunistic Routing with Window-aware Adaptive Replication," in *Proceedings of the* 4^{th} *Asian Conference on Internet Engineering*, pp: 103–112, ACM, November 2008

(III) Gabriel Sandulescu and Simin Nadjm-Tehrani, "Optimising Replication versus Redundancy in Window-aware Delay-tolerant Routing", in *Proceedings of the* 3^{rd} *Inter-*

national Conference on Communication Theory, Reliability, and Quality of Service, pp: 191–201, IEEE, June 2010.

(IV) Gabriel Sandulescu, Peter Schaffer and Simin Nadjm-Tehrani, "Vicinity Resource Cartography for Delay-Tolerant Networks: a Holistic Perspective", 3^{rd} *IFIP Wireless Days Conference '10*, pp: 1–7, IEEE, October 2010.

(V) Gabriel Sandulescu, Peter Schaffer and Simin Nadjm-Tehrani, "Exploiting Resource Heterogeneity in Delay-Tolerant Networks", *Wiley Wireless Communications and Mobile Computing, Special Issue on Mobility in Challenged Environments: Models, Protocols, Applications*, (under review - expected July 2011).

Chapter 2

DTN preliminaries

This chapter aims at providing a preliminary perspective on routing in Delay- and Disruption-Tolerant Networks (DTNs), and more particularly on resource-centric routing, the area of research to which this thesis belongs. The chapter defines and describes the main concepts and mechanisms making up the theoretical framework of the thesis, and is conceived as a guide to facilitate the understanding of subsequent chapters. While this chapter only provides a brief overview of related work, a more extensive coverage of the literature relevant to each line of study will be included in the following chapters.

Section 2.1 of this chapter presents the generic architecture proposed for DTN networks by the Delay Tolerant Network Research Group (DTNRG) [DTN03], which has come to be widely accepted by the scientific community. This is basically an overlay architecture that sets the common ground on which every delay-tolerant routing protocol would operate. Section 2.2 then discusses the most common primitive functions that can be assembled by a generic routing protocol working under the assumptions of a DTN architecture. Then, Section 2.3 explores various methods proposed for maximising the knowledge a node can acquire autonomously from its environment. Section 2.4 is dedicated to providing an overall perspective on the key topic of this thesis: resource-aware routing mechanisms in DTN networks. This section also presents some insights on how relevant challenges in a DTN context call for new, resource-centric strategies.

2.1 Routing in the absence of an end-to-end route

2.1.1 Challenged environments

Traditional networks, such as the current Internet or fully connected MANETs, are based on the implicit assumption that nodes are connected most of the time. In fact, not only end-to-end connectivity is assumed, but also short round trip times, symmetric data rates, as well as low error rates. The research area targeted in this thesis concerns scenarios that differ considerably from those involving traditional networks in a number of ways. These environments, also known as *challenged environments* [Fal03], are characterised by one or several of the following attributes [War03]:

- *Intermittent connectivity*: There is no contemporaneous path from the source node to the destination node, therefore the network is said to be *partitioned*. There may be quite a few reasons behind this intermittence: high node mobility or network sparsity, radio obstruction, intermittent power supply from energy management schemes (on/off), or malicious attacks.

- *Long and variable delays*: There is an assumption that long and variable propagation delays (minutes, hours, or even days) may arise between nodes. As a consequence, routing protocols that rely on a quick return of data acknowledgements will fail.

- *Asymmetric data rates*: Although the Internet may accommodate moderate asymmetries such as in the case of ADSL (Asymmetric Digital Subscriber Line), it can fail when affected by substantial imbalances between upstream and downstream data rates. Examples can be found in some satellite-based or military networks[1], where conversational protocols may fail on account of similar data rates flowing in both directions.

- *High error rates*: Some links may have high error rates. In this case either they require more processing - as there may be bit errors at those links, or they require more network traffic - as an entire packet may need to be retransmitted.

In addition to the sparse MANETs scenario that was introduced in the previous chapter, the list below contains some examples [Fal03] of real-life scenarios falling under the challenged networks model:

- *Exotic media networks*: These networks are based on links running in exotic media, for instance under water or in deep space. In underwater networks, communication takes place through wireless acoustic modems characterised by severely limited bandwidth, long propagation delays, and high error rates. In deep space communication, where one end node is in space, and the other is on Earth, networks may suffer outage due to environmental conditions (weather may impact visibility). Moreover, nodes can only exchange information if they are within line-of-sight of each other, the exchange schedule being generally predictable, albeit discontinuous.

[1] The imbalance ratio for some satellite-based or military networks may be as high as $1 : 10000$, while the typical imbalance ratio for ADSL is about $1 : 10$

- *Military ah-hoc networks*: These systems operate in hostile environments where mobility, environmental factors, or malicious attacks may create partitions. Moreover, the heterogeneity of communication media and mobility types is an important constraint in these networks, as military networks may connect troops, aircraft, satellites, and sensors (on land, in space, or under water). These systems also have high requirements in terms of reliability and scalability.

- *Sensor/actuator networks*: In this kind of challenged networks, nodes have extremely limited power, memory, and CPU capabilities. Communication is often scheduled in such a way as to conserve power. As a consequence, specific protocols are developed to deal with connectivity disruptions and resource issues.

The routing problem in partitioned networks may at first appear as a standard problem of dynamic routing, but with extended link failure times [JFP04]. Therefore some early attempts were made to adapt routing protocols known to work well in fully connected MANETs to the field of intermittently connected networks. However, these attempts were largely unsuccessful because reactive routing protocols such as AODV (Adhoc On-demand Distance Vector [PR99]) or DSDV (Destination Sequenced Distance Vector [BMJ+98]) fail to discover a complete path due to frequent network partitions, while proactive routing protocols such as OLSR (Optimized Link-State Routing [Tøn04]) or DSR (Dynamic Source Routing [BMJ+98]) are overwhelmed by the rapid topology changes and therefore fail to reach a pertinent decision during the routing information exchange stage.

So then what alternatives are there to traditional routing in intermittently connected networks?

2.1.2 Store-carry-forward routing

An alternative to traditional routing that is well suited for intermittently connected networks appeared with the introduction of the new *store-carry-forward* paradigm, as explained by Fall [Fal03]: messages are *forwarded* (moved) or *replicated* (copied) to one or several relay-nodes in the network in order to increase the opportunity for those messages to reach their intended destination. Messages travel towards their destination taking advantage of node mobility - in other words, it is the nodes that *carry* messages across the network. This implies that an intermediary (also called *custodian*) node needs to provide enough buffer space to *store* these messages over a relatively long period of time. This is a brief description of the store-carry-forward mechanism through which messages can be delivered to destination even in the absence of an end-to-end route from source to destination. The context of social networks provides a similar mechanism that has also been described in the literature as Pocket Switched Networks (PSNs) [HCS+05], emphasising the fact that messages are carried towards their destination literally in the pockets of network actors.

It is also important to mention that, in the forwarding phase, messages can only be transmitted while nodes are in contact. A *contact window* is defined as the time during which two or more mobile nodes meet, i.e. when they are within each other's radio transmission range, and are able to transfer messages. Therefore mobility in challenged environments adds another constraint to the functioning of the store-carry-forward mechanism described above, as it may significantly affect the duration of a contact window, as well as the volume of data that can be transferred over that contact window. This is why it is interesting to analyse meetings between nodes, and the relationship between node mobility and message size, a topic that will be covered in Chapter 3 of this thesis.

Despite the slight semantic dissimilarity between the concepts of *store-and-forward* and *store-carry-forward*, the two terms are often used as synonyms in the DTN literature, since they actually serve the same purpose. This thesis will also reflect that usage, although the author's option clearly goes for *store-carry-forward*, as the more comprehensive and appropriate term.

2.1.3 The bundle protocol

The architecture [CBH+07] proposed by the DTN Research Group [DTN03] is an attempt to refine the OSI layered model in order to cope with partitioned networks. It introduces a new layer above the transport layer and below the application layer called *bundle layer*, allowing a non-conversational communication which is better suited for intermittently connected networks. This architecture defines how a set of DTN nodes can form a cooperative overlay using the store-carry-forward paradigm to deliver messages [FF08]. Every node acts as a router, like in the case of mobile ad-hoc networks, that have already been the subject of extensive research, with the difference that in this case nodes are connected only from time to time. The data units exchanged between nodes are called *bundles* and this name suggests a non-chatty, all-in-one approach. Bundles consist basically of two things: (1) the user data as constructed by the source application, (2) some control information (headers) describing how data should be handled from source to destination. Although bundle may be the most appropriate term to designate a data unit exchanged in a DTN context, in this thesis the terms bundle and message shall be considered equivalent and used interchangeably.

Figure 2.1 shows the protocol stack as defined in the DTNRG architecture [CBH+07] and represented in different works [War03, DH10]. Bundles are constructed at the level of the application layer, the aim being to restrict exchange to a minimal conversational

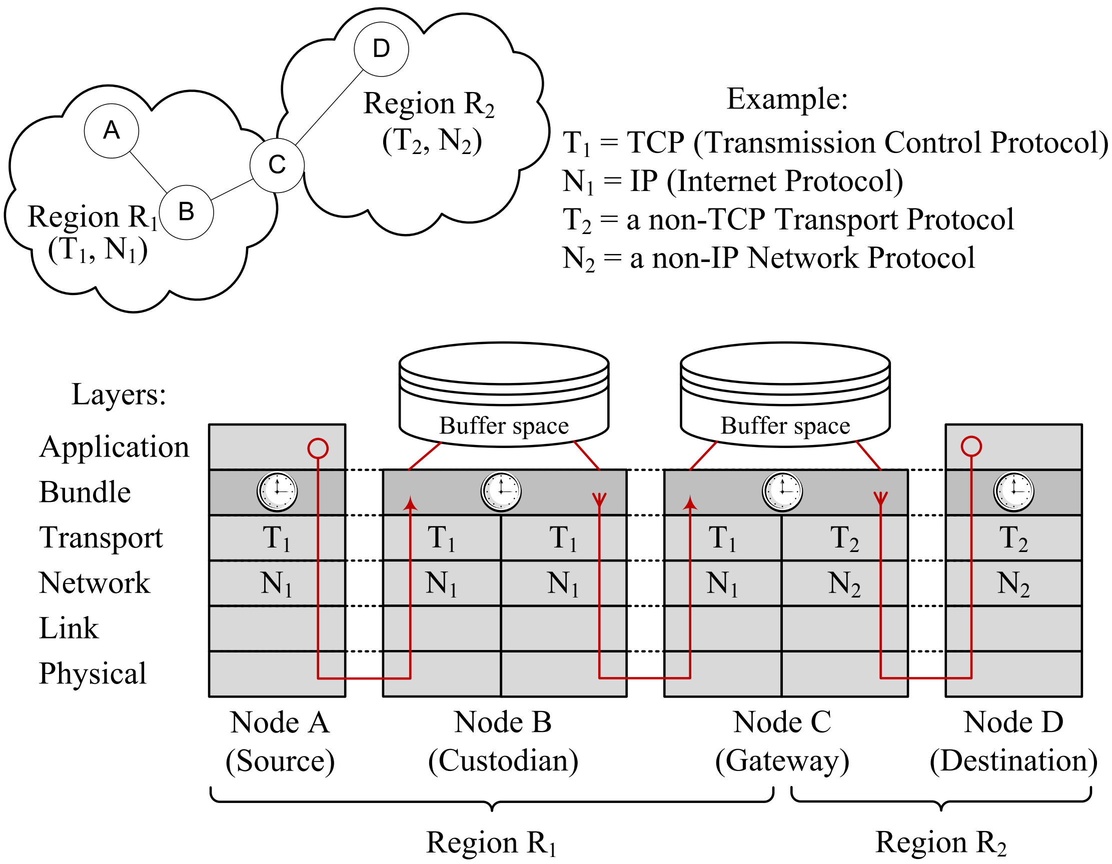


Figure 2.1 DTNRG reference stack

model. Therefore, DTN applications should be designed in such a way as to minimise the number of end-to-end transactions [Muk06], using self-contained messages. Then, bundles may be transferred towards destination over a number of custodian nodes by using a different protocol stack. DTN architecture neither recommends, nor restricts any lower layer protocols, but DTN research [BHT+03] shows that typical protocols used over the Internet, such as TCP, may fail in a partitioned network because of the long propagation delays.

Bundle layer architecture has been designed with interoperability in mind[2]. Therefore the architecture also defines the notion of networking regions with several DTN gateways

[2] On the other hand, some newly proposed architectural approaches, that will however not be analysed in detail in this thesis, start out from a different assumption. Instead of ensuring interoperability with older systems, their main goal is to possibly design a completely new communication system from scratch [JHF+08]. Rather than coming up with fixes to the current layered approach, they are proposing a new architectural framework based on ontologies, properties, means, and design patterns to support and sustain the long-term flexibility of the Internet (or other communication systems).

at their edges. Gateways act as proxy-agents [Fal03], their role being to secure the passage of bundles across various regions. DTNRG reference stack allow bundles to be carried across various regions, each region potentially having its particular protocol stack.

Let us consider, for instance, a bundle generated from a handheld device which is part of a sparse terrestrial mobile network meant to control an autonomous underwater vehicle for ocean monitoring. The bundle will find its way first through the sparse terrestrial region by using TCP over WiFi links. Then, it will reach the underwater gateway, and finally it will be transmitted over a very different set of physical, data link, network and transport layers corresponding to the undewater region [APM04].

The bundle layer [CBH$^+$07, War03] provides six classes of service (CoS) for a bundle:

- Custody transfer: Delegation of retransmission responsibility to an accepting node, so that the sending node can recover its retransmission resources. The accepting node returns a custodial-acceptance acknowledgement to the previous custodian.
- Return receipt: Confirmation to the source, or its reply-to entity, that the bundle has been received by the destination application.
- Custody-transfer notification: Notification to the source, or its reply-to entity, when a node accepts a custody transfer of the bundle.
- Bundle-forwarding notification: Notification to the source, or its reply-to entity, whenever the bundle is forwarded to another node.
- Priority of delivery: Bulk, Normal, or Expedited.
- Authentication: The method (e.g. digital signature), if any, used to verify the sender's identity and the integrity of the message.

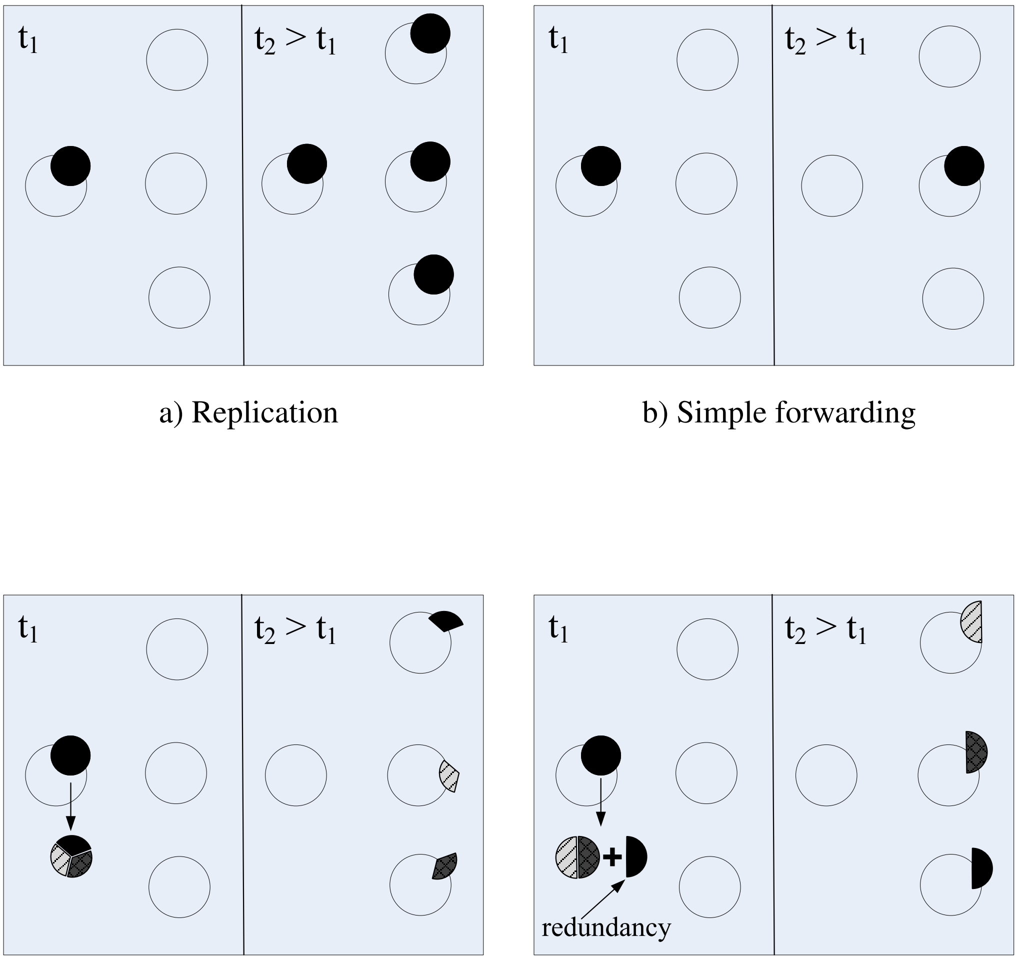
These classes of service propose a simple and coarse-grained architecture designed to provide a level of service similar to the postal system [Fal03]. Starting from priority of delivery, one direction for tackling resource allocation in DTNs would be to analyse message priorities. In traditional networks, allocating resources signifies implementing mechanisms such as flow control and congestion control in order to maximise the perceived quality of service for end users. In a partitioned network, however, the borderline between these two mechanisms fades; actually, these mechanisms require a similar approach that can be implemented via queue management policies. For instance, in the field of DTN Networks, flow control would signify that the sending rate of a DTN node should be correlated with the sending rate of the next hop, so that no excessive data accumulation should occur at one node. Correspondingly, congestion control would point to quite similar mechanisms that alleviate storage congestion when buffers are filled up with too many bundles [BJS06]. Mechanisms such as controlling the degree of data replication [TNB+10] have been proposed in order to implement congestion control. A node experiencing congestion has several options for mitigating the problem, such as: drop expired bundles, send some of the bundles to another custodian, refuse new bundles or refuse them with custody transfer notification, or drop unexpired bundles for which the node has custody. Choosing among all these options may not be an easy task. However, Chapter 3 will present some ideas on how message properties such as priorities, size, and time-to-live can be used together in order to perform a resource-aware differentiation among messages. Therefore, a decision to send or to delete a message from a queue is made locally and autonomously using algorithms derived from the message properties mentioned above. DTN architecture itself does not suggest any particular mechanisms for taking control over energy consumption, and this is an important issue in many DTN scenarios, especially when untethered devices are used. Chapter 3 and 4 will propose several mechanisms that can operate on top of the proposed DTN architecture and are designed to improve energy consumption by minimising partial transmissions.

2.2 Primitive functions on bundles

There are many different equivalent *terms* used for DTNs in the literature, such as partitioned networks, challenged networks, space-time routing, partially connected, or extreme networks (see Zhang [Zha06] for more extensive reference). Whatever the term used, what is important to recall is that routing in this kind of networks employs a store-carry-forward paradigm and that bundles (or messages) are exchanged between nodes only when contact opportunities arise, as shown in the previous section. This section further develops that topic and provides a description of some basic functions on bundles that are widely used in DTN routing such as replication, simple forwarding, fragmentation, and erasure coding. These basic functions act as building blocks within each DTN routing protocol and have been the subject of extensive study by Spyropoulos et al. in their routing taxonomy [SRT+10]. The description below will be limited to the functions actually used in the course of this thesis.

2.2.1 Replication

In many real-life scenarios involving delay- and disruption tolerant networks, node contact schedules are unknown. In order to cope with this uncertainty, a common technique is to send multiple copies of each message to a few custodians in order to increase the chance



a) Replication b) Simple forwarding

c) Fragmentation ($n = m = 3$) d) Erasure coding ($n = 3, m = 2, k_r = n/m = 1.5$)

Figure 2.2 Primitive functions on bundles

that at least one copy will be eventually delivered. This is known in the literature as *message replication* and is presented in Figure 2.2 a). Sending more copies not only increases the probability of delivery, but also decreases delivery latency. However, it also consumes resources (bandwidth, energy, as well as storage at custodians) depending on the number of copies sent.

The Epidemic protocol [VB00], also known under the name of ”greedy replication” or ”flooding”, was an early example where replication was used. The proposed strategy is very simple: each time a new node is encountered, it receives a copy of the message, in case the new node did not already have the message in its buffer. Although this simple strategy works well when message volume and node density are relatively low, protocol performance declines when it has to deal with high numbers of input messages (overloading problem) or it is deployed over networks with large numbers of nodes (scaling problem).

A first attempt to put a limit on greedy replication schemes was made by Grossglauser and Tse [GT02] who proposed a scheme where messages only make two hops at the most. Under this scheme, known as the two-hop scheme, the source node passes the message on to a custodian that holds that message while waiting for a contact with the destination node. The authors show that, in large networks, the overhead in a two-hop scheme can be reduced as compared to the overhead in greedy replication schemes at the cost of slightly increasing latency. This mechanism can be extended from 2 to k hops, and it is then called a k-hop scheme [ZNKT07].

Another scheme using replication, and implemented for the first time by Spyropoulos et al. [SPR05], overcomes the overloading and scaling problem of epidemic schemes by maintaining only a controlled number of copies for each message in the network. This scheme, called ”SprayAndWait”, works in two phases. In the first phase, called the *spray* phase, copies are forwarded to a number of nodes encountered, according to a protocol parameter *L* called *replication factor*. In the second phase, called the *wait* phase, nodes carry the copies until direct encounter with the destination in order to deliver the message.

Spyropoulos et al. also show that, in order to achieve a given delivery latency by using SprayAndWait [SPR05], the ratio between the replication factor and network size should remain approximately constant, even when network size varies over a wide range. This protocol also proposes an analytical model [SPR08a] for estimating delivery latency that will be described in detail and then built upon in Chapter 5. By varying the replication factor, the controlled replication scheme allows simple but effective performance tuning: on the one hand, it achieves low overhead but high latency when the replication factor is low, and on the other hand it achieves lower latency but higher overhead when the replication factor is high. By relating this performance tuning mechanism to message differentiation performed using message utility, Chapter 3 will propose a new routing protocol, thus introducing a new, more sophisticated replication type: the utility-based replication scheme.

2.2.2 Simple forwarding

Although replication is a widely used technique, not all DTN protocols need to rely on replication in order to achieve good routing performance. A whole class of protocols use simple forwarding instead of replication. *Simple forwarding* means that, instead of copying the message to be transmitted to the new custodian and keeping one copy for itself, a node will delete the original copy from its own buffers as soon as transmission has been completed correctly. The mechanism is depicted in Figure 2.2 b). In the literature, routing protocols using simple forwarding only are collectively referred to as the single-copy case [SPR08b], while those using replication are collectively referred to as the multi-copy case [SPR08a].

When a routing protocol makes exclusive use of simple forwarding, the obvious implication is that only one single copy of each message exists in the network. Therefore, simple forwarding minimises the overhead in the network by minimising the message multiplication overhead. Such a protocol is known to minimise buffer consumption at the cost of increasing delivery latency. Finally, the number of transmissions is not necessarily reduced as compared to the multi-copy case, as single-copy messages may be transmitted over a multitude of hops, including loops, in order to reach their destination.

The single copy case is the most common choice in traditional networks, such as the Internet or fully connected MANETs, as long as acknowledgements and retransmissions come at a reasonable cost. However, in the context of partitioned networks, this seems to be a riskier strategy, as the loss of the single existing copy of a message may jeopardise completely the delivery of that particular message. Moreover, as no contemporaneous end-to-end path exists, and knowledge about topology is in the generic case very limited, a node may encounter difficulties in selecting a single good custodian in a meaningful way. Therefore, single-copy protocols are to be found where the amount of knowledge can compensate for the lack of replication. Such scenarios may correspond to cases such as geographical routing [KNT08], where information about the localisation of the destination node is provided, gradient routing [OE08] or trajectory routing [NN03], where at least the sense of a trajectory between the source and the destination is given, so that the message is routed in a direction closer to the destination. These cases may also be found in scheduled environments, but since they are quite different from the scenarios to be considered in this thesis, simple forwarding will not be used in subsequent comparisons.

2.2.3 Fragmentation and erasure coding

The ability to fragment bundles, either prior to transmission (*proactive fragmentation*) or while in transit (*reactive fragmentation*) has been introduced early in the DTN design [FF08]. As bundle layers are agnostic about lower layers, it is the responsibility of upper layers (bundle layer, application layer) to limit the impact of a challenged environment by using smaller messages [PKO08]. Figure 2.2 c) depicts the fragmentation mechanism.

Sending smaller data units over challenged and opportunistic environments usually better accommodates cases when bandwidth, contact time and buffer space are limited because: 1) a smaller message can be transmitted more reliably over the limited contact time between two nodes, and 2) letting fragments take independent paths from source to destination may improve the reliability of the protocol, especially when some kind of fragment redundancy is put in place (see below). However, when a high level of fragmentation is used, the network will face an increasing reconstruction effort at destination and additional resources will be taken up by every fragment header. The bundle protocol does not set any limits on the transmitted data units. Moreover, there is no notion of Maximum Transmission Unit (MTU) defined in DTN. Chapter 4 will look at ways in which an appropriate fragmentation size can be found, by deducing it from other network parameters.

The vast majority of routing protocols in the literature (SprayAndWait [SPR05], MaxProp [BGJL06], RAPID [BLV07]) consider bundles as indivisible, which makes routing decisions simpler. Jain et al. [JDPF05] formulate the problem of optimising the probability of successful message delivery by *erasure coding*. The idea behind erasure coding is that, instead of sending a full copy of a message over a relay, only a fraction of the code-blocks are sent over each relay. However, transferring a large amount of data in small fragments without explicit acknowledgements may lead to a degradation of reliability in best-effort networks, such as DTNs, because all fragments need to arrive at destination in order for the initial message to be reconstructed. A *redundancy factor* k_r can then be chosen so that a message of size s is split into n, s/m-sized blocks ($k_r = n/m, k_r \geq 1$). Encoding is usually based on Reed-Solomon [Pla97] or Tornado [BLM99] codes which allow the reconstruction of the original message from any m different fragments arrived at destination. Therefore, fragmentation can be considered as a particular case of erasure coding with the redundancy factor $k_r = 1$.

Erasure coding may increase system resilience by improving load balancing among the multiple paths taken by each fragment. The main mechanism is depicted in 2.2 d). It is particularly relevant in DTNs, since original messages can be large in size and may not be sent over a single contact. Still, some questions need to be answered, such as what the ideal size of a fragment is, given a certain mobility, and what redundancy needs to be added in order to achieve some delivery parameters such as given latency. Resource optimisation becomes even more critical when the redundancy introduced in the system by erasure coding is augmented by a routing protocol using message redundancy. However, the main advantage of a strategy including erasure coding is that coding can be performed only at source, and decoding is done only at destination. This means that erasure coding is not involved in routing, therefore the scheme can be applied to any store-carry-forward protocol. Consequently, this mechanism not only increases network reliability, but also solves the problem of delivery over a limited contact window, since smaller fragment sizes are used.

Chapter 5 also proposes an optimisation framework allowing joint message replication and fragmentation, coupled with some redundancy mechanisms, with quantifiable effects on network latency.

2.3 Exploiting knowledge

The amount of information considered as known about a network may exert a considerable influence on the type and performance of routing strategies to be deployed in that network. In their book about DTN, Farell and Cahill [FC06] proposed a graphic representation of routing algorithms arranged around a circle according to the amount of information available, depicting the two extremes that go from acquiring full knowledge about a network to making heuristic estimates. In their survey, Jones et al. [JLSW07] also mention that one way to optimise routing performance in a network is to make full use of the knowledge available.

Knowledge about a network may range between total knowledge and no knowledge at all. Having total knowledge about a network means having specific information about message workload, contact schedule, and resource distribution in the network. The strategy that corresponds to the full knowledge case is known in the literature as *Oracle routing*. This is a simple strategy, hard-coded in advance and best adapted for all nodes in the network, which usually requires minimal configuration and control messages. Of course, such a strategy cannot dynamically adapt to variable network conditions, so it immediately makes sub-optimal routing decisions when network parameters deviate from baseline settings. Moreover, in most practical cases it is rather unrealistic to assume total knowledge about message workload, contact schedule, and resource distribution. Considering the above, this thesis aims at developing adaptive protocols that can demonstrate their robustness in various scenarios, and can adapt to a wide range of parameter variations.

2.3.1 Knowledge about topology and traffic

In some particular delay- and disruption-tolerant scenarios, contact schedules between nodes may be known in advance. For example, in satellite or interplanetary communication, the trajectory of a satellite and therefore its contact opportunities may be calculated in advance based on point-to-point visibility, eclipses or maintenance schedules. The cases mentioned above can be considered as instances of *scheduled* DTN scenarios.

Things may happen somewhat similarly in the case of vehicular scenarios, where bus schedules, for instance, are - up to a certain point - known in advance. However, a bus can be delayed by a traffic jam or it may be slightly ahead of time, generating a deviation from its expected schedule. The cases that are reasonably regular, although they are not based on precise timetables, are known in the literature as *predicted* DTN scenarios.

In all other cases, that are however typical for real-life contexts, network contact schedules are unknown, and this is the case of *opportunistic* scenarios. Although most of the results in this thesis may also work in scheduled or predicted environments, the main focus will be the generic case, i.e. the opportunistic assumption.

In delay tolerant networks, the most efficient routing algorithms can be proposed in conjunction with the amount of knowledge available on network topology and traffic demand [FF08, Zha06, JLSW07]. One intriguing question would be whether it is knowledge on network topology, or knowledge on traffic demand that should prevail, and how this would affect an Oracle-based routing protocol.

Balasubramanian et al. suggested in [BLV07] that an Oracle routing having complete traffic knowledge but no topology knowledge would yield better results in terms of delivery ratio than a similar routing having total topology knowledge but no traffic knowledge.

Refining the information available for a routing protocol, Jain et al. [JFP04] defined four types of central knowledge:

Topology	1. *Contacts Summary*	contains aggregate information about contacts, such as number of meetings for every node (that is, with no time-varying information)
	2. *Contacts*	contains information about contacts between two nodes at any point in time (equivalent to knowing the contact schedule completely)
Traffic	3. *Queuing*	gives information about instantaneous buffer availabilities at any node at any time
	4. *Traffic Demand*	contains information about present or future traffic demand

Based on the availability of these types of knowledge, the authors [JFP04] propose several algorithms, starting from a linear programming approach when all information (from 1. to 4.) is available, continuing with a modified Dijkstra approach in case only information under point 2. is available, and finishing with a simple forwarding approach when no knowledge is available. Despite the real theoretical interest of this approach, such detailed information very rarely happens to be at hand. Complete contact information may be known in some scheduled DTN senarios, but certainly not in the case of opportunistic scenarios - that are the focus of this thesis. Having complete queuing and traffic demand information available as central knowledge appears to be even more unlikely.

Knowing the topology of a network may help estimate the resources available for a particular node or associated with a particular region. Topology is rarely known beforehand, but it can be investigated by nodes autonomously by acquiring and exploiting knowledge in existing pockets of connectivity using, for example, token-traversal strategies. Piyatumrong et al. have studied various connectivity tree traversal algorithms [PBGL08, PBGL09], but these distributed algorithms make sense only when connectivity pockets contain a sufficient number of nodes. These strategies obviously cannot be used when nodes meet only in pairs, a scenario that corresponds to very low node density, which is however one of the scenarios this thesis will try to explore.

There is still an important potential in exploiting topology and traffic knowledge in a routing algorithm, assuming that nodes are able to collect information themselves during encounters with other nodes, that is, without central knowledge. There are three implications to consider in relation to this approach: 1) the information so gathered will be based on heuristics, 2) because of the limited connectivity, the information so gathered will be relevant for a Euclidean vicinity of the node, depending on node mobility, and 3) historical meetings will be used to extract information. Chapter 6 presents a distributed scheme that can estimate the available resources in the vicinity of a node by using information exclusively gathered from peer nodes.

2.3.2 Knowledge about mobility

Under the opportunistic assumption, the actual trajectory of a node in the network is unknown; on the other hand, however, node mobility patterns are usually known in advance, as nodes observe a certain *mobility model*. Mobility models considerably affect *network capacity*, defined as the amount of traffic a network can handle over a particular time period. Moreover, the movement model that captures node behaviour in real usage scenarios is therefore needed for a reliable assessment of a new routing protocol.

There are two classes of mobility models that have been proposed in the literature:

1) *synthetic models*, such as the random waypoint model [HLV06], or random direction [Bet01], that are both generic high level models aiming at producing movement that is accurate enough but generated analytically from relatively simple mathematical functions, and

2) *circumstantial models*, such as the Working Day Movement Model [EKKO08] or Bonn Motion [AGPG+07], that are more reality-oriented and describe incidental scenarios,

generally aiming at a more accurate depiction of specific devices used in such real usage scenarios.

Mobility has a major impact on routing in DTNs, as messages use node mobility to travel towards their destination. Moreover, relative node speed affects the accumulated size of the messages that can be transmitted over given contacts. Finally, the social affinity between nodes, as captured by mobility patterns, significantly affects the distribution of contact opportunities in time and space. In the complex picture involving mobility, some contradictory epiphenomena may arise. For example, when network mobility increases, connectivity in that network may also grow. However, increased mobility will also result in the shrinking of the average contact window, which makes the handling of bigger messages more difficult.

Some recent studies have focused on the idea of social correlation between nodes [YGC09], as well as on the particular movements building up higher densities around points of interest [PSDG09]. In a city context, for instance, it is easy to imagine that nodes visiting restaurants, railway stations or simply walking in the downtown area are likely to exchange more data than nodes with a less "social" trajectory. Moreover, these exchanges are also time-dependent, for instance the members of a family will exchange more data after hours than during working time, such as in the Working Day Mobility Model [EKKO08].

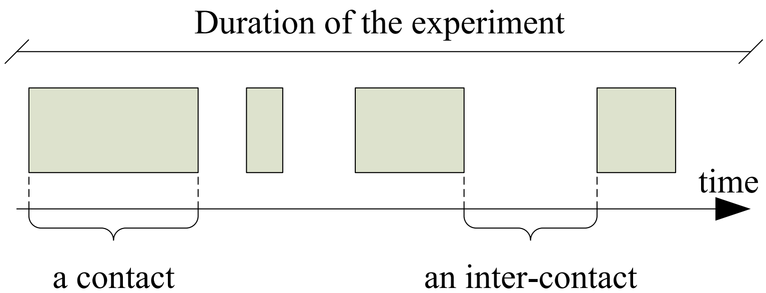


Figure 2.3 Contact and inter-contact times for a pair of nodes

The distribution of contact window time in a network is of course a result of the mobility model and is an important factor in determining the capacity of that network [CHC$^+$05]. Beside the distribution of contact window times, another relevant effect of the mobility model is the distribution of inter-contact times. *Inter-contact time* is a common mobility property defined as the time lapse between two successive contact windows for a given node pair. Contact and inter-contact times are represented in Figure 2.3.

Inter-contact time (also known as *inter-meeting time*) can exert a considerable influence on latency in partially connected networks [YGC09, CE08], and therefore has been much studied either analytically or experimentally [ZNKT07, GNK05]. Assuming that either messages are negligible in size or bandwidth between nodes is infinite, inter-contact time would be enough to evaluate latency distribution in a homogeneous network [GNK05, SPR08b, SPR08a]. However, in real-life scenarios, large messages compete with each other for limited bandwidth, therefore the assumptions above rarely hold in practice. This means that a thorough assessment of network performance (such as latency or capacity) would be incomplete if it failed to take into account both aspects of mobility, quantitative (number of meetings, represented by inter-contact time) and qualitative (duration of meetings, represented by contact windows). Therefore, Chapter 5 proposes a new optimisation approach that takes into account both contact window and inter-contact times, while Chapter 6 presents a distributed scheme for analysing resources available in the vicinity of a node based on both a qualitative and quantitative analysis of mobility parameters.

2.3.3 Knowledge about past encounters

Relaying information between nodes in partitioned networks, especially over multi-hop or broadcast facilities, leads to inefficiency in data transmission and is an important source of overheads. Besides, as huge delays are usually associated with these transmissions, highly dynamic information such as topology generally becomes outdated upon arrival at destination. Transmitting such dynamic information appears to be useless when it comes to enabling nodes to perform meaningful routing or to make appropriate resource allocation decisions. For this reason, a node may prefer to use information it can obtain during encounters with other nodes and store it as historical information in its own buffers. Moreover, a node that has encountered the destination many times, is likely to encounter the destination again [DJ10]. This is how nodes can learn various parameters during contacts, store them and then derive them in order to make prediction-based decisions, also relying on some heuristics.

In Prophet (Probabilistic ROuting Protocol using History of Encounters and Transitivity) [LDS04], for instance, nodes will gather encounter history information in order to estimate the probability $P_{(a,b)}$ of node a to encounter node b. Nodes will then exchange those probabilities along with scaling and ageing mechanisms, so that at each encounter the node with the best probability to deliver a message to destination becomes (or remains) the custodian. Prophet has been tested within the Sámi Network Connectivity project in Sweden, it is being standardised by the Internet Engineering Task Force [LD07] and it is one of the few tested in a real-life scenario [DUP02]. The Prophet example shows that even when a node ignores the global topology of a network, it can still make autonomous decisions, by relying on estimates acquired during peer meetings. Therefore, the focus in opportunistic networks is on how *estimation-based* techniques rely on collecting information from their vicinity and on exploiting it in order to improve delivery performance.

It is not only encounter history, as in the case of Prophet [LDS04], but also mobility parameters such as velocities or revisiting patterns [LFC06] that can be used in order to make meaningful forwarding decisions. In the same way, node profiles, representing the degree of connection to a specific community, can also be exploited in order to improve message dissemination in a human-based context [NGP07]. Exploiting the social context in DTNs remains a promising line of work, but it falls outside the scope of this thesis. Thus, the distributed mechanisms to be presented in subsequent chapters will only involve technical exchanges between nodes, and will remain generic as far as social aspects are concerned.

Chapter 3 of this thesis will demonstrate how the estimation of contact time between two nodes may lead to improved performance, while Chapter 6 will show how the evaluation of resources available in the vicinity by using only distributed mechanisms and information gathered over past encounters may also contribute to optimising performance in intermittently connected networks.

2.4 Resource-centric routing

This section aims to clarify the relationship between routing protocols, on the one hand, and available network resources, on the other. It lays out the basic elements that will be developed in detail in subsequent chapters. The section first identifies relevant network resources, and then it presents various techniques aimed at resource preservation. It is suggested that node heterogeneity can be used to the benefit of resource-aware routing, making sure that resources in limited supply are not exhausted. The section then presents generic routing decisions together with success factors that can validate the design of a resource-aware routing protocol.

2.4.1 A taxonomy of resources

A critical point to start building a resource-aware routing scheme is to identify the resources relevant for such a scheme and to characterise their cost and availability.

As mentioned before, DTN routing is achieved using the store-carry-forward scheme, therefore a node will need:

1) some *buffer space* in order to be able to store a message,

2) some *energy* to keep node power on and to supply energy to the communication interface, and

3) some *bandwidth* to be able to forward messages towards their destination.

The notion of bandwidth used here should not be considered as equivalent to the notion of nominal bandwidth used in relation to the communication interface. Instead, it should be given a more generic interpretation, as the total quantity of data a node can exchange over a given period of time, considered to be relatively long. During this relatively long period of time a node will be involved in various episodes, with an alternation between connected and disconnected states. Bandwidth is an important resource in fully connected MANETs, and DTNs should also be able to function as MANETs when node connectivity exceeds a particular threshold. Thus, the notion of bandwidth as used in this thesis rigorously remains backward compatible. However, what lies underneath the notion of bandwidth in DTNs is the idea of *connectivity*, which defines this particular interpretation of bandwidth in the context of partitioned networks. In delay- and disruption-tolerant networks, connectivity is determined by two parameters:

1) *meeting quantity*, formally characterised by the distribution of inter-contact times, and

2) *meeting quality*, formally characterised by the distribution of contact times.

In addition, the data volume that can be exchanged between two nodes during given contact periods is also in direct proportion to the nominal bandwidth of the wireless protocol used.

The characterisation of the three resources as presented above partially overlaps with their characterisation in MANETs. However, the demand for storage is significantly higher in DTNs than in MANETs, considering that custodian nodes in partitioned networks store bigger amounts of data and for a longer time than their counterparts in traditional mobile ad-hoc networks. Moreover, a MANET node usually has multiple concurrent connections, therefore it is constrained to allocate the bandwidth available in short supply to various partner nodes. In DTNs multiple concurrent connections can be rare, which implies that bandwidth allocation should focus on: 1) the finite but variable connection time between nodes, and 2) the variable frequency of meeting with peers. In typical DTN scenarios, usually both are in short supply.

Of course, all three resources: storage space, energy, and bandwidth are required *simultaneously* at node level in order to build a successful routing scheme. A node that provides only energy and bandwidth with no available buffer space, or bandwidth and buffer space but no energy is useless from a routing perspective. This idea will be further explored in Chapter 6, and it will be demonstrated that these three resources need to be analysed together, using a holistic approach.

Although all the three resources are needed in order to ensure the success of a routing scheme, their relative criticality to the routing process is not the same. As an example, a node with no energy availabilities (suffering battery depletion) will completely stop routing, potentially losing all messages in its buffer, while a node with no buffer availabilities can still deliver messages available in its buffer to other nodes.

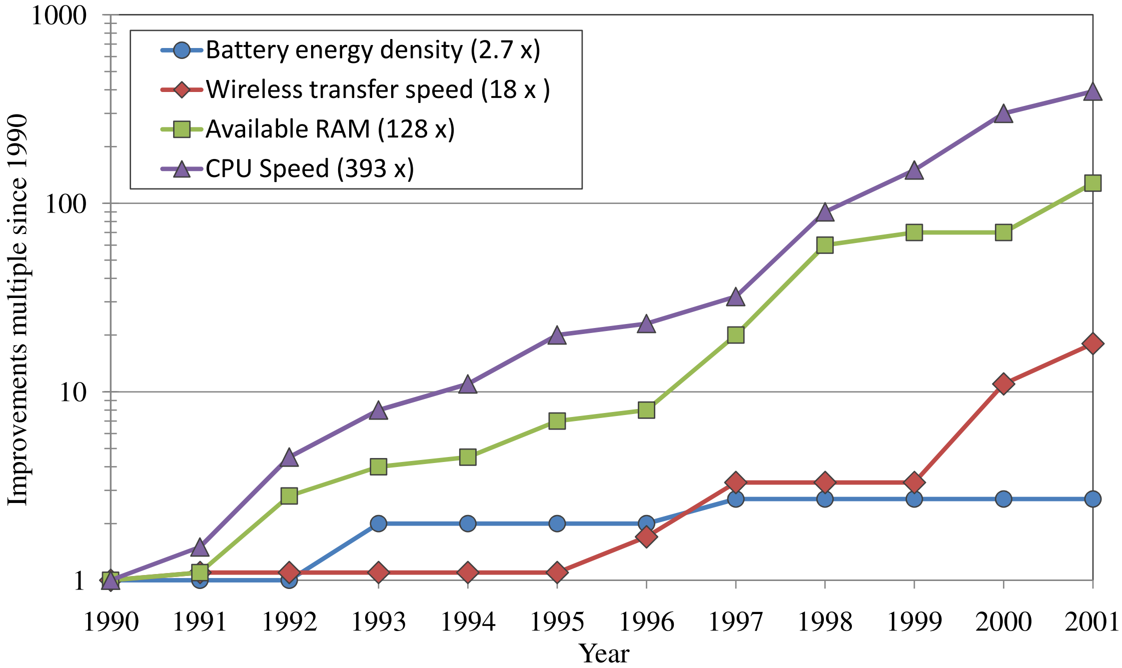


Figure 2.4 Laptop technology improvements during the past decade (cf. [Sta02])

A retrospective look at how resources developed over the previous decade will reveal that resource availability and resource cost evolved at different paces [Sta02]. Figure 2.4 shows that improvements in energy were 7 times slower than advances in bandwidth, while improvements in bandwidth were 7 times slower than advances in buffer space. The figure above tries to suggest why both research and industry have focused on strategies that maximise availability of energy, bandwidth and buffer space by organising priorities in this particular order.

Resource-aware routing schemes cannot afford to miss resources; they really have to make the most of particular resource distributions in order to make successful deliveries. Nodes in DTN networks will be characterised by different levels of storage, energy or communication resources. While routing algorithms make the most efficient use of these resources, they must also make sure that resources in limited supply are not exhausted.

2.4.2 Mobility and bundle size

DTN bundle architecture [CBH$^+$07] suggests combining all application level data and metadata to form a single bundled message, in order to minimise the number of request/response exchanges between two hosts. For example, while e-mail exchange is fundamentally asynchronous, current application protocols for sending (SMTP) and retrieving (POP3, IMAP4) e-mails are fairly verbose, involving numerous message exchanges, and often require user credentials to be provided. So, in a DTN context, it is more appropriate that all e-mail data and metadata (login-name, password, host, port, message body, attachments, request headers, etc.) should be sent together, bundled into one single message, in order to build a non-conversational protocol[3]. From the perspective of a network where a store-carry-forward scheme is applied, it is more reliable to transfer a single big bundle than a sequence of small bundles through the network. When sending series of many small bundles, the unreliability comes from the fact that each small bundle depends recursively on the success of the previous one and each is usually associated with a timeout. Therefore, in the absence of a DTN bundle layer size limit, this will make messages get increasingly bigger in size.

Following extensive laboratory and field measurements carried out on vehicular networks, Ott and Kutscher [OK05] have shown that cars moving at 120 km/h can reach about 1800 m of connectivity when connecting to a stationary WLAN point of access on a highway. They have also shown that the size of data exchanged is between 30 and 70 MB in one pass. Their experiment shows that in the case of mobile transfers there is a maximum size limit for a bundle to be exchanged, depending on relative node speed. In order to transmit a bundle exceeding this size, a node has no other alternative than: 1) to wait for a better contact opportunity (i.e. a node with lower relative speed), or 2) to use proactive fragmentation or erasure coding. The first alternative will be analysed in detail in Chapter 3, while the second alternative will be looked at in Chapter 4 and Chapter 5. Whatever alternative one may choose, it is obvious that a resource-aware routing algorithm cannot dissociate message size from node mobility, so it should take into account - some way or other - the variable and limited contact windows between nodes.

2.4.3 Message differentiation

There are cases when allocating the same amount of network resources to each message may translate into a suboptimal and rigid allocation, especially when traffic demand exceeds network capacity or when available energy, bandwidth or buffer space become scarce. In this case it is then more suitable to fine-tune the mechanism by allocating resources in

[3]Of course this approach also generates security and privacy concerns that should be addressed in a separate analysis.

accordance with the differentiation policy. The utility of a message is introduced as an abstract statement of the benefit of transmitting that particular message instead of others.

Using utility as an entry point for resource allocation has been proposed in fully connected mobile ad-hoc networks [CNT08, XLN06] where charting the route also involves optimising the accrued utility for the whole network. In fully connected MANETs, the calculation of the utility/price factor for each route is usually done by finding a maximum clique in the network graph. In a graph, a complete subgraph is referred to as a clique. A maximal clique is defined as a clique that is not contained in any other cliques. However, this theory is impossible to apply to partitioned networks because of the disconnected network graph.

In the context of DTNs, Balasubramanian et al. [BLV07] use utility in order to optimise resources with respect to delay-related metrics, in particular minimising average delay, minimising missed deadlines, or minimising maximum delay. Spyropoulos et al. [SPR07] use utility to choose the fittest custodian node to carry a message. However there is no network-wide optimisation of the accrued utility in either of the two papers quoted above.

In the cases analysed in this thesis, the use of utility will result in an efficient use of transmission power and an optimisation of bandwidth and storage space. This technique requires a message differentiation that is set up by the user at application level. As shown in Section 2.1.3, the bundle layer provides a class of service similar to the postal system with three priorities of delivery: bulk, normal, or expedited. In Chapters 3 and 4, the concept of utility is used in an attempt to achieve network-wide optimisation, and it is also related to message priority, as defined by the bundle architecture, in order to enforce differentiation. The global optimisation mechanism is, however, in-built in the routing algorithm in a distributed fashion. As discussed in the previous section, the size of a message is important in respect of its adaptation to contact window size. Therefore, Chapters 3 and 4 propose a utilitarian framework taking into account message utility, message size, and potentially also message time-to-live.

2.4.4 Challenging resource heterogeneity

In a heterogeneous environment with different node types, where resources usually range over a wide spectrum, estimating resource availability is a more challenging task than in homogeneous networks. This thesis sets out to investigate mainly city scenarios, where network nodes are mobile: cars, trams, mobile phones, PDAs, each characterised by different mobility models and each having different energy, bandwidth and buffer space availabilities. In addition to this, as in the case of many realistic city scenarios, nodes are expected to freely join and leave the network. First, this phenomenon increases the dynamic character of network topology. Second, it calls for mechanisms through which nodes that decide to leave can find an appropriate custodian for the bundles in their buffers before they actually leave the network. Third, it calls for mechanisms to inform newly entered nodes about the resources available and about network conventions.

A resource-aware routing mechanism should take into account resource heterogeneity by selecting capable nodes that can provide enough resources to deliver messages towards their destination. Chapter 6 shows how topology and mobility knowledge gathered autonomously by one node over past encounters can be combined with knowledge about resources available in its vicinity in order to help that node make meaningful routing decisions. These routing decisions can be grouped into custodian election decisions and queue management decisions.

2.4.5 Custodian election and queue management

In a DTN routing context, in order to increase the probability of successful message delivery, it is crucial for nodes carrying messages to choose the right custodian out of all those available, i.e. the node with the highest probability to carry a message towards its destination. This process is referred to in the literature as *custodian election*. Custodian election is important for all routing protocols, and it becomes increasingly critical as the replication factor decreases, the extreme case being single-copy protocols.

There are many strategies available for detecting which custodian is better suited for receiving a message. For example, some geographical routing protocols, such as LAROD [KNT08], try to detect which node is likely to be closer to the destination, as this translates into a higher probability for that node to actually meet that destination. Conversely, encounter-based protocols, such as Prophet [LDS04], look at the encounter history in order to detect which potential custodian is more suitable for delivering a message. Chapter 6 of this thesis also proposes a new custodian election policy based on the resources found in node vicinity.

Queue management specifies send and drop conditions for messages currently in a node's buffer. As network contact opportunities and related bandwidth are scarce resources, a routing algorithm should define which message to send, and the required order of sending. On the other hand, in order to avoid the filling up of a node's own buffer, thus preventing other legitimate messages from being further admitted into custody, the algorithm should also implement a message dropping policy. Simple techniques implement prioritisation schemes based on criteria such as the urgency header or the remaining message time-to-live. More complex schemes are based on the utility of the message to be delivered, as discussed above.

One correspondent issue that affects queue management policies is detecting messages that have been successfully delivered to destination, so that all remaining copies can be deleted. Since only the first copy of a message delivered is useful, all remaining copies should be deleted in order to avoid wasteful multiple deliveries and to alleviate buffer congestion at custodians. This issue affects messages that have been sent in multiple copies as an effect of replication. However, because these networks are not fully connected, intermediary nodes holding copies of delivered messages are not automatically notified. In such cases, the destination might use the same store-carry-forward mechanism to send back a notification that some particular message has already been delivered, in order for all custodians that carry the delivered message to delete it. This mechanism of informing custodian nodes about completed deliveries is known in the literature as anti-packets or delivery acknowledgements and is described in detail in Chapter 3 of this thesis.

Some of the ideas presented briefly in this section will be detailed in the following chapters. These chapters will describe techniques aiming at minimising consumption of relevant resources in DTNs: energy, bandwidth, and buffer space. While some techniques will focus on message differentiation and delivery utility, others will focus on how reasonably big-sized messages can be transmitted over limited and variable contact windows. Heterogeneity of resource distribution will be used in order to leverage the selection of a custodian, as well as the selection of a message in the queue that is to be transmitted or deleted. Custodian election and queue management are modules that are easy to delimit in any DTN routing scheme, and they contibute to a better understanding and implementation of resource-aware routing schemes.

Chapter 3

The ORWAR protocol

This chapter presents a resource-efficient protocol for **O**pportunistic **R**outing in delay-tolerant networks with **W**indow-aware **A**daptive **R**eplication, or ORWAR for short. The proposed approach exploits the context of mobile nodes (speed, direction of movement, and radio range) to estimate the size of a contact window. This knowledge is used to make better forwarding and replication decisions and to minimise the probability of partially transmitted messages. While optimising the use of bandwidth during overloads, this also contributes to reducing energy consumption, since partially transmitted messages are useless and only waste transmission power. Another feature of the algorithm is the use of a differentiation mechanism based on message utility, whereby more resources can be allocated for high utility messages. More precisely, messages are replicated in the order of highest utility first, and removed from the buffers in reverse order. To illustrate the benefit of such a scheme, global accumulated utility is used as a system-wide performance metric. Simulations illustrate the benefit of this model and show that ORWAR provides lower overhead and higher delivery ratio, as well as higher accumulated utility as compared to a number of well-known algorithms.

Section 3.1 makes an overview of the context in which DTN routing protocols operate, while Section 3.2 is devoted to related works in delay- and disruption-tolerant networks and includes a short presentation of several comparable algorithms. Section 3.3 includes the main objectives of the algorithm, while Section 3.4 contains a complete presentation thereof. Then, a comparative evaluation of the protocols is presented in Section 3.5 and the chapter ends with a summary in Section 3.6.

3.1 Overview

As described in Chapter 2, the architecture of delay-tolerant networks defines an abstraction layer on top of the transport layer and below the application layer, called bundle layer. As no assumption can be made about underlying networks, this overlay architecture is responsible for routing data from source to destination. DTN architecture neither defines any fixed-length data units nor puts any upper or lower bounds on application data unit size. The bundle layer is responsible for the end-to-end message delivery mechanism, called virtual message forwarding [CBH+07, War03].

Popular application layer protocols such as FTP, HTTP, Telnet, IMAP, POP are historically designed for networks relying on TCP as the transport layer. These application layer protocols are in general interactive, very chatty, thus using a lot of bi-directional communication. As an example, Figure 3.1 a) presents the simplified sequence diagram for the traditional FTP protocol between two machines, usually called client and server. This diagram clearly shows the large volume of bi-directional communication this protocol needs to use even for performing such a simple task as downloading a single file.

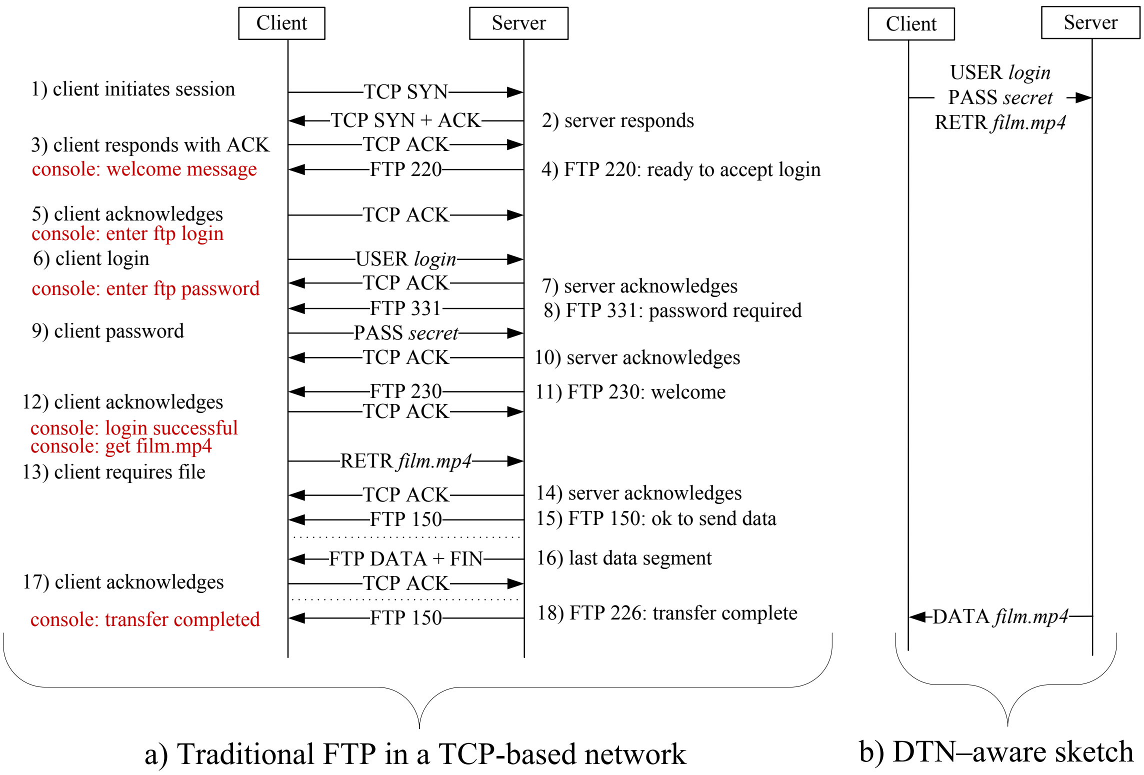


Figure 3.1 Sequence diagram of a file transfer protocol: FTP in a TCP-based network compared to a DTN-aware sketch

However, sending such a sequence of interconnected messages would be largely infeasible from a DTN perspective, because nodes cannot rely on quick data acknowledgements. This observation points to the need of adapting the concepts and semantics of TCP-based applications to a disconnected environment. Therefore, DTN-aware applications should minimise the number of end-to-end transactions, by packaging as much data into one bundle as possible. A potential DTN-aware file transfer protocol, as outlined in Figure 3.1 b) would require that the client aggregates all the data (ftp user name, ftp password, a list of files to retrieve, etc.) into one single bundle that is then sent to the server. Similarly, the response of the server may contain host and account identification data, as well as the requested files, all aggregated together into one single response bundle. The problem is, however, that this process will make bundles grow in size, and eventually relatively large bundles will have to pass over limited and variable contact windows. To tackle that problem, this chapter proposes some mechanisms aimed at corellating the amount of data transferred over a contact window with the duration of that contact.

DTN architecture actually offers three relative priority classes which differentiate traffic based on an application's expression of urgency at the message source. These have some impact on solving traffic contention as well as resource allocation issues. For example, in the current reference implementation [DTN07], when storage at one node becomes short, expiration of bundles will start with the low priority class. While this is a suitable mechanism for differentiation at user level, it does not take account of message size and thereby does not provide an optimised use of resources at system level. This chapter demonstrates how *per bit utility* can be combined with the traditional idea of priorities to achieve better use of resources at system level.

In particular, considering that networking activities account for 10-50% of the energy spent by a mobile device [KK98] and that the gap between battery capacity and mobile device energy requirements is increasing, designing energy-efficient network architectures is vitally important. Starting out from these two basic factors, large bundle sizes in DTN and a constant need for energy-efficient schemes, this Chapter proposes ORWAR as a new DTN protocol.

3.2 Related work

In their paper about routing in delay- and disruption-tolerant networks, Jones et al. [JLSW07] describe two main strategies for achieving higher delivery ratios: a) acquiring *knowledge* about the network, and b) *replication*. In an opportunistic scenario, such as that assumed in this thesis, a node can make decisions based exclusively on information it can acquire from its vicinity.

In order to increase the chances of successful delivery in a partitioned network, some routing algorithms replicate copies of each message to several custodians. However, this also consumes resources (bandwidth and implicitly energy, as well as storage space at custodians) in proportion to the number of copies forwarded. Some of the better-known examples are the Epidemic protocol [VB00] and the SprayAndWait protocol [SPR05]. The latter algorithm introduced the idea of *controlled replication* in an attempt to limit the extent of replication. ORWAR, the protocol proposed in this thesis, takes controlled replication a step further, by introducing the notion of *utility-aware replication*. The underlying concept of utility is considered here as an abstract metric used to compare the usefulness of delivering each message.

The notions of utility and message differentiation have also been used in fully connected mobile ad-hoc networks [CNT08] where construction of the route also requires maximising the accumulated utility for the whole network. However, this maximisation is much more difficult to achieve in a partially connected network, because of the limitations that unconnected nodes may impose on a potential bidding algorithm. In the context of DTNs, Balasubramanian et al. [BLV07] use utility for achieving optimisation with respect to delay-related metrics, in particular minimising average delay, missed deadlines, or maximum delay. Spyropoulos et al. also use utility in [SPR07] to choose the fittest custodian nodes that can carry messages. However, in the context of DTN, neither of the papers provides for a network-wide optimisation of the accrued utility.

The performance of the proposed protocol will be evaluated later in Section 3.5, where ORWAR is compared to five competing algorithms. ORWAR, similarly to the other five baselines, performs opportunistic unicast routing in delay-tolerant networks. The algorithms to be considered in this comparative analysis have been chosen from a list of algorithms known to achieve high performance levels, that also fulfill the additional condition of fitting a sparse network in a city scenario, such as that considered for ORWAR, with nodes carried by cars, buses, or pedestrians. On the other hand, the list excludes specialised protocols used in satellite or interplanetary communications, as the latter rely on a schedule-based rather than an opportunistic scenario. Below is a brief description of these competing protocols:

- *MaxProp* [BGJL06] is a routing protocol originally developed at the University of Massachusetts for vehicle-based disruption-tolerant networks. It uses a combination of mechanisms such as greedy replication, advanced queue management, and message acknowledgements in order to achieve higher delivery rates. Greedy replication means that all contact opportunities are used to send a message copy. However, Maxprop performs a strict selection and ordering of the messages that should be sent and those that should be deleted via advanced queue management policies. In other words, the message queue is kept in good order based on the likelihood of the messages in that queue to meet their destination. As a result, nodes that are seen infrequently tend to score lower values over time. In order to calculate those values, let every node i hold a vector f^i representing its likelihood to meet every other node. The likelihood of node i meeting node j is initially set to $f^i_j = 1/(n-1)$, n being the total number of nodes in the network. At each meeting with j, f^i_j is increased by 1, then the vector of likelihood f^i is renormalised using incremental averaging, such that $\sum_{j=1}^{n-1} f^i_j = 1$. Messages are sent and deleted from the queue based on path calculation using subsequent likelihoods. Finally, the acknowledgments for the messages delivered that propagate through the network will allow custodians to eliminate redundant copies of the messages that have already arrived at destination.

- *SprayAndWait* [SPR05] is the emblematic routing protocol using controlled replication, and it was developed at the University of Southern California. The idea behind this protocol is to mitigate the overhead problem of epidemic schemes, proposing a *replication factor* L that determines the maximum number of copies that can be found in the network at one time. As mentioned earlier, the original SprayAndWait scheme includes 2 phases. In the first phase, also called the *spray phase*, copies are forwarded to a number of nodes encountered, according to the replication factor *L*. In the second phase, also called the *wait phase*, nodes carry the copies until direct encounter with the destination in order to deliver the message. Spyropoulos et al. also show in their work [SPR05] that the number of copies necessary to produce a certain delivery latency is independent of network size. There are two versions of the protocol depending on how L decreases at every encounter in the spray phase: L decreases by 1 in the *normal version* of the protocol, but it is halved at each encounter for the *binary version* of the protocol. The binary version has been demonstrated to achieve better latencies, therefore it is considered to be the optimum spray strategy. Consequently, only the optimal binary scheme has been considered in the simulations included in this thesis.

- *PROPHET* [LDS04] or **P**robabilistic **Ro**uting **P**rotocol using **H**istory of **E**ncounters and **T**ransitivity is an opportunistic protocol using predictability of encounters in order to achieve higher delivery rates. It was developed at the Luleå University of Technology and it is also one of the few protocols implemented in a real-world scenario (providing opportunistic communication and Internet access for reindeer herders in Sweden). The protocol maintains at each node a probability $P(a,b)$ of node a to encounter node b. $P(a,b)$ is updated at each encounter $P(a,b)_{new} = P(a,b)_{old} + (1 - P(a,b)) \times I_c)$ where I_c is an initialisation constant. All predictabilities are also aged such that $P_{new} = P_{old} \times \gamma^k$ where γ is an ageing factor and k is the time elapsed since the last ageing. Finally, predictabilities are exchanged between nodes using transitivities such that $P(a,b)_{new} = P(a,b)_{old} + (1 - P(a,b)) \times P(a,c) \times P(c,b) \times \beta$ where β is a scaling constant. The delivery predictabilities are recalculated at each opportunistic encounter according to the rules presented above.

- *Epidemic* represents the emblematic routing protocol using greedy replication. At each encounter it transmits a copy of a message if the node met does not yet hold a copy of that message. Despite being a simple protocol, it is known to have, at least in resource unconstrained environments, one of the shortest delivery latency. It is also known to be an unscalable protocol, meaning that performance suffers when the number of network nodes increases.

- *Direct delivery* uses only node mobility in order to deliver messages. In fact, a message will travel along with the source node up to the moment where it meets the destination node. As a consequence, this protocol uses minimal overhead and consumes very few network resources. On the other hand, latency is very high.

Most of the protocols proposed in DTN conspicuously ignore the size of messages to be forwarded to a custodian. However, in a realistic DTN scenario, the size of messages to be sent usually varies over a wide range. Moreover, as argued before, contact windows are limited, which allows only limited amounts of data to be transmitted. The Rapid protocol [BLV07] is one of the few protocols that records the size of previous transfer opportunities and uses a heuristic approach to select a message that runs good chances of passing through, given the history of encounters. However, it fails to propose an estimation for the contact window or transfer opportunity taking into account actual meeting data.

3.3 The rationale behind the protocol design

ORWAR uses local connectivity knowledge in order to route messages from source to destination. Connectivity knowledge is unavailable in advance, but is gathered from the vicinity on a peer-to-peer basis during contact. Neither message arrival rates nor meeting schedules are known specifically, therefore routing is completely opportunistic. While message sizes are considered relatively large as compared to contact windows, custodian buffers are assumed to be finite. Nodes cannot rely on global knowledge, but an individual node will know its own speed, direction, and in most cases also own location using, for example, in-built geolocation support. In fact, the algorithm only needs to estimate the relative distance to the neighbours found within radio range. Relative node location can be provided also by low-power short-range radar systems[1] that are increasigly used in the automotive industry [Aus01].

This protocol has been designed with two objectives in mind. The first is to optimise system level resources, in particular energy and bandwidth. Message priority levels are a simple means of achieving differentiation when resources are scarce. However, when message sizes vary considerably, a more fine-grained differentiation mechanism is required. Thus, utility/bit is proposed as an abstract declaration of the benefit of one transmission as compared to others. Assuming that every message comes with a given utility value, accumulated utility can be used as a system-level evaluation metric. The unit for measuring utility is irrelevant, since it only reflects a global measure of benefit.

The second objective is to achieve a high delivery ratio in a partitioned network. This can be done using store-carry-forward and replication mechanisms in DTNs. In order to make replication energy-efficient, forwarding decisions should be based on both utility/bit for the messages involved, and on local connectivity characteristics.

To perform routing under intermittent connectivity, ORWAR proposes a multi-copy routing scheme, using controlled replication, that is the distribution of a fixed number of copies over the network. At each contact, a node tries to forward half of the message copies in its buffer to a custodian, keeping the rest for itself. Up to this point, the scheme is similar to the SprayAndWait mechanism presented by Spyropoulos et al. in [SPR05]. However, enhancements are made in 4 directions:

- Messages with the best utility per bit ratio are first selected, and they are sent only if their size matches the properties of the contact, which reduces the volume of partially transmitted messages.
- The replication factor is determined as a function of message utility, which increases delivery probability and diminishes latency for bundles with highest utility.
- The purging of messages from a buffer starts with the least utility per bit message.
- Bundles known to be delivered are removed.

This thesis uses the concept of utility for achieving network-wide optimisation, while also relating it to message priority, in order to allow better message differentiation. A simple illustration of the way in which the utility criterion applies is an exclusively priority-based scheduling policy, in which higher utilities are always preferred to lower ones.

The thesis also assumes that utility is not a system-internal parameter used to enforce internal policies at system level, but it is rather directly specified for each message. It

[1]The range of a typical automotive radar system is 150-300 meters and associated radio equipment usually works in the 76-77 GHz bands.

reflects the satisfaction (or benefits) the user assigns to the results of running an application, thus being an important element for constructing a quality of service (QoS) standard.

3.4 Presentation of the algorithm

3.4.1 Algorithm data structures

Every node i keeps the following data structures: 1) the message queue (mq_i) that includes information about utility (u_k) and size (s_k) for each message m_k, kept in utility/bit order, and 2) a record of known delivered messages (kdm_i).

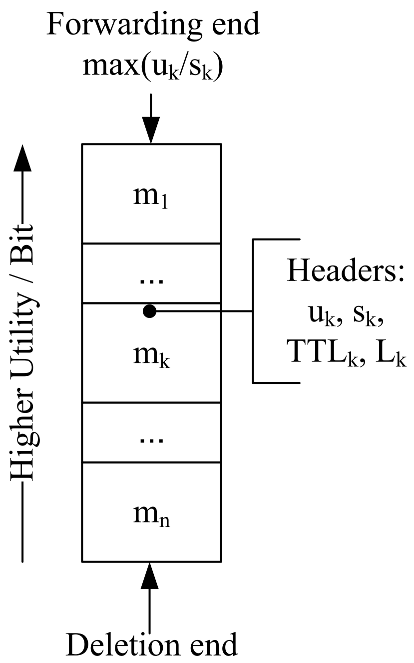


Figure 3.2 ORWAR message queue

Figure 3.2 shows the structure of a node message queue. New messages from the application layer, as well as messages from neighbouring peers are inserted in the correct position with respect to the u_k/s_k ordering. Messages are deleted from the lower end of the queue. This may occur when a new message with higher utility per bit rate is to be inserted and the queue is full. In relation to the notion of priority in DTNs, and in order to improve message differentiation, three utility per bit values have been defined in this model. The approach is however general and can be extended to multiple levels of utility. In this thesis, utility per message is considered to be time-invariant. Every message header also includes L_k which denotes the intended number of message copies.

Finally, TTL_k is an application-based parameter that indicates message time-to-live. This can be implemented as an absolute value where all nodes can be considered to have access to synchronised (for instance GPS-based) clocks, or as an interval to be recalculated at each node where a message arrives, using the local clock of that node.

The parameter kdm_i is used to keep track of delivered messages using a hash table where the keys are message IDs. These records are exchanged at each meeting and all messages known to have been delivered are subsequently deleted from the message queue. This technique is also known as the *vaccine* mechanism. The size of kdm_i will be kept to a minimum using the message time-to-live (TTL) parameter, as described in the algorithm presented below.

3.4.2 Contact windows

A contact window or link duration is the time during which the connection between two mobile nodes is possible, and it is dependent on a node's wireless and mobility properties. A simple approach has been chosen for modelling the wireless subsystem, based on the common assumptions made in simulations of wireless networks, as presented by Kotz et al. [KNG+04]. To begin with, nodes are assumed to be in 2D space. Disregarding atmospheric conditions, node placement and orientation, as well as battery status, every node i is considered to have a circular radio range with radius r_i. Even if real world radio ranges are neither regular, nor circular or contiguous, they are considered to be perfect circles in this analytical model. Transmission is considered to be perfectly symmetrical. Obstacles that cause signal obstruction, diffusion or scattering are specifically ignored. This model is built for a sparse network, as it is often the case in Delay- and Disruption-Tolerant Networks. Therefore the model is limited to contacts between node pairs only.

A contact window t_{cw} is calculated from the nodes' respective speeds ($\vec{v}_1$, $\vec{v}_2$), the nodes' coordinates and transmit range (r_1, r_2), as shown in Figure 3.3, in which dashed trajectories denote the movement of a node. Given two nodes advancing at vectorial speeds of $\overrightarrow{v_1}$ and $\overrightarrow{v_2}$ respectively, having the respective radio ranges r_1 and r_2, the contact window time t_{cw} can be calculated as being:

$$t_{cw} = \frac{2 \times \min(r_1, r_2) \times \cos\alpha}{|\overrightarrow{v}|} \tag{3.1}$$

where $\overrightarrow{v} = \overrightarrow{v_1} - \overrightarrow{v_2}$ and α is the angle between $\overrightarrow{v}$ and the line defined by the two nodes at contact time, as depicted in Figure 3.3. Nodes will be in contact as long as the distance between them will not exceed the minimum radio range, therefore maintaining a distance between nodes smaller than $\min(r_1, r_2)$ is the condition for maintaining a connection. Moreover, in equation (3.4), α and $\overrightarrow{v}$ are affected by the mobility model, therefore

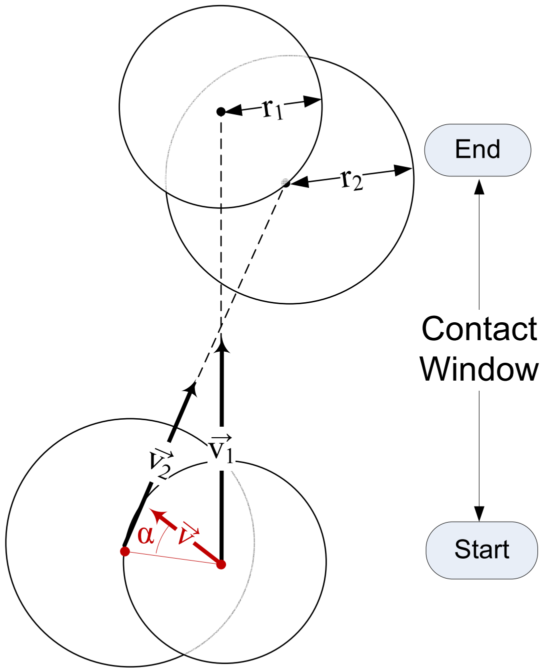


Figure 3.3 Contact window calculation

they can only be evaluated under a specific scenario. For example, in the case of a city scenario where nodes are carried by pedestrians, cars and trams, most of the contacts are established between nodes having rectilinear and parallel trajectories, as defined by roads.

Before forwarding/delivering a message, ORWAR computes the size of the largest transmittable message (s_{max}) based on the current connectivity context, by first estimating contact window time (t_{cw}) and data rate (b):

$$s_{max} = b \times t_{cw} \tag{3.2}$$

This largest transmittable message will be used to relay only messages that have a lower risk of transmission failure, thereby saving transmission power and bandwidth. Both energy and bandwidth are further optimised by the selection of messages that fit into s_{max} in the order of utility/bit. Data rate (b) is given by the radio properties of the device (e.g. for Bluetooth 2.0 data rates are about 250kBps).

Of course, mobility implies that nodes can change speed or movement path during a given transmission. If the actual contact window is different from the calculated contact window t_{cw}, the transmission of some selected message may fail. Although such cases cannot be avoided completely, calculating the fittest message to relay is by far a better solution than randomly taking any. Moreover, in some scenarios, e.g. in a city where nodes (cars, pedestrians) generally have rectilinear trajectories (given by streets) velocity is expected to be mostly constant for the short interval of a contact.

By preventing a node from transmitting a message that has no chance of reaching its destination, ORWAR achieves two objectives: 1) limiting overhead in terms of bandwidth, and 2) conserving power, as the radio signal is not wasted for messages that cannot be sent anyway.

3.4.3 Description of the algorithm

Before moving on to a more detailed description of the algorithm, let us recall the assumptions on which it rests:

- nodes can estimate distances to other nodes in the vicinity. This vicinity is comparable in size with the radio range, and can be estimated for instance by geolocation, or short-range radars. However, nodes lack knowledge about the location of the destination node or of potential custodians outside radio range;
- a disk model is used for radio range, and constant bandwidth is assumed during the entire contact;
- energy resources are limited and transmission power is a significant factor in the function that describes battery discharge;
- there is a lack of knowledge about meeting schedules and durations.

The pseudo-code for the algorithm is described in Figure 3.4. The algorithm includes three main parts: (1) generating a packet from an application and inserting it into the message queue, (2) exchanging metadata and estimating the contact window and, (3) transmitting the bundles while observing a particular priority ordering.

The first part involves the insertion of a new message into the message queue mq of the node in the order of *utility/bit*. The second part consists in exchanging data with a partner node, such that the *vaccine* mechanism is put in place and the contact window can be evaluated. At each encounter, the node updates its kdm list using the knowledge of its neighbours in their respective kdm. Known delivered messages that appear in a node's mq are then deleted. Furthermore, messages that are older than their stipulated TTL are removed from the queue in order to prevent them from spreading across the network. After that, the longest contact window is computed for each pair of nodes that meet. This contact window is used first to directly deliver messsages intended for a neighbour, and then for forwarding replicas of the messages held in the node queue.

Finally, the third part is represented by the bundle exchange, and it includes several phases, corresponding to final delivery transmission, replication transmission or message

```
For each node i:
        \vec{v_i}     //  node speed
        r_i           //  node radio range
        kdm_i         //  delivered messages
        mq_i          //  current message queue
        sb_i          //  available storage buffer
For each message m_k:
        m_k.u         //  message utility
        m_k.s         //  message size
        m_k.ttl       //  message time-to-live
        m_k.ack       //  message transmission ACK
        m_k.dest      //  destination node of m_k

on m_k initiation do // original messages from applications
    insert m_k in mq_i // such as mq_i remains ordered based on u_k/s_k
foreach meeting between i and j do
    send kdm_i to j                  // inform about delivered messages
    receive kdm_j from i             // receive information on delivered messages
    kdm_i = kdm_i ∪ kdm_j            // merge known delivered messages
    foreach m_k ∈ mq_i do
        remove m_k if m_k.ttl has expired // delete expired messages
        remove m_k if m_k ∈ kdm_i     // delete messages known delivered
    send r_i, \vec{v_i}, receive r_j, \vec{v_j}   // exchange meeting geometry information
    s_max = (2 × min(r_i, r_j) × cos α) / |\vec{v_i} − \vec{v_j}|   // calculate contact window;
    while s_max > 0 do
        foreach (m_k ∈ mq_i) ∧ (m_k.dest = j) ∧ (s_k < s_max) do
            // sent to final delivery
            s_max = s_max − s_k
            deliver m_k to j
            if m_k.ack then
                insert m_k in kdm_i
                remove m_k from mq_i
        foreach (m_k ∈ mq_i) ∧ (m_k ∉ mq_j) ∧ (s_k < s_max) ∧ (L_k > 1) do
            // replicate to custodian
            s_max = s_max − s_k
            replicate m_k with L_k/2 to j
            L_k = L_k/2
        foreach m_k received from j do
            // receiving message
            if sb_i < m_k.s then
                if u_k/s_k > last(mq_i).u / last(mq_i).s then
                    mark last(mq_i) for deletion // make some space if buffer full
                if sb_i < m_k.s then return // may not be enough
            s_max = s_max − s_k
            send m_k.ack to node j
            if m_k.dest = i then
                insert m_k in kdm_i // message delivered
            else
                insert m_k in mq_i // such as mq_i remains ordered based on u_k/s_k
```

Figure 3.4 ORWAR pseudo-code

reception. Final delivery transmission has priority over replication transmission to a custodian. Then, messages are transmitted to a neighbour, while the diminishing contact window is still accommodating new messages. It is important to mention that replication is performed in the order of utility/bit for all fitting messages, therefore this technique has been called utility-aware replication. Similarly to binary SprayAndWait [SPR05], L_k is divided by 2 at each replication. The value of L_k is utility driven, chosen according to Table 3.1, where L and Δ are algorithm parameters.

Table 3.1 Initial message copies as a utility function

Priority Class	Utility	L_k=# message copies
High	3	L + Δ
Medium	2	L
Low	1	L - Δ

3.5 Performance evaluation

This section proposes an evaluation of ORWAR's performance in comparison with the five well-known delay-tolerant network routing protocols presented above: SprayAndWait [SPR05], Prophet [LDS04], MaxProp [BGJL06], Epidemic [VB00], and DirectDelivery. ONE (Opportunistic Network Environment) [KOK09], a dedicated simulator for delay-tolerant, opportunitic networks was used for this comparative evaluation. It is interesting to note that ONE also allows simulations to be visualised interactively in real time and it is also a powerful tool for generating mobility traces. DTN simulations were run for all these six routing protocols and the results of those simulations are presented below.

3.5.1 Simulation setup

As both SprayAndWait and ORWAR use a fixed number of replicas, both are run in the evaluation with the same replication factor (L=6). Since messages are evenly distributed across the three utility classes considered in the experiment, the total (maximum) number of copies in the system does not differ for the two protocols under consideration, thus yielding comparable results. What ORWAR does is simply to apply a higher replication factor (L + Δ) for high utility messages, and a lower factor (L - Δ) for low utility messages. The ideal Δ is defined experimentally as being about L/3, thus in this evaluation $\Delta = 2$. On the other hand, Prophet [LDS04] is run with the following parameters: delivery predictability $I_c = 0.75$, the scaling constant $\beta = 0.25$ and ageing constant $\gamma = 0.98$. ONE version 1.3 comes with the following protocol implementations: SprayAndWait, Prophet, MaxProp, Epidemic, and DirectDelivery. The evaluation has been run using these shipped protocol versions.

The evaluations below are based on a city setup with 126 nodes (80 pedestrians, 40 cars, 6 trams) sharing a 4500 m x 3500 m playground. Every point plotted in the figures of this section is the result of 10 measurements for which different initial node positions and initial directions of movement have been chosen. The confidence interval is relatively limited, i. e. 1-4% of the average value. Each node is assumed to have a network interface allowing a transmission range of 10 m for pedestrians and 20 m for cars and trams. For both cars and trams, a transmission speed of 250 kBps (2 Mbps) is considered. Buffers are considered to be 5 MB, with the exception of trams, whose buffers can hold 50 MB. The mobility pattern is close to reality, and pedestrians, cars and trams follow a map-based movement. Cars drive only on roads, while trams run only on their well-defined tracks (the Helsinki map and the original setup have been preserved in order to ensure comparability of results). Speeds are set in the interval [10, 50] km/h for cars, and [1.8, 5.4] km/h for pedestrians, and pauses are random.

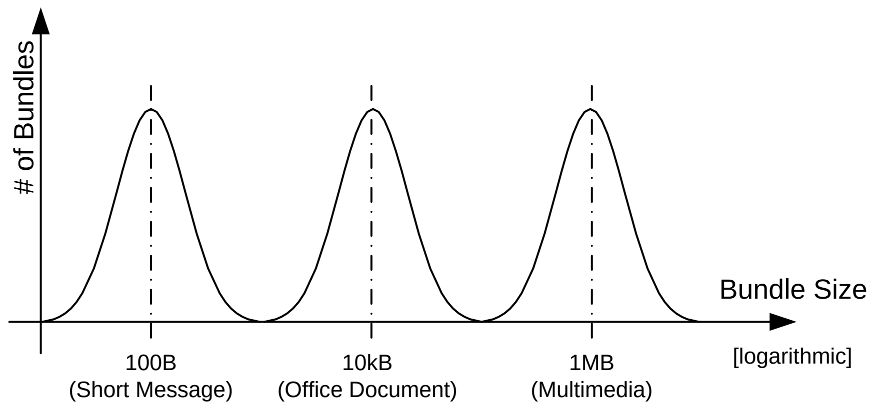


Figure 3.5 Standard message set as a distribution of message sizes (S)

The network is very sparse, with the accumulated transmission area (the sum of radio coverage) for all nodes being 0.25% of the playground, and total meeting time (the sum of all meeting times) accounting for about 3% of elapsed time. Each simulation runs for 12 hours and TTL is considered to be infinite for all messages. The mix of bundle sizes includes the following items:

- 1000 short messages averaged at 100B,

- 1000 documents averaged at 10kB,
- 1000 multimedia files averaged at 1MB.

Size distribution is shown in Figure 3.5. Every message comes with a constant utility which is evenly distributed over size classes, i.e. every size class (short messages, documents, and multimedia) includes an equal number of bundles of utility 1, 2, and 3. In what follows, this size distribution shall be referred to as S.

3.5.2 Message size implications

The aim of the first experiment is to analyse the impact of message size on performance. The starting position uses the standard message size distribution S, then message size is gradually reduced, while the number of messages is increased. That is, first 3000 messages are injected at initial size, then 6000 messages are injected where message size is halved, and finally 30000 messages are used with the initial size divided by 10. Thus, at every simulation the same total amount of data is injected into the system.

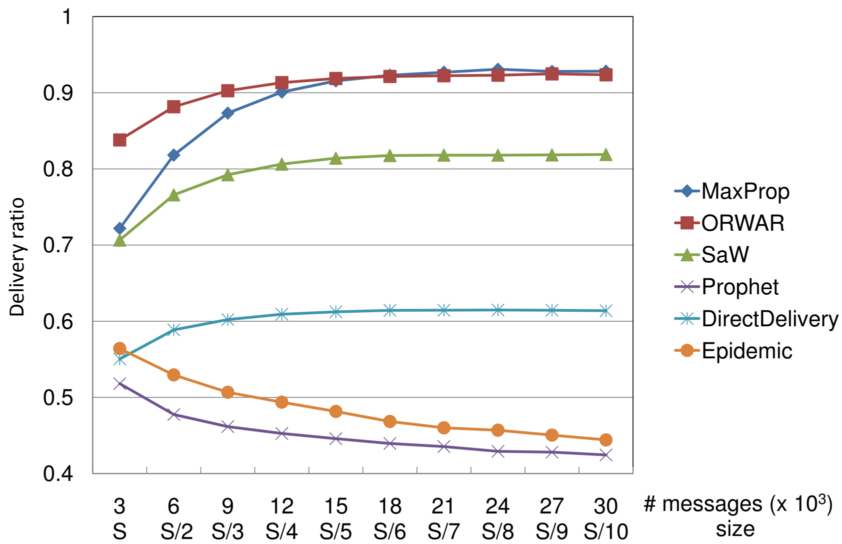


Figure 3.6 ORWAR: Delivery ratio versus message size

Figure 3.6 shows that ORWAR has the best delivery ratio and performs better when bigger messages are injected into the system. When analysing overheads, defined here as the number of bundles transmitted divided by the number of messages delivered to destination, Figure **??** shows that ORWAR has the lowest overhead with the exception of DirectDelivery. Moreover, it compares favourably with SprayAndWait by a margin of roughly 10%, which can be explained by the fact that ORWAR diminishes partial transmissions.

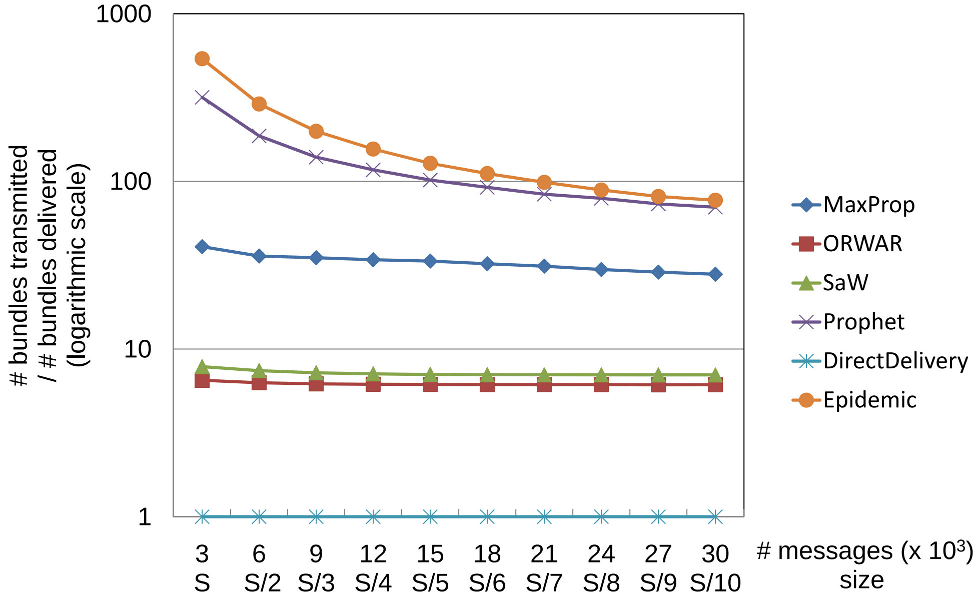


Figure 3.7 ORWAR: Overhead versus message size

From the two figures above, it may be concluded that ORWAR has the best overall performance among the comparable algorithms included in this experiment. Moreover, it appears as an effective alternative for cases where larger messages are to be transmitted and fragmentation is not available/desirable. Considering the results obtained by comparing ORWAR with the other five protocols, as presented in Figures 3.6 and 3.7, the subsequent sections of the paper will concentrate only on the top three performers: MaxProp, ORWAR and SprayAndWait.

3.5.3 Energy implications

The most important goal in designing ORWAR has been to reduce partial transmissions in order to save energy. ORWAR has been shown to have 10% less overhead than SprayAndWait, and much less overhead as compared to the other schemes. Although the notion of protocol overhead, defined as the number of messages transmitted divided by the number of messages delivered, is widely used in the literature, simply counting the number of messages potentially disadvantages protocols that use (small-sized) acknowledgements, such as MaxProp, over protocols that do not, such as SprayAndWait. Rather than focusing on a number of bundles that might have very different sizes, these experiments have concentrated on the total amount of data transmitted, aborted or dropped. Another element to be considered is that overheads may be related to different mechanisms: connection abortions (i.e. neighbour out of reach while sending message, wireless contention), messages sent but dropped (i.e. buffer shortage at custodians), or overheads inherent to the replication factor (number of copies in the system). By estimating contact window and estimating s_{max}, ORWAR tries to diminish aborted transmissions. Therefore, this can be taken as an appropriate metric for measuring *waste cost*.

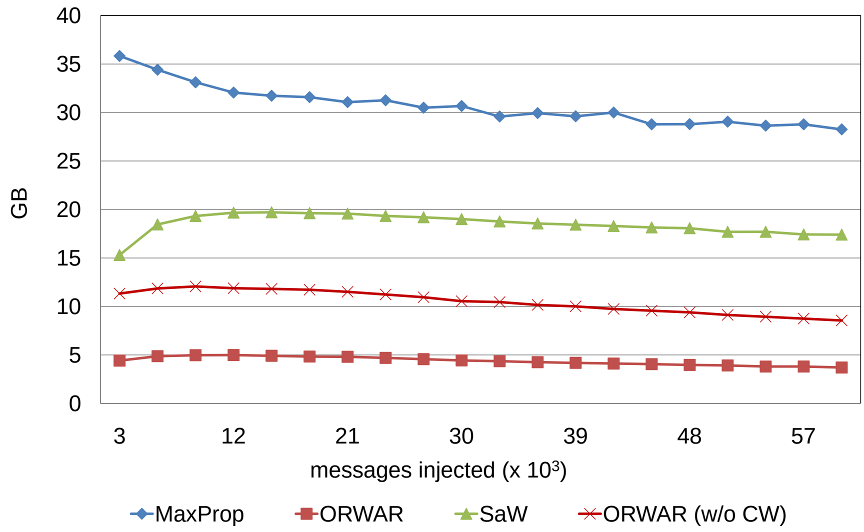


Figure 3.8 ORWAR: Partial transmission total size versus load

In another experiment, network load is increased by gradually increasing the number of messages, while preserving the same average message size - as defined by S. The simulation starts with 3000 messages injected within 12 hours, and then the number is increased to 6000, 9000, 12000, and eventually 60000 messages, in order to demonstrate the effects of growing loads. Figure 3.8 depicts the accumulated size of aborted transmissions on the y axis.

In addition to ORWAR, MaxProp and SprayAndWait, a new (fourth) curve is plotted in Figure 3.8: it is ORWAR, where the module responsible for estimating contact window is eliminated. This allows a direct measurement of the added value of contact window estimation, and distinguishes it from other ORWAR mechanisms, such as queue management or utility-based replication. Measurements reveal an improvement by a 4 to 6 factor as compared to both MaxProp and SprayAndWait. The remaining aborted transmissions in ORWAR can be explained by nodes changing trajectories or speed during message transmission, or by wireless contention. Obviously these cases cannot be avoided, and the computation of contact windows is shown to yield a 50% reduction of aborted transmissions as compared to the situation where they are not computed at all.

As ORWAR computes the most valuable message to be sent in a given meeting context, it will not always send small messages at the cost of dropping bigger ones, as expected.

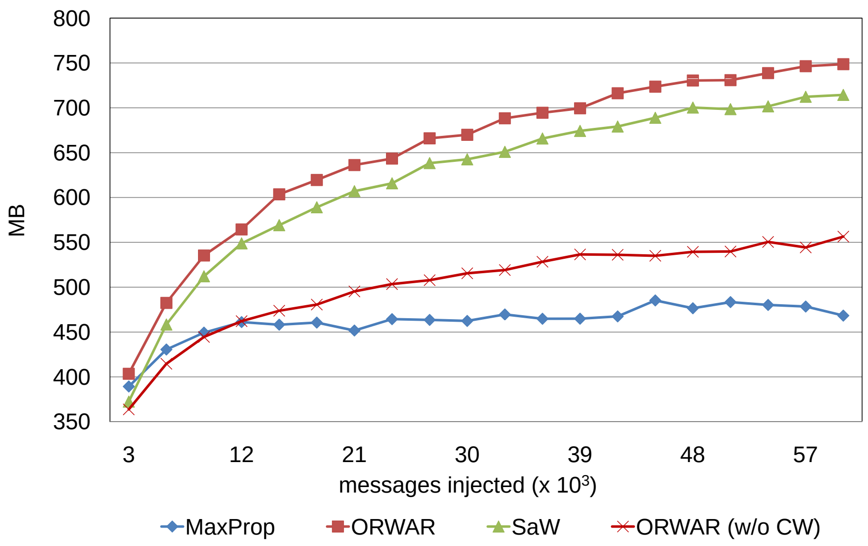


Figure 3.9 ORWAR: Total data delivered versus load

Obviously, it would be unacceptable to make energy savings at the cost of delivering less data. To verify this, Figure 3.9 plots the total data delivered during 12 hours.

The figure shows that ORWAR sends 10-90% more data than SprayAndWait and MaxProp over the same time interval (12h). It also shows that the total volume of data delivered increases at the same rate when contact window estimation is used. In conclusion, this section has shown that the accumulated volume of aborted messages is much more favourable for ORWAR than for competing protocols and, more importantly, the limitation of the biggest message to be sent within s_{max} gives a 50% reduction over applying no limitation at all. By estimating the contact window and selecting the "fittest" message to be sent, ORWAR will not only diminish partial transmissions, but will also increase total data delivered.

3.5.4 Load implications

Figure 3.10 shows the values measured for message delivery ratio during the gradual increase of the load in the network. The number of messages injected in the system is 3000, 6000, and so on, up to 60000 messages over 12h, while keeping the message distribution S.

ORWAR not only has the best overall delivery ratio, but its relative performance as compared to other protocols also increases at higher loads. The explanation is that ORWAR maintains a low overhead which pays off when the network is congested. Other mechanisms, such as effective queue management, utility-based replication, the vaccine mechanism, and contact window estimation also contribute to achieving improved perfor-

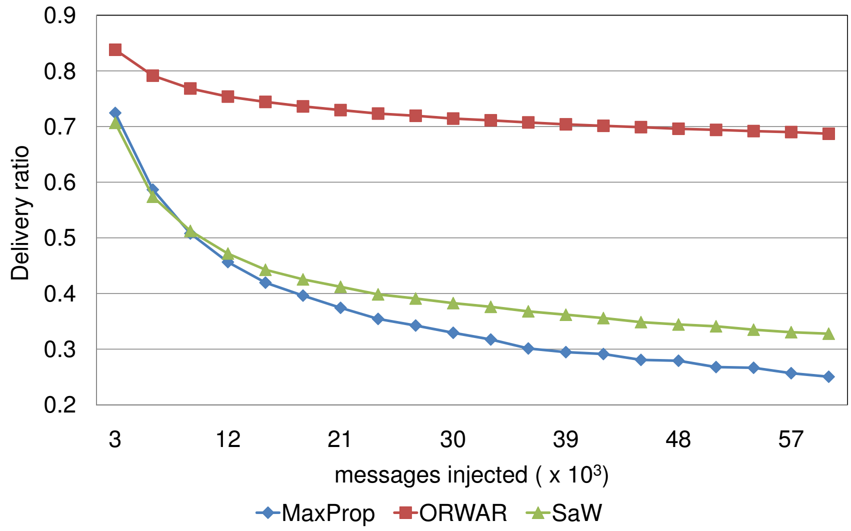


Figure 3.10 ORWAR: Delivery ratio versus load

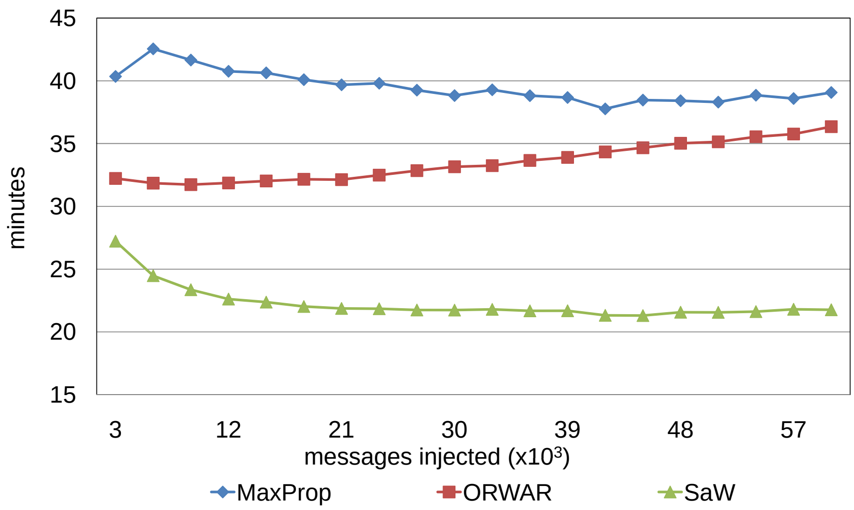


Figure 3.11 ORWAR: Mean latency versus load

mance. As far as latency is concerned, as shown in Figure 3.11, ORWAR performs second best after SprayAndWait. This is reasonable, as messages will stay longer in the buffers while waiting for suitable contact windows.

A major benefit of ORWAR is demonstrated in Figure 3.12, which plots accumulated utility.

It is important to note that utility is accounted for only if a bundle reaches destination. Because messages are treated differently according to their utility, i.e. more resources are available for high utility messages, ORWAR obtains higher accumulated utility over similar time intervals. Recalling that the messages used in these experiments fall into three utility

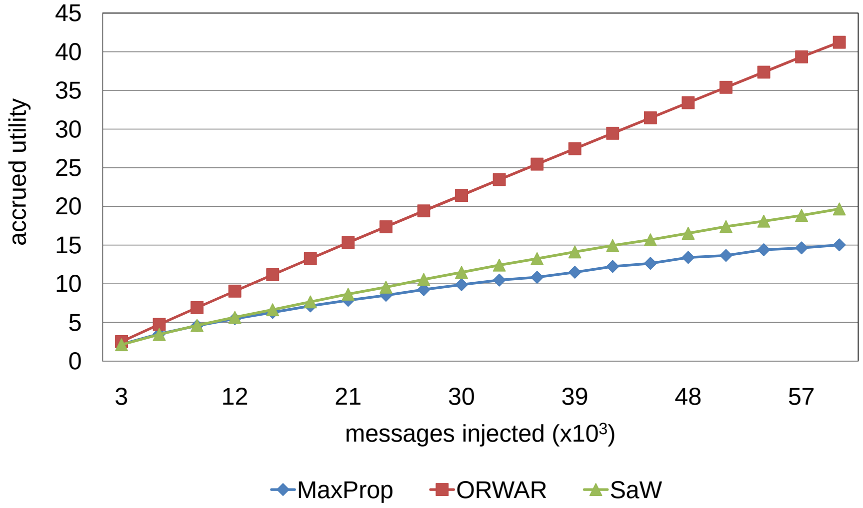


Figure 3.12 ORWAR: Accumulated utility versus load

classes, accumulated utility has been computed in the same way for all algorithms, as a function of the number of messages delivered in each utility class, as follows:

$$U_A = \sum_{c}^{1} u_i \times n_i \tag{3.3}$$

where:

where:	U_A	=	accumulated utility
	u_i	=	message utility class (see table 3.1)
	n_i	=	# messages delivered within the class
	c	=	# utility classes (in our case 3)

The higher accumulated utility for ORWAR can be explained by a higher replication factor for high utility messages, and the deletion of low utility messages first. All in all, ORWAR shows a better performance as compared to the best two of the alternative protocols.

3.5.5 Mobility implications

The experiments have also aimed at demonstrating how node speed affects ORWAR performance, and more precisely when related to contact window estimation. Figures 3.8 and 3.9 show that, if only messages that run a good chance to arrive within the contact window are sent, partial transmissions can be diminished without affecting delivery ratios, and this may even result in an increase of the total data delivered over a given time period. These gains correspond to the medium speed shown in Table II. These measurements are going to be extended in the next figure for other speeds, as defined in Table 3.2.

Table 3.2 Different speeds test bed

Speed	Pedestrians	Cars and trams
High	3.9-10.8 km/h	20-100 km/h
Medium	1.8-5.4 km/h	10-50 km/h
Low	0.8-3.7 km/h	5-25 km/h

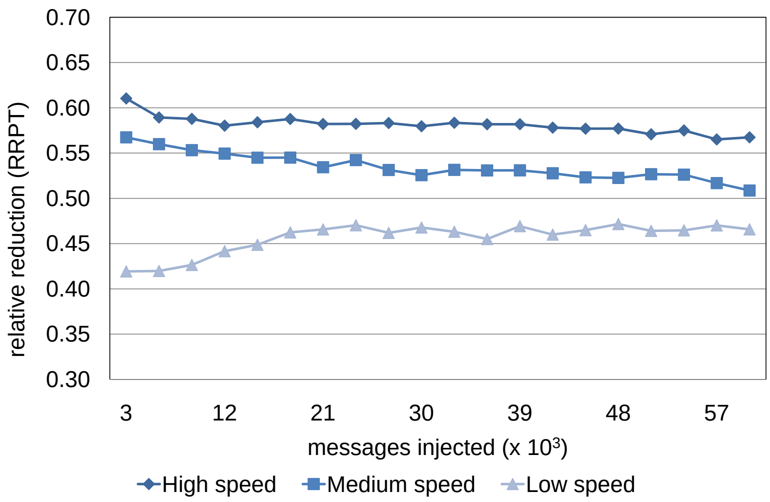


Figure 3.13 ORWAR: Partial transmissions versus load

Figure 3.13 shows the relative reduction of partial transmissions (RRPT) when using contact windows, defined as:

$$RRPT = 1 - S_O / S_{OwoCW} \tag{3.4}$$

where: S_O = total data volume lost due to message abortion using ORWAR **with** Contact Window estimation.

S_{OwoCW} = total data volume lost due to message abortion using ORWAR **without** Contact Window estimation.

Irrespective of speed and message load, the relative reduction of partial transmissions is shown to range between 40% and 60%. It also appears that gains are more significant at higher speeds. At medium speed - which is the most likely situation to find in a city scenario - the gain is still significant (around 55%).

3.6 Summary

This thesis is concerned with the development of tools meant to improve the efficiency of store-carry-forward schemes deployed over DTNs. This chapter has aimed at achieving a better understanding of local network properties that could contribute to making efficient routing or forwarding decisions in routing protocols. The analysis has focused on properties that can be sensed by the nodes themselves in a distributed fashion, such as speed, direction of movement, and radio range. The protocol presented and analysed is ORWAR, a store-carry-forward algorithm that takes into account the limited nature of contact windows between nodes. Another significant mechanism used by this algorithm is message differentiation based on the concept of utility. Utility is not seen as a system-internal parameter used to enforce internal policies at system level, but it is rather directly specified for each message. The criterion on which differentiation is based is message utility/bit. Additionally, the algorithm also includes other convenient mechanisms, such as controlled replication or the deletion of copies remaining in custodian buffers once messages have been delivered.

By using a simulation setting, this chapter has illustrated the superior performance of the ORWAR algorithm in comparison with five existing algorithms (Direct Delivery, Epidemic [VB00], Prophet [LDS04], MaxProp [BGJL06] and SprayAndWait [SPR05]), including detailed studies in relation to the closest algorithms (SprayAndWait and MaxProp). The notion of message utility has been implemented in the simulation environment ONE [KOK09], where three classes of messages have been generated with equal probability. The analysis showed that ORWAR has a similar delivery ratio to MaxProp while creating far less overhead. It also shows a 10% higher delivery ratio as compared to SprayAndWait, with an overhead that is approximately 10% lower. Since ORWAR generates little overhead, its relative performance will increase at higher loads. This chapter has shown that the benefit of using ORWAR is particularly significant when having to deal with larger messages. ORWAR is the first protocol proposing a routing scheme well-suited for large message sizes with no fragmentation, and taking account of resource optimisation at the same time.

After it was first presented in a paper, the ORWAR protocol [SNT08] was the object of study for 2 master theses. Fredrik Herbertsson implemented a NS-3 [NS310] simulation study [Her10] including a calculation of overheads in lower level protocols. His results are largely confirmed by the simulation studies that are part of this thesis. Later on, Davide

Anzaldi implemented a chat application on top of this protocol, based on the Android platform [Anz10]. Although the limited number of devices available for the testbed (only 2) was insufficient to provide a full picture of the real-life performances of this protocol, this implementation laid out an interesting direction for future work in this field.

The overview to this chapter, Section 3.1, put forth the argument that, since bundle architecture relies on a non-conversational model for data transmission, DTN applications tend to favour relatively big bundles. Forwarding big bundles over limited contact windows is a key issue in DTNs that can be solved either 1) by calibrating the size of the messages selected for sending to the actual size of the contact opportunity, or 2) by fragmentation. While the current chapter has been dedicated to an in-depth study of the former option, the next chapter will be devoted to an analysis of the latter.

Chapter 4

Adding erasure coding to a replication based protocol

Chapter 3 was dedicated to ORWAR, an efficient routing protocol considering bundles as indivisible data units. Among others, the chapter made reference to one of the mechanisms built into ORWAR, contact window estimation, which is a method for accommodating reasonably big-sized messages over limited contact windows, without a need for message fragmentation. An alternative method for coping with big-sized messages to be transferred over limited contact windows would be the fragmentation of original messages into smaller data units, which will be presented in this chapter.

4.1 Overview

While straightforward fragmentation may lead to a reduction in routing performance, since all message fragments - up to the very last - need to be received before a message can be reconstructed at destination, adding some redundancy to the system could improve overall performance at the cost of a small overhead only. One way of doing this is to erasure code a message, while adding some redundancy, and distribute the generated code-blocks over a large number of relays, each block being obviously smaller in size than the original message. This chapter presents an extension of the ORWAR framework to include fragmentation and redundancy, a method that may yield substantial gains in delivery ratio on top of the base protocol. Although this chapter evaluates gains obtained in routing performance when erasure coding is added to ORWAR, the technique is supposed to work in any other store-carry-forward protocol.

Chapter 4 will look at scenarios where higher cost infrastructure networks may also be available to a set of nodes in addition to the low-cost, best-effort delay-tolerant network.

Such scenarios provide a good opportunity for improving delivery ratios in a network while making maximum use of the resources available at lower cost. In other words, while most message fragments are delivered using the low-cost delay-tolerant network, the *last few dropped fragments* are retrieved via the infrastructure network. Hence, the DTN-based delivery method is also shown to work in such hybrid contexts.

As a target application for this method one can imagine an urban scenario where all nodes (i.e. cars, buses, or pedestrians using handheld devices) are mobile and use short-range interfaces for opportunistic communications, as well as, to a lesser extent, wide-range interfaces for infrastructure communications. Section 4.4 evokes a crisis management scenario where the (cellular) infrastructure is highly overloaded, and demonstrates how an opportunistic scheme can be combined with a selective use of infrastructure resources to increase the capacity of the DTN network while putting only a minimal extra load on the infrastructure network.

4.2 Related work

The first part of this section contains an overview of related work on fragmentation and coding, while the second part explores the literature on hybrid networks. The relevant literature concerning the replication mechanism, as well as some of the prominent routing protocols in DTNs have already been covered in the related work section of the previous chapter.

In an attempt to increase reliability in delay-tolerant networks, Wang et al. [WJMF05] and Jain et al. [JDPF05] have proposed a mechanism whereby messages are first reincoded with erasure codes, also allowing for a certain degree of redundancy, and then various message parts are distributed over a large number of different paths, so that the original messages can be reconstituted even if not all parts are eventually received. Moreover, nodes may send out packets with linear combinations of previously received information, which leads to network coding [WB05]. However, network coding adds a new level of complexity to the scheme as packets are combined, split, and then retransmitted over different links. In addition, knowledge about topology is usually needed to choose the appropriate links to which packets should be sent, which is rather difficult in opportunistic networks such as those considered in this thesis.

Pitkänen et al. show [PO07] how an appropriate redundancy ratio can then be selected in order to improve the delivery ratio while keeping the volume transferred over the network at an acceptable limit. On the other hand, a continuous fragment generation mechanism, such as that described by Byers et al. [BLMR98] and tested in fully connected networks, overcomes the need for receiving a fixed set of messages before reconstruction can occur, but this is difficult to implement in DTNs because of the high latencies associated with this type of network, and because of the need for two-way communication between source and destination in order to complete message transmission.

In the context given by Pocket Switched Networks [HCS$^+$05] focusing on human mobility, the Haggle project [SHCD06] proposed a new architecture allowing nodes to benefit transparently from more than one network. It is suggested that nodes found in many real life scenarios (or settings) may have several interfaces connecting them simultaneously to 1) infrastructure networks, 2) fully connected ad-hoc networks (neighbourhood), as well as 3) nodes carried by users (mobility). In the proposed architecture all these 3 network types cooperate in order to deliver messages towards their destination. It is then suggested that applications should not be forced to specify endpoints using IP-like addresses, but rather higher layer information, such as URLs, that can be managed at the application layer using asynchronous communication. Nevertheless, this architecture does not define exactly how communication between these networks may operate.

Bellavista et al. have studied hybrid networks [BG10] using a cross-layer approach combining the network layer (L3 approach) with the application layer (L7 approach). They claim that managing intermittent paths at the network layer (L3 approach) can yield good performance levels when paths are short and relatively stable, but this is done at the expense of flexibility. Instead, application-layer solutions (L7 approach) can achieve much better flexibility levels, for instance by enabling the exploitation of different multi-hop heterogeneous paths crossing the same node, at the expense of a relatively greater overhead. However, the scenarios considered in this thesis are based on the assumption that paths from source to destination are neither short, nor stable. Moreover, these scenarios also assume very low connectivity levels (i.e. most connections are between no more than 2 nodes at a time), so that exploiting the higher layers, i.e. the bundle or the application layer, is the only choice left.

Hybrid networks, consisting of fully connected mobile ad-hoc networks (MANETs) and infrastructure networks, have been studied by Andronache et al. [ABR08], who have proposed an algorithm for selecting the right cluster heads in order to disseminate information in a reliable way and at the best cost. However, the mechanisms presented in their paper, for instance clustering algorithms, consider fully connected MANETs, and cannot be deployed over delay-tolerant networks because of the insufficient connectivity that is characteristic of such networks.

In another work, Whitbeck et al. [WC09] have proposed a routing protocol called HYMAD (Hybrid MANET & DTN) which combines techniques from traditional mobile adhoc routing with DTN approaches. This scenario considers a network at the borderline between DTNs and fully connected MANETs, which contains disjoint groups of fully-connected nodes. HYMAD periodically scans for network topology changes and builds temporary disjoint groups of connected nodes. Intra-group delivery is performed by a conventional ad-hoc routing protocol and inter-group delivery by a DTN protocol. However, there are some differences between the scenario described above and the proposal presented in this chapter. First, while HYMAD assumes that all nodes are part of a single network evincing either MANET or DTN features at different times and different locations, the proposal in this chapter assumes that all nodes are simultaneously included in two different networks, with distinct characteristics. Second, HYMAD does not consider fragmentation/redundancy as a possibility. Although the two protocols rely on different asumptions, in both cases there is a benefit to be derived from exploiting the fully connected network over the disconnected network.

In yet another work, Lakkakorpi et al. [LPO10] also consider a scenario including DTN and MANET networks. As opposed to HYMAD [WC09] mentioned above, this scenario considers nodes to be homogeneously distributed in space, so that the networks may not be easily divided into groups with full connectivity coupled with intermittently connected groups. The authors propose a method that dynamically chooses a routing agent between 1) an ad-hoc on-demand distance vector routing protocol (AODV) with end-to-end TCP delivery, and 2) a delay-tolerant network (DTN) routing with a bundle protocol hop-by-hop delivery. It is demonstrated how, depending on local node density, node speed and message size, it may make more sense to route using one routing agent or the other. Particularly when node density is low, node speed is high, and messages sent are large, DTN routing appears to be more appropriate (and things also hold the other way round). However, this scenario also fails to consider fragmentation/redundancy in the DTN part.

4.3 Introducing erasure coding

The idea behind erasure coding is that, instead of sending a full copy of a message over a relay, only a fraction of the code-blocks is sent over each relay. However, transferring a large amount of data in small fragments without explicit acknowledgements may lead to a degradation of reliability in best-effort networks, such as DTNs, because all fragments need to arrive at destination in order for the initial message to be reconstructed. A *redundancy factor* k_r can then be chosen so that a message of size s is split into n, s/m-sized blocks ($k_r = n/m, k_r \geq 1$). Encoding is based on Reed-Solomon [Pla97] or Tornado [BLM99] codes which allow the reconstruction of the original message from any m different fragments arrived at destination. For the purposes of the method proposed in this chapter, the choice of an exact erasure coding algorithm is not particularly important. Rather, the focus is on showing that fragmentation and coding are particularly worthwhile in case only the *last few* fragments need to be pulled via an alternative infrastructure that is more reliable, but also highly expensive.

This approach puts forward a practical solution where messages are erasure coded at source, one of the parameters applied being the (fixed) size of a fragment. Then each fragment is routed independently over potentially different paths and the message is reconstructed at destination. Obviously, this strategy can also be applied to any other store-carry-forward protocol. This mechanism not only increases network reliability, but also solves the problem of delivery over a limited contact window, since smaller fragment sizes are used.

4.4 DTN and infrastructure - a hybrid approach

This section proposes a hybrid architecture, and demonstrates the gains that can be obtained if erasure coding is indeed an option, and some or all nodes support it. More concretely, this section shows that the delivery ratio can be improved by 30-35% just by using fragmentation/redundancy with ORWAR. The figures presented in Section 4.5 illustrate the results obtained in the absence of any other complementary infrastructure network. The next step is a study of the benefits that can be derived from having some help from some other (e.g. cellular) infrastructure in order to boost the advantages of fragmentation in a hybrid DTN context.

4.4.1 Proposed architecture

Many papers on DTN in a mobile context mention delivery ratios below 90%, and delivery ratios around 10% are not unusual, which is barely acceptable in real life scenarios. From a practical point of view, most mobile devices nowadays come with two or more network interfaces. Interfaces such as Wi-Fi or Bluetooth can be used in a first phase on a partitioned mobile ad-hoc base. Others, such as UMTS/HSDPA, WIMAX can then be used only to complete what failed to be delivered in the first phase. As mentioned earlier, this section presents a scheme that reaps the benefits of using a cheap but less reliable ad-hoc network (DTN) in combination with a high-cost but reliable alternative network, based on infrastructure.

Another fact making this architecture more appealing is that the infrastructure network is usually billed per usage, therefore less usage means lower overall costs.

In general, mobile ad-hoc networks, and in particular DTNs, consist of heterogeneous nodes that keep joining and leaving the network independently. The architecture proposed for this network is also meant to accommodate a small subset of nodes that do not support fragmentation. Consequently, network coding is no longer an option in this architecture, as it is based on the assumption that all nodes know how to decompose a received message and then send linear combinations of message fragments along various paths. Therefore a more portable option would be to apply erasure coding and to send fragments over different paths. By doing this, messages are (proactively) fragmented at the source node and every data block is routed independently towards the destination node. As a result, no nodes - with the exception of the source and destination nodes - are required to support fragmentation.

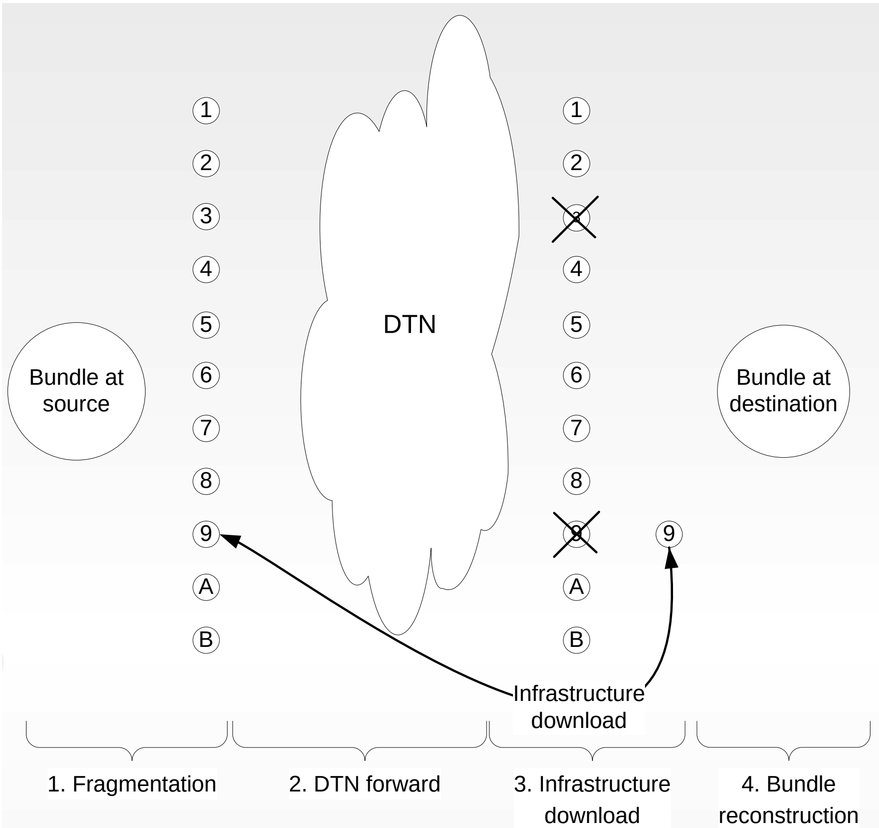


Figure 4.1 Hybrid mechanism

Figure 4.1 explains the proposed mechanism materialised in 4 steps:

1. **Fragmentation/Coding**: A bundle of size s is encoded into n, s/m sized data-blocks (where $n \geq m$). Encoding is based on Reed-Solomon or Tornado codes which allow reconstruction of the original message from any m different fragments arrived at destination. In this example, a 500kB message is encoded into $n = 11$ fragments (identified in Figure 4.1 as messages 1 to B, in hexadecimal) each of 50kB, meaning that a 10% redundancy is used ($m = 10$).

2. **DTN forward**: All these fragments are forwarded independently (the *push* phase) over a DTN network; some can arrive at destination, while others cannot. In this example fragments 3 and 9 have failed to arrive at destination.

3. **Infrastructure download**: The destination node requires m different fragments for reconstructing the message and decides to download the rest via infrastructure (the *pull* phase). In this example only 1 fragment is still needed (10 different messages at destination) and fragment 9 is chosen.

4. **Reconstruction**: The message is reconstructed from the m different fragments available ($m = 10$).

It is important to note that most transmissions are made over a DTN network and that infrastructure is used only to complete the transfer when the first mechanism fails to deliver. Fragmentation and erasure coding are the key to the scheme, allowing infrastructure downloads to be kept at a minimum. A comprehensive example of an application would be a disaster management scenario in an urban environment. In the case of a large-scale disaster (earthquake, bushfire, tsunami), infrastructure may be partially damaged. But even when the infrastructure is not damaged at all, it may still not be prepared to handle the peak of communication associated with such events. In this case, routing as much data as possible over intermittently connected networks (DTNs), and retrieving only the last few bits over the infrastructure may be a viable solution from both a technical and an economic point of view.

4.4.2 Evaluation of fragmentation size

As the bundle layer is usualy agnostic about lower layers, the DTN architecture cannot rely on them for fragmentation. Instead, it is the responsibility of upper layers (bundle layer, application layer) to limit the impact of a challenged environment by using smaller data units. There is no notion of Maximum Transmission Unit (MTU), defined as the largest protocol data unit that the layer can pass onward, according to the models to be found in IP networks. Because of the large delays in multi-hop DTNs, it is practically impossible to find a suitable data size unit according to the model of Path MTU Discovery in IP Networks. However, as confirmed by the simulation in section 4.4.2, there is such a thing as the best-fitted fragment size, which is shown to be related to the smallest contact window within a network with a given mobility.

Proactive fragmentation is used in the architecture proposed, which means that fragmentation is decided in advance at the source node, and then every fragment is routed towards destination independently. Three criteria need to be analysed before selecting fragmentation size:

1. Fragmentation size (s_f) should not exceed the expected maximum transferable size determined by the contact window. This corresponds to the s_{max} already introduced in *ORWAR* and this limit is relevant for every contact in the network. To put it differently:

$$s_f < \min(s_{max,i,j}) \tag{4.1}$$

where:	s_f	=	size used for fragmentation
	$s_{max,i,j}$	=	maximum transferable message between node i and node j

   As shown in previous sections, s_{max} depends on the maximum relative speed between two nodes, as well as on radio range.

2. Using fine-grained fragments helps minimise infrastructure download. This scheme allows the downloading via the infrastructure of only the fragments that are needed. As the size of each individual fragment decreases, the system will download a smaller volume through the infrastructure, which is also helped by a higher degree of fragmentation.

3. Too much fragmentation increases latency. The bundle is reconstructed at destination when *the last of the* m fragments arrives at destination. When many small fragments are used, the larger number of fragments implies a multiplication of paths. Therefore, multiplication of paths increases latency, as end-to-end latency corresponds to the highest latency path.

Nodes may not possess knowledge about the global mobility model, but they do know what their own movement properties are, which means that they are also aware of their own speeds. The mobility model may be homogeneous, therefore composed of nodes observing a single mobility model, in other words moving roughly at the same speed and in a given perimeter. For these cases it is of course easy to calculate a fragmentation size based on the worst case contact window, using equation 4.1. Node heterogeneity, which may be interpreted for instance as different node types moving at different speeds (e.g. pedestrians, cars, and trams in a city scenario) is not a problem as long as nodes may record the speed of their meeting partners. The real limitation for using the proposed approach appears when a network includes nodes observing various mobility models; the problem in this case is that these mobility models are circumscribed to different spaces, in such a way that not every node can meet every other node. These particular network types, where node mobility is segmented into regions, should be treated as a special case.

To conclude, when sending a fragmented message over a DTN network, one of the following 3 cases may occur:

- **Complete deliveries (CD)** - when the bundle can be reconstructed at destination from fragments arriving only through DTN (at least m out of n fragments are delivered at destination)
- **Partial deliveries (PD)** - when k fragments arrive at destination $(1 \leq k < m)$, meaning that the bundle can be reconstructed only by downloading the remaining necessary fragments via infrastructure
- **No delivery (ND)** - meaning that no fragment is delivered at destination. This case is the worst because the destination node does not know that a bundle has been sent, therefore it does not know when to initiate an infrastructure download.

4.5 Performance evaluation

The hybrid scheme presented above has been evaluated in ONE [KOK09], where a fragmentation-aware ORWAR protocol and a fragmentation-aware SprayAndWait [SPR05] protocol have been implemented.

4.5.1 Simulation setup

The evaluations below are also based on a city setup including 126 nodes (80 pedestrians, 40 cars, 6 trams) sharing a 4500 m x 3500 m playground. Every point plotted in the figures presented in this section is the result of 10 measurements where initial node positions and initial directions of movement have been varied. The confidence interval is relatively small, i.e. 1-4% of the average value. Each node is assumed to have a network interface allowing a transmission range of 10 m for pedestrians and 20 m for cars and trams. A transmission speed of 250 kBps (2 Mbps) is considered for both cars and trams. Buffers are considered to be 5 MB, except for trams, that have 50 MB buffers.

As in Chapter 3, mobility is chosen to be map-based, with pedestrians, cars and trams walking and driving along the roads of downtown Helsinki. Car speeds are set in the interval [10, 50] km/h, pedestrians move at speeds between [1.8, 5.4] km/h, and pauses are random. The network is still very sparse, with the accumulated transmission area for all nodes being 0.25% of the playground, and total meeting time accounting for about 3%

of elapsed time. Each simulation runs for 12 hours and message TTL is considered to be infinite.

4.5.2 Fragmentation size

The three charts that are presented in this section illustrate simulations where 1000 messages are injected into the network, size distribution being the standard message set (S) defined previously in Figure 3.5. Fragmentation size ranges between no fragmentation at all and fragments of up to 1kB, while redundancy remains constant at 10%. In this section $L = 12$ and $\Delta = 4$ are used as ORWAR's own parameters. This is due to the fact that more copies can be accommodated when fragments are smaller.

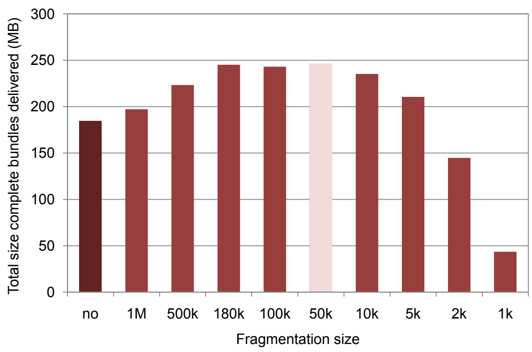


Figure 4.2 Complete bundles delivered (CD) total size vs. fragmentation size

In these three figures, the best results (50 kB fragments) are highlighted in light colours, while the results measured in the absence of fragmentation are shown in dark colours. All results are the average of 10 runs with different seeds (initial node positioning, direction and speed).

Figure 4.2 shows the total size of all completed bundles **CD** arrived at destination via DTN mechanisms, meaning that partial deliveries **PD** are excluded from this count. The delivery ratio reaches a peak when fragment size is between 180 kB and 50 kB. Indeed, calculations indicate that the worst case contact window in this scenario would correspond to 2 cars running in opposite directions at 50 km/h, which also corresponds to 180 kB. This confirms the intuition formulated in Section 4.4.2 concerning the optimal fragmentation size and its relation to the worst-case contact window.

Figure 4.3 confirms that, due to path multiplication, latency increases with fragmentation. Due to this increase in latency and in the presence of bounded TTLs, more downloads are made via the expensive infrastructure. This figure also confirms the intuition formulated in Section 4.4.2, which basically says that over-fragmentation above an ideal value is not worth the effort.

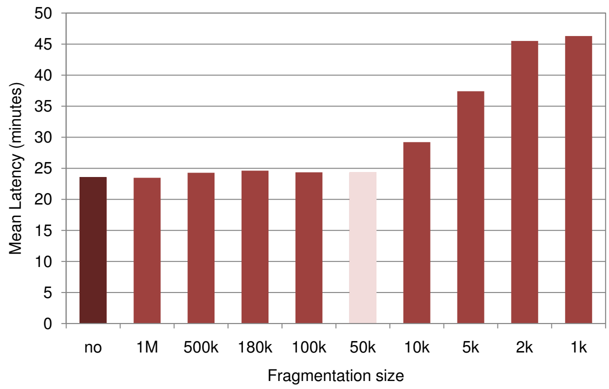


Figure 4.3 Mean latency vs. fragmentation size

Figure 4.4 shows the total size of fragments downloaded via the infrastructure, and indicates that a fragmentation range between 50kB and 180kB continues to be interesting but, when fine-grained fragmentation is used, the part retrieved via infrastructure is reduced, as expected.

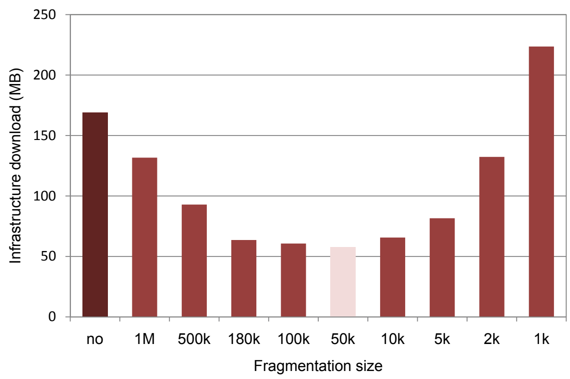


Figure 4.4 Infrastructure download vs. fragmentation size

The benefits of fragmentation are obvious here: instead of downloading 170MB via the infrastructure (which would be the total amount of lost messages in the absence of fragmentation - as shown by the chartbar in Figure 4.4) only a third of this is pulled when using a hybrid scheme with 50kB sized fragmentation.

4.5.3 Load implications

The previous sections have demonstrated the performance of ORWAR, showing that it performs well against other protocols when increasing loads are injected into the system. This is due to the fact that message utility is also taken into account when forwarding decisions are made.

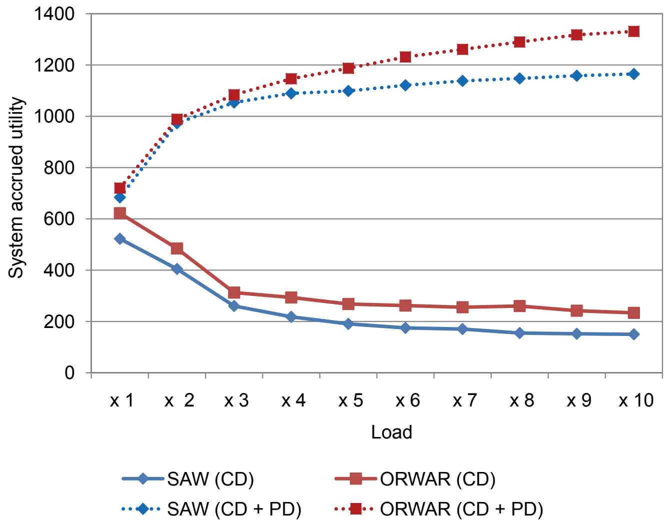


Figure 4.5 System level accrued utility vs. load

However, fragmentation implies that the contact window now has a negligible effect, since only small (50kB) fragments are involved, which also results in almost no partial transmissions. In this context, it would be useful to demonstrate that the competitive advantage of ORWAR is at least maintained when fragmentation is used. This can be done by comparing ORWAR and SprayAndWait, as well as their performance, when using fragmentation. Load is simulated by increasing the initial message size by factors of up to 10 and by keeping fragmentation size constant (50kB). Therefore, in the following two figures the load is represented on the X axis by points labeled: x1 .... x10.

As load grows, the number of partial deliveries (PD) increases at the expense of complete deliveries (CD). Although fewer and fewer complete bundles are delivered at destination, if the hybrid DTN /Infrastructure scheme is used, eventually all messages can be reconstructed at destination by pulling the missing fragments via the more expensive infrastructure.

Figure 4.5 plots system utility taking into account complete deliveries (CD) as well as total deliveries (CD + PD). Partial deliveries are also beneficial, as they contribute to limiting costly download via infrastructure. The chart indicates that, as loads get higher, the number of bundles delivered completely to destination decreases, and the number of PD bundles increases instead. ORWAR shows better resilience to load than SprayAndWait. System-wide utility increases with added load, with roughly 15-25% improvement over SprayAndWait. This shows that utility-driven routing decisions are worthwhile, especially in highly loaded scenarios.

ORWAR was designed with the goal of diminishing energy consumption by limiting overheads. So the next step would be to look at overheads (defined as total volume trans-

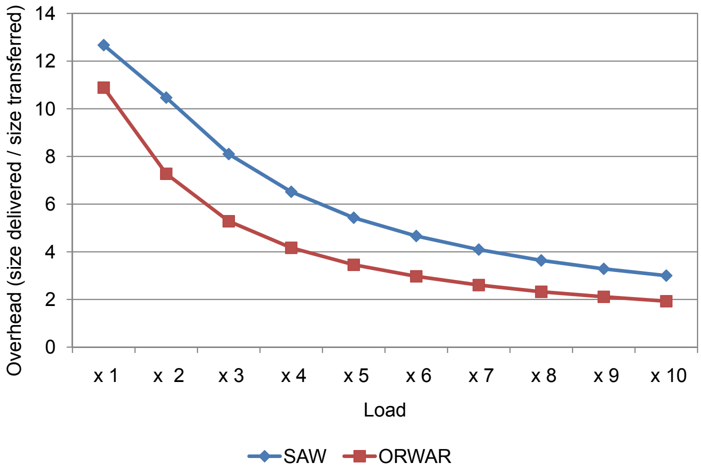


Figure 4.6 Overhead vs. load

mitted / total volume arrived at destination) retaining only the DTN part. Figure 4.6 shows that although ORWAR delivers more in terms of both delivery ratio and utility, the number of transmissions remains relatively small. The competitive advantage against SprayAndWait is also about 20% and it is fairly independent of load.

This subsection has shown that, even when fragment sizes are smaller than the contact window, meaning that the calculation of the contact window does not yield too many benefits, other in-built ORWAR mechanisms remain of interest.

4.6 Summary

This chapter has analysed the effects of adding erasure coding to the ORWAR protocol, and has shown that this improves the performance of the original protocol. The chapter also contains an analysis of the relation between fragment size and the mobility model, and presents an experimental evaluation of fragmentation size, which confirms the theory presented in Section 4.4.2.

The chapter also presents a scheme based on proactive fragmentation, where every fragment obtained by erasure coding from the original message at the source node is routed towards the destination using virtually any routing protocol. Only the source and the destination nodes should be able to encode/decode messages into/from their respective fragments, which means that this approach facilitates the security and confidentiality of communication. Moreover, no custodian node, more exactly no other node except the source and destination nodes, needs to implement fragmentation in any way. Therefore, this approach may be extended to any other routing protocol. However, there is a question that remains unanswered: how much redundancy should be added to the ORWAR protocol in order to achieve the desired delivery metrics in point of delivery ratio or mean latency? This is a rather complex question, given that the degree of redundancy to be added by erasure coding will overlap with the message replication mechanism already in-built into a couple of routing protocols. Chapter 5 will try to come up with an answer to that question.

A detailed study of a hybrid DTN/Infrastructure scheme was demonstrated to lead to the routing of most of the traffic via the DTN network, allowing the pulling of the remaining lost fragments via the infrastructure. The advantages of this hybrid scheme may be analysed from two perspectives. First, store-carry-forward schemes have generally been presented as a better-than-nothing solution. However, users usually expect all data units sent to be delivered from source to destination. In order to do this, the proposed hybrid DTN/Infrastructure uses fragmentation as an instrument to minimise load on infrastructure, and makes use mostly of the DTN network, whose costs are lower. Second, even if cost were not an issue, sending all the traffic over the infrastructure network would not be a solution either. When communicating in a disaster area, the attempt to handle all data over the infrastructure network would lead to massive overloading and communication failure. An intelligent split between a cheap but unreliable DTN network and the more reliable infrastructure may not only be more cost-effective, but may also increase network survivability.

Chapter 5

Optimising redundancy over replication

The literature on delay- and disruption-tolerant networks contains an important number of protocols using redundancy as a mechanism to increase the probability of delivering a message to destination. ORWAR, presented in Chapter 3, is an example of such a protocol. Subsequently, Chapter 4 has demonstrated that erasure coding original messages and adding a certain redundancy helps increase protocol performance. Obviously both mechanisms use up resources. What this chapter does is to analyse how these two methods can be combined in order to optimise the use of resources for achieving a given protocol metrics.

5.1 Overview

Opportunistic routing in delay-tolerant networks makes no assumptions about node contact schedules. As mentioned earlier, in order to increase the chances of successful delivery, some opportunistic routing algorithms including ORWAR transmit multiple copies of each message to several custodians, which obviously consumes resources in proportion to the number of copies transmitted. As a result, multi-copy schemes such as Epidemic [VB00] are confronted with the problem of excess traffic overhead, which wastes network resources. Other protocols, such as SprayAndWait and ORWAR, limit the number of message copies maximally found in the network, up to a limit called the *replication factor*.

An alternative method for improving opportunistic routing performance, also discussed in the previous chapter, is to erasure code a message, and then distribute the generated code-blocks potentially over multiple paths. Sending smaller data units in an opportunistic network uses up resources (bandwidth, buffer space, and energy) at a finer granularity. Instead of sending a copy of a message over one relay, only a fraction of the code-blocks is sent at each forwarding opportunity. A *redundancy factor* k is then chosen so that a message of size s is split into n, s/m-sized blocks ($k = n/m, k \geq 1$) and then sent in order to increase the probability that at least n blocks arrive at destination. This allows the reconstruction of the original message from any m different fragments arriving at destination.

This chapter considers the following scheme: the initial message is erasure coded at source by applying a redundancy factor k, and then every fragment is delivered independently by a protocol using controlled replication based on a factor L. The impact on performance of the combined scheme is analysed as an instance of an opportunistic routing mechanism that employs both controlled replication and redundancy. In concrete terms, this means building an approximate mathematical model that describes the probability of successful delivery of a message as a function of various parameters, such as time-to-live, number of nodes, fragment size, and degree of replication and redundancy, respectively. This model proves to be an effective means for optimising performance for a given time-to-live parameter while minimising the overheads associated with replication and redundancy.

To summarise, the contributions of this chapter are the following: (1) a mathematical model for optimising combined replication and redundancy parameters on a per message basis, or on a network basis for uniform networks, and (2) a proposal to adapt the parameters over time, depending on actual behaviour as well as on recorded latencies.

The model proposed in Section 5.3 computes the probability for a complete message to be delivered before a given time-to-live elapses, the computation being based on fragment size, as derived from network mobility assumptions. Thus, a scheme is provided for determining replication and redundancy pairs (L, k), from which one particular pair (L, k) is selected and is then optimised from a cost perspective. The model includes a probability distribution component for message latency. This component is initially derived from earlier equations for expected latency as proposed by Spyropoulos et al. [SPR08a, SPR08b]. These equations are adapted by (1) integrating erasure coding into the model, and (2) adapting it to resource-constrained networks. This contributes to making the model more realistic, as it revises the original assumption that contact time and buffer space do not impose limitations on forwarding. Numerical analyses as well as network simulations have been used to validate the model and to demonstrate the practical benefits of using this optimisation.

Finally, a framework for optimising the generic protocol using controlled replication and erasure coding is defined. It works in two phases, the initiation phase and the adaptation phase. In the initiation phase the basic model is used to obtain the presumed optimised redundancy and replication factors, given a desired probability of message reconstruction at destination within a given time-to-live. When running the system with the parameters thus obtained, actual data on average latency is stored for various source-destination pairs. In the adaptation phase, the recorded distribution for (real) average latency replaces the initial approximation, thereby producing new settings for the replication and redundancy factors that are closer to the characteristics of the network, the mobility model, and the protocol.

This makes the method generically applicable to any opportunistic protocol that uses controlled replication and erasure coding, e.g. the ORWAR routing protocol as introduced in Chapter 3 and extended with erasure coding in Chapter 4. While the former two chapters describe protocol performance using simulation runs, this chapter adds a new element by proposing a mathematical framework for optimisation.

5.2 Related work

There are numerous examples of protocols using message replication as a mechanism to increase delivery ratio. Replication may be seen as flooding, such as in different variations of the Epidemic scheme [VB00], or as controlled replication, such as in SprayAndWait [SPR05]. Under some simplifying assumptions, such as basic mobility patterns, a large number of nodes and small message sizes, some of these protocols can be studied analytically. Zhang et al. [ZNKT07] have obtained a rich set of closed form formulas on delivery delay and on number of copies sent under the following extended epidemic schemes: k-hop forwarding, probabilistic forwarding, and limited-time forwarding. One of these extended schemes, probabilistic forwarding, was first proposed by Lindgren et al. [LDS04] and further developed by Haas et al. [HS06]. Another approach proposed by Lindgren et al. [LDS04] and continued by Spyropoulos et al. in SprayAndFocus [SPR07] was that encounter history could determine future delivery probability. Section 5.6 leverages on this by reusing history to aid optimisation. More specifically, delivery history is used to get delivery time distribution that is further used in cost optimisation.

Message dissemination time in a network has been studied analytically in two papers that are briefly presented below. Fracchia et al. have proposed a mathematical model [FM08] based on vehicular traffic density in order to compute the average delay of small-sized signalisation messages. Agarwal et al. [ASL08] have analytically calculated the lower and upper bounds of latency in a delay-tolerant setting considering exponential distribution of nodes in a one-dimensional highway setting. However, fragmentation is not considered in either of these two works, and specific mobility is largely unidirectional, as imposed by the subjacent vehicular traffic.

In a paper dedicated to message fragmentation, Pitkänen et al. [PKO08] discuss the choice between proactive fragmentation and reactive fragmentation and show that both can accommodate limited contact windows. This chapter takes up the idea of proactive fragmentation as a result of erasure coding and argues that parameters such as replication factor, redundancy factor, and data unit size can be initially optimised based on an analytical model, and later adapted to real mobility conditions. The relation between node speed, radio range, and the mobility pattern was also studied by Groenvelt et al. [GNK05] using a rigorous analytical framework, but the approach was limited to three mobility patterns and, more importantly, it assumed infinite bandwidth during contacts.

In another theoretical paper, Altman et al. [ABP10] proposed a trade-off between transmission delay and energy consumption by letting a replication protocol deliberately bypass some contact opportunities - in order to conserve energy - and still deliver within the required delay. However, message coding was not considered, nor any other solution that takes into account limited contact windows or real message sizes. The benefit of this solution, since messages are split by erasure coding, is that connectivity can also take advantage of short contact opportunities, particularly in very dynamic networks. In a different approach, Lin et al. [LLL08] studied the mechanism of network coding as an add-on to replication. However, network coding and erasure coding are essentially different. One significant difference is that, unlike erasure coding, network coding requires all intermediary nodes to be aware of the protocols involved, with important computational requirements, that are rather burdensome and have a heavy overhead. On the other hand, this thesis proposes a solution relying on source-based erasure coding, so that computation is only involved at source, for message splitting, and at destination, for message reconstruction. Moreover, as the routing mechanism between source and destination is not affected, the optimisation mechanism is expected to remain suitable for a wide range of routing protocols using replication.

5.3 The mathematical model

SprayAndWait represents the archetypal approach to a protocol using replication. However, the original version does not use redundancy and assumes infinite available bandwidth, meaning that an indefinite number of messages can be exchanged over a given meeting between two nodes. This section proposes an approximate probability distribution component for message latency for this protocol by imposing some constraints on message forwarding.

5.3.1 Assumptions

The basic layout includes a set of M nodes moving in a closed area A, and various source-destination pairs. The mobile nodes are equipped with a wireless device with a fixed transmission range r that is small as compared to area A. All nodes are assumed to move according to some mobility models whose meeting times are almost exponentially distributed. At each meeting, nodes can exchange a limited number of messages according to contact window, message size, and available buffer space at custodian, this being captured by the forwarding probability β - assumed to be uniform over the network. Moreover, considering a small β, only small subsets of the message list will be forwarded at each contact, thus multiplying message paths between source and destination.

Messages are erasure coded at source into a number of fragments of constant size f; this size can be defined according to a set of criteria, as proposed earlier under Section 4.4.2. The fragments are then routed to destination over independent paths. The assumption that fragments are routed over independent paths appears reasonable considering that M is big enough and nodes abstain from transmitting more than one fragment of the same message during one single meeting.

As for erasure coding at source and reconstruction at destination, they are considered to occur with negligible delay and with negligible effect on system resources (primarily bandwidth and transmission energy).

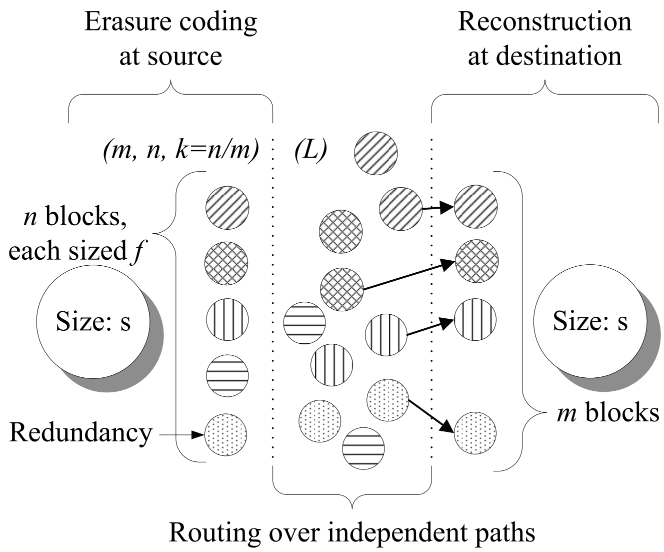


Figure 5.1 Routing using replication and erasure coding

Figure 5.1 shows an initial message being erasure coded, then routed to destination over independent paths. In this example $n = 5$, $m = 4$, $k = 1.25$, and $L = 3$. No further assumptions are made about the routing protocol, except that it should rely on controlled replication and allow erasure coding.

5.3.2 Expected delay boundaries without network constraints and erasure coding

In order to determine the probability of a message being delivered before a certain time, it is first necessary to study the arrival times of messages in a network that uses replication. Earlier work has studied delivery metrics using binary SprayAndWait with no erasure coding in a uniform network, and this is an element that will be reused here.

An important parameter that allows calculating the expected delay is pairwise meeting time $\tau_{(a,b)}$, which represents the time between two successive meetings of two arbitrary nodes a and b. This parameter is considered [ZNKT07, GNK05, BCF07] to be well approximated by the Poisson distribution, so that the probability $P\{\tau_{(a,b)}, k\}$ for $\tau_{(a,b)}$ to have exactly k occurrences is given by:

$$P\{\tau_{(a,b)}, k\} = \frac{\lambda^k e^{-\lambda}}{k!} \tag{5.1}$$

Considering that processes $\{\tau_{(a,b)}\}$ between various node pairs a and b are mutually independent and homogeneous, Groenvelt et al. [GNK05] show, using Markov chains, that these processes are again of the Poisson type for all the nodes in the network, and their intensity λ can be calculated for 3 mobility models (random walk, random waypoint, and random direction). Mean pairwise meeting time $\tau = 1/\lambda$ will be used henceforth as a model parameter. Table 5.1 lists all the variables to be used in the model proposed in this chapter.

Given a particular network model, performance can be estimated in the presence of the optimal algorithm, known as Oracle routing. Spyropoulos et al. [SPR08b] show that expected delivery time can be approximated as follows:

$$T_o = \frac{\sum_{i=1}^{M-1}(1/i)}{(M-1)} * \tau \tag{5.2}$$

T_o acts as a lower bound on expected delay for every other routing algorithm that would have been deployed in this given network. For a network deploying replication, in another work [SPR08a], Spyropoulos et al. detail the upper bound for the expected delay when a binary SprayAndWait routing protocol is used:

$$T_{sw} = \underbrace{\sum_{i=1}^{L-1}\left(\frac{\tau}{(M-i)}\right)}_{\text{forwarding (spray)}} + \underbrace{\left(\frac{(M-L)\tau}{(M-1)L}\right)}_{\text{final delivery (wait)}} \tag{5.3}$$

Variable	**Definition**
τ	mean pairwise meeting time
M	total number of nodes in the network
λ	Poisson process intensity (or slope) considering M number of nodes
T_o	expected latency under Oracle routing
T_{sw}	expected latency under binary SprayAndWait routing
$T_{sw}(n)$	expected latency under binary SprayAndWait routing and erasure coding, with n fragments
β	forwarding probability of a message during a contact
L	maximum number of copies in the network (protocol parameter for controlled replication)
T	message time-to-live
p_T	probability that a message is delivered before time T
$p_T(n)$	probability that n fragments are delivered before time T
s	message size
f	fragment size
m	number of fragments (excluding redundancy, calculated as $\lceil s/f \rceil$)
n	total number of fragments per message with redundancy
k	redundancy factor ($k = n/m$)

Table 5.1 Summary of variables used

Equation (5.3) highlights the two components of the SprayAndWait delay: the forwarding or spraying phase, and the final delivery, also known as the waiting phase.

5.3.3 Delivery time distribution without erasure coding

It can be stated, therefore, that for a given replication factor (L) and network parameters (M and τ), a message routed by SprayAndWait will have an expected latency bounded by the following two values: T_o and T_{sw}. The next question is then: what is the shape of distribution for delivery delays between these two bounds?

Extensive simulations have been conducted to define the distribution of arrival latency for messages arriving between T_o and T_{sw}. Figure 5.2 shows the distribution of delivery times gathered in the simulation environment for two different replication rates L when using classical SprayAndWait routing. The x-axis shows 10-minute time intervals, while the y-axis indicates the corresponding share of the total number of messages delivered within each time interval. Results are simulated using ONE [KOK09] in a 100-node network[1] moving in a random waypoint pattern. Other simulations (not shown here) confirm that the shape is very similar also when nodes follow a random walker and map-based mobility model, as well as when different replication rates L are used.

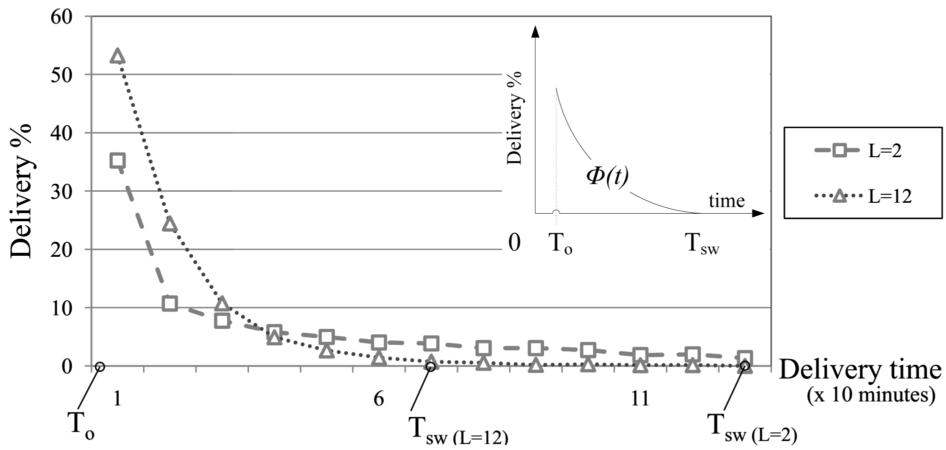


Figure 5.2 Latency distribution - simulation results

If the above behaviour can be approximated with a mathematical distribution, then it can be used to calculate the probability of a message to be delivered before its time-to-live T expires. In most cases, message latency is below T_{sw} and a meaningful approximation should cover a large area under the curve by time T_{sw}.

Figure 5.2 shows a simple logarithmic approximation that can be noted as $\Phi(t) = \gamma * ln(t) + \delta$ with $\Phi(T_{sw}) = 0$ and $\Phi(T_o) = 1$. γ and δ can be calculated from the above conditions. Moreover, by integrating $\Phi(t)$ between T_o and T, the expected probability p_T that the message will be successfully delivered before time T can be calculated as follows:

$$p_T = \begin{cases} 0 & \text{if } T < T_o \\ \dfrac{ln(\dfrac{T}{T_o}) - (T - T_o)ln(\dfrac{e}{T_{sw}})}{ln(\dfrac{T_{sw}}{T_o}) - (T_{sw} - T_o)ln(\dfrac{e}{T_{sw}})} & \text{if } T_o < T < T_{sw} \\ 1 & \text{if } T > T_{sw} \end{cases} \tag{5.4}$$

Equation (5.4) shows that, under these assumptions, the probability p_T of a message

[1]Section 5.5 provides details of the simulation setup.

reaching its destination before time T can be approximated using Oracle and SprayAndWait estimates. These can simply be determined by Equations 5.2 and 5.3, provided that M, L and τ are known. For this equation, simple message delivery without erasure coding, as well as successful forwarding at every physical encounter are assumed. The next section extends the model to deal with these two factors.

5.3.4 Delivery time distribution with erasure coding and constrained resources

In this subsection, the equations presented above are augmented to deal with cases where (1) erasure coding is used, and (2) message forwarding is not automatically successful at every physical encounter. The upper bound for expected latency for n fragments can be calculated as being:

$$T_{sw}(n) = \sum_{i=1}^{L-1}\left(\frac{\tau}{\beta * (M-i)}\right) + \frac{(M-L)\tau}{(M-1)L} + \sum_{i=1}^{n-1}\left(\frac{\tau}{\beta * (M-i)}\right) \qquad (5.5)$$

Two changes have been made here as compared to equation (5.3): the introduction of β and the addition of a third element in the formula. If a homogeneous network with exponential inter-meeting times statistics is considered, sending a message would be successful only in β proportion of the cases. Thus τ from equation (5.3) translates into τ/β in equation (5.5). However, because β intervenes exclusively in the forwarding process, this change affects only the *spray* element in formula (5.3). The *wait* element, which represents the time needed for nodes to meet their final destination after all copies are sprayed, is unaffected by β since transmission success is assumed at each physical meeting in case of direct delivery.

As compared to non-fragmented forwarding (equation (5.3)), the fragmented case, *at worst*, needs $n-1$ physical encounters to forward the n^{th} fragment through the network, which is reflected in the third component of the formula (5.5). That is, the worst case expected latency corresponds to the situation when each fragment is forwarded to a different custodian. This provides the time at which the message can be reconstructed at destination, supposing erasure coding into n fragments is performed at source.

Equation (5.5) shows that message latency is affected by both a replication factor L and a redundancy factor n. In order to isolate these two effects, and to reveal only the latency component associated with redundancy, we define the time span δ_n. δ_n represents the time span between expected latency with no fragmentation and expected latency for the n^{th} fragment. It is easy to show that for a given M and L:

$$\delta_n = T_{sw}(n) - T_{sw}(1) = \sum_{i=1}^{n-1} \left(\frac{\tau}{\beta * (M - i)} \right) \tag{5.6}$$

δ_n reflects the networking latency overhead due to erasure coding. This equation will be revisited in the evaluation section.

5.4 Proposed approach to optimisation

The goal here is to detect a set of equi-potent pairs (L, k) yielding the (same) required average latency in the network. The condition of average latency means that at least m fragments arrive at destination before T expires so that the message can be reconstructed at the destination node. This function can be analysed in a first instance using the mathematical model and, for the most generic case, it can be completed by analysing actual network performance.

5.4.1 The equilibrium function

Under the assumptions formulated in subsection 5.3.1, this function can be numerically deduced from the mathematical model presented in Section 5.3. In other words, given a replication factor L_i, the number of nodes M, the number of fragments $m = \lceil s/f \rceil$, mean pairwise meeting time τ, forwarding probability β and time-to-live T, the aim is to find an appropriate n_i such that the message can be reconstructed at destination with a given probability p_r before time T. Because this function aims at finding the *minimum* number of fragments n_i, it is called the *equilibrium function*. The calculation method includes two phases that are outlined in Figure 5.3.

$$\underbrace{\begin{bmatrix} M, \tau, \beta, T, m \\ \begin{pmatrix} L_1 \\ L_2 \\ \dots \\ L_i \\ \dots \\ L_l \end{pmatrix} \end{bmatrix}}_{\text{Input parameters}} \quad \overset{\text{Eq 5.4}}{\underset{\text{(Phase 1)}}{\Longrightarrow}} \quad \underbrace{\begin{pmatrix} p_{1,L_1} & p_{2,L_1} & \dots & p_{j,L_1} \\ p_{1,L_2} & p_{2,L_2} & \dots & p_{j,L_2} \\ \dots & \dots & \dots & \dots \\ p_{1,L_i} & p_{2,L_i} & \dots & p_{j,L_i} \\ \dots & \dots & \dots & \dots \\ p_{1,L_l} & p_{2,L_l} & \dots & p_{j,L_l} \end{pmatrix}}_{\text{Fragment delivery probability}} \quad \begin{matrix} \forall\, i \in (1, l) \\ \text{select} \min n_i \\ \Longrightarrow \\ \text{such that} \\ Pr\{n_i \mid m\} \geq p_r \\ \text{(Phase 2)} \end{matrix} \quad \underbrace{\begin{pmatrix} k_1 = n_1/m \\ k_2 = n_2/m \\ \dots \\ k_i = n_i/m \\ \dots \\ k_l = n_l/m \end{pmatrix}}_{\text{Output parameters}}$$

Figure 5.3 (L, k) calculation method

- Phase 1: Compute expected latency under worst conditions for different fragments $(1, 2, \dots, j)$ using equation 5.4. Input parameters are M, L_i, τ, β and T

- Phase 2: Compute the minimum number of fragments n_i that should be sent by the source node such that at least m arrive at destination with the required probability p_r. This can be calculated by applying the Poisson trials theory[2].

By varying this procedure for different Ls, a series of (L, k) pairs is obtained which produce the same (given) reconstruction probability. Next, the cost optimisation function described in subsection 5.4.2 is applied.

5.4.2 Minimising delivery costs

The previous subsections presented the calculation of a series of (L, k) pairs such that a sufficient number of different fragments (at least m, to be specific) arrive at destination before expiration of message time-to-live T. The next step that needs to be made is to calculate the system resource costs related to different (L, k) pairs.

A simple assumption would be that the system resources consumed by the protocol are proportional to message size and to the total number of copies sprayed before time-to-live T expires. Let $I_f(t)$ be a function[3] describing the number of copies of fragment f at time t. Since controlled replication is used, there is one copy of the message present at protocol initialisation and L copies after spraying time. Altman et al. have shown [ABP10] that $I_f(t)$ has a sigmoidal shape which is however close to the following linear form:

$$I_f(t) = \begin{cases} L * t/T_s & \text{if } t \le T_s \\ \\ L & \text{if } t > T_s \end{cases} \tag{5.7}$$

where T_s is the spray phase time

$$T_s = \sum_{i=1}^{L-1} \left(\frac{\tau}{\beta * (M - i)} \right)$$

As opposed to replication, erasure coding at source immediately brings into the system a number of copies that remains constant in time - namely n (immediately, as erasure coding takes place before the first fragment is sent). Let $C(t)$ be a measure of resource consumption for a given message as a function of time. The assumption is that, when T is reached, the system is able to clean the messages with expired times-to-live. This implies

[2]Note that Poisson trials are applicable due to the independent arrival of fragments at destination. Repeated independent trials in which there can be only two outcomes, and where the probability for each trial is known, are called Poisson trials. Poisson trials are similar to Bernoulli trials, except that in the former each trial occurs with uneven probability.

[3]Also known as the infected nodes function.

that C reaches its peak for $t = T$. System optimisation may be attained, in this case, by minimising costs for a known m (as a result of knowing message size m and fragment size f), given a number of nodes M, and a time-to-live T before which the entire message should be reconstructed successfully at destination.

Thus, the problem of minimising delivery costs can be formulated as follows:

$$\begin{cases} \textit{calculate} & (L, k) \\ \textit{subject to minimising} & C(T) = I_f(T) * n \end{cases}$$

5.5 Simulation and results

The aim of this section is to validate the model presented in the previous section by comparing simulation results with those obtained from numerical evaluation, as described below. The objective here is (1) to provide evidence that the mathematical model is reasonable as compared to the simulations, and demonstrate its use, and (2) to present some optimisation results and obtain some insights into the relation between resource cost and protocol perfomance.

5.5.1 Simulation setup

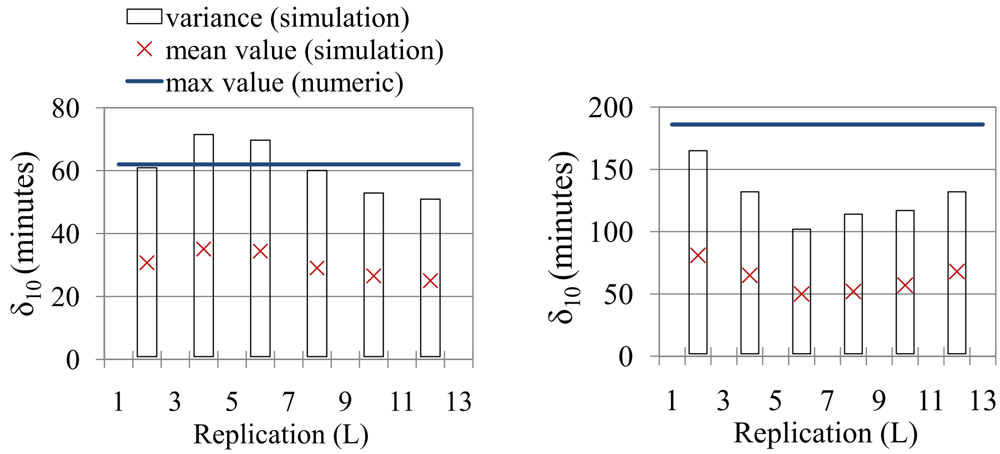
The simulation environment is set up using ONE [KOK09], built upon a delay-tolerant network of $M = 100$ nodes moving according to a random waypoint mobility model. Mean pairwise meeting time is set at $\tau = 60$ minutes, while message time-to-live is set at $T = 9$ minutes[4]. The test scenario is to inject 100 messages, each sized 1kB, from every node to a different random node while forward probability is modelled by β.

Every message is erasure coded at source and at least $m = 10$ fragments need to be delivered to allow a message to be reconstructed at destination. β has been enforced in the routing layer of ONE, in such a way that only a subset of the message list, corresponding to the given β, is forwarded at each physical meeting. Random waypoint mobility has been chosen in order to build up a homogeneous network, given the fact that the mathematical model is also built on this assumption. Every point plotted in the simulation figures is the result of 10 measurements, where the initial node position and initial direction of movement have been varied.

[4]This represents an indication of the high requirements placed on the system. Considering the mobility characterised by τ, a direct delivery algorithm would have delivered approximately 50% of the messages before τ. Consequently, the analysis of the relevant (L, k) pairs should be based on much lower T/τ ratios, such as those proposed here.

5.5.2 Validation of the model

Recall that equation (5.6) describes the latency overhead for introducing erasure coding into the network. To validate equation (5.6) - and consequently equation (5.5) - the computations arrived at by using equation (5.6) were compared with different simulation results.



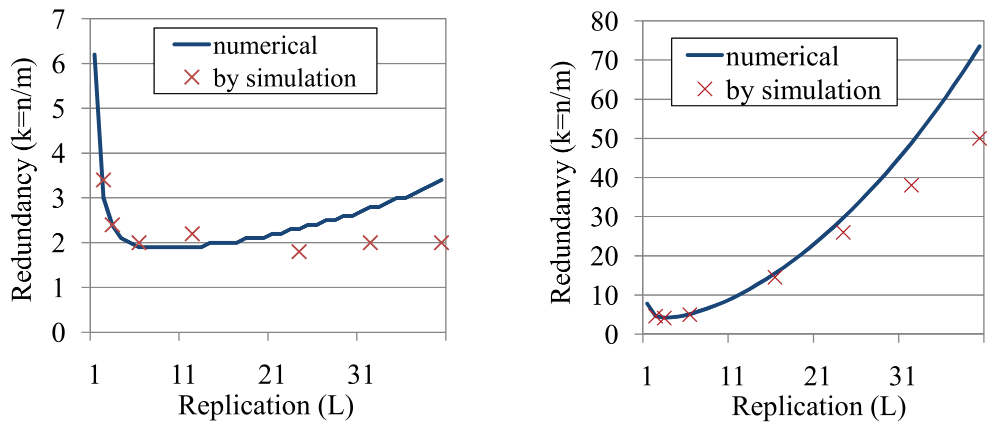
$\beta = 0.75$ $\beta = 0.25$

Figure 5.4 Approximation of mean latency due to fragmentation

Figure 5.4 plots the latency overhead with respect to 10 fragments (δ_{10}) from simulations and compares it with the computed latency overhead. It shows that the calculated δ_n is a good upper bound approximation as compared to the simulation attempts that are denoted by the crosses on the chart. It appears also that δ_n is in reverse proportion to β.

Figure 5.5 shows the result of computing a range of ks given a range of Ls using the mathematical model denoted by the numerical curve. The TTL used in these simulations has been chosen to be long enough so that it matches the average latency for the 10^{th} fragment with good accuracy (1%). Every cross on the figure is the result of 10 such experiments providing a k based on the corresponding L. The experiments have been repeated for 2 different βs. Note that changing β will allow the 'emulation' of different buffer and bandwidth availabilities, in such a way that various proportions of messages (anything between 0 and 100%) are forwarded. Experiments have also been made with two extreme values (almost perfect forwarding, 0.9, and almost non-existent forwarding, 0.1).

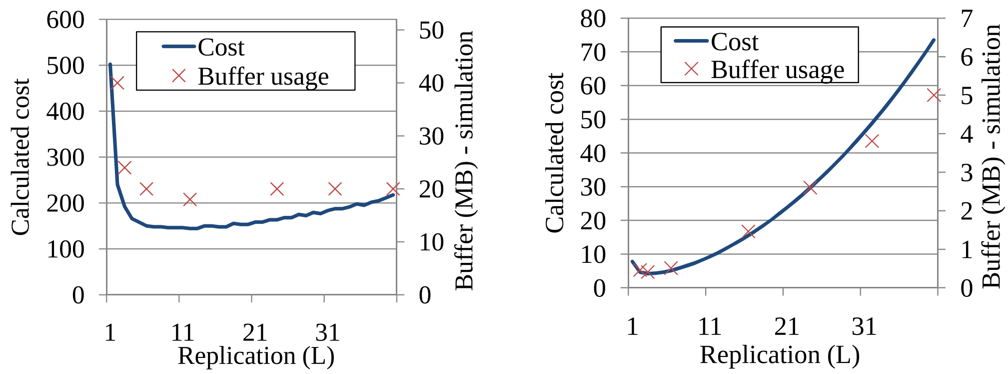
Simulations demonstrate a reasonable match with the numerical results, with closer matches for smaller values of L, as well as for smaller values of β. The multiplication of paths may explain why, for smaller βs, simulation matches numerical results better. Recall that, in this model, equation (5.5) provides expected latency under worst case conditions, when each fragment takes a different path, which is more likely to happen with a smaller



$\beta = 0.9$ $\beta = 0.1$

Figure 5.5 Replication L versus redundancy k

β.



$\beta = 0.9$ $\beta = 0.1$

Figure 5.6 Cost versus buffer occupancy

Next, it is shown that network costs, equivalent to the overall number of message copies at time T, represents a meaningful value in the actual economy of the network. Based on the same network simulation settings as before, Figure 5.6 plots calculated cost and actual buffer occupancy after T time units on all nodes, as a function of replication L. Buffer occupancy is calculated by adding up all occupied buffer space found at all node locations at the end of the simulation time. These findings also provide a good validation for this theory through simulation.

5.5.3 Optimisation results and insights

The practical benefit of the proposed optimisation scheme is that the system calculates the best suited (L, k) pair as a function of the given time-to-live T and the number of fragments required m. Message time-to-live T is the most important parameter because the aggressiveness of the protocol (also implying bigger costs) is greater when short delivery times are required. The figures below show a number of numerical studies varying a number of input parameters.

Figure 5.7 presents the cost associated with 4 different time-to-live parameters ranging from a short one ($T = 0.06\tau$) to a long one ($T = 0.5\tau$). Note that, for this network, the minimal cost is achieved for a replication factor generally falling between $L = 4$ and $L = 20$ together with a correlated amount of redundancy.

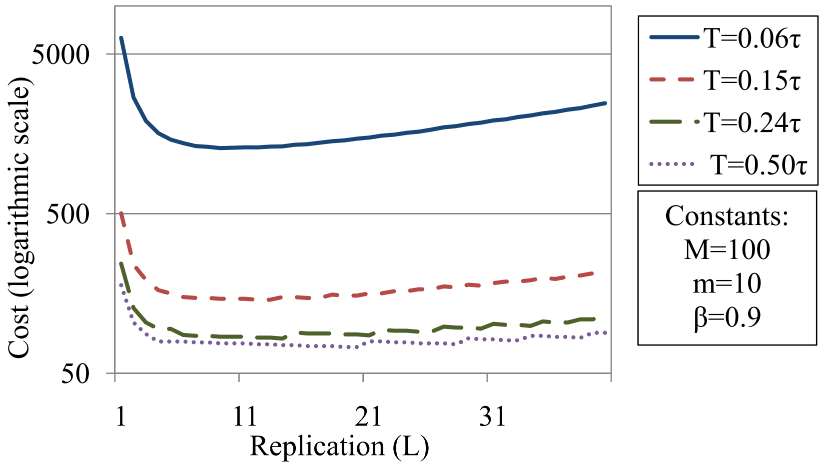


Figure 5.7 Effects of T on cost

It also appears that redundancy is a practical way to overcome the fact that the replication factor L is an integer, while the ideal L may not be one. Especially for long Ts, associated with low protocol aggressiveness, fractional values of k may be used together with an integer L in order to produce results "similar" to a fractional L.

Figure 5.8 is an attempt to see whether cost optimisation would be materially affected by the choice of m, which is ultimately influenced by message size. Supposing that the network allows a maximum fragment size f, more fragments are needed for a long message than for a short message (calculated as $m = \lceil s/f \rceil$). Figure 5.8 shows that overfragmentation is not beneficial from a cost perspective. A large m will imply more buffers being used up, which eventually and implicitly delays message reconstruction. Besides, when a high level of fragmentation is used, there may be additional drawbacks that are not studied in this thesis, but include increasing reconstruction effort at destination, additional resources

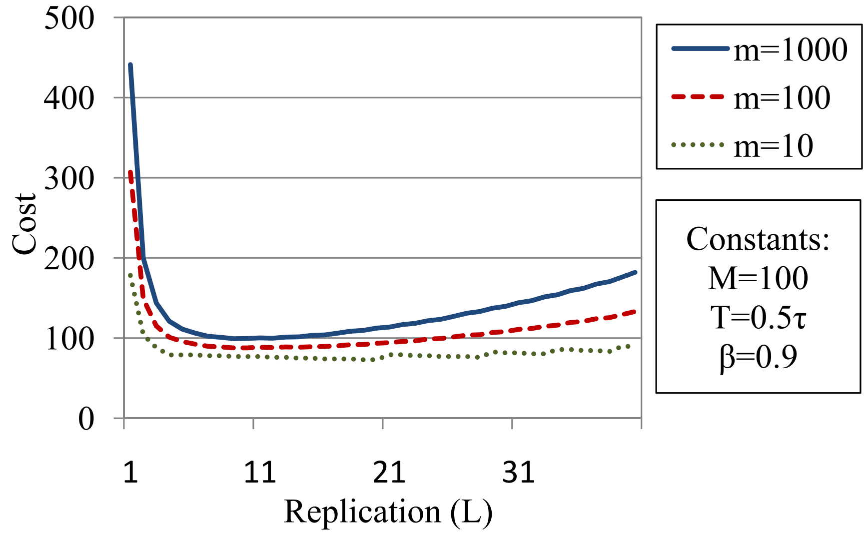


Figure 5.8 Effects of m on cost

taken up by every fragment header, and so on. All these facts speak to the need for a careful correlation between contact window and fragmentation, as proposed in section 4.5.2.

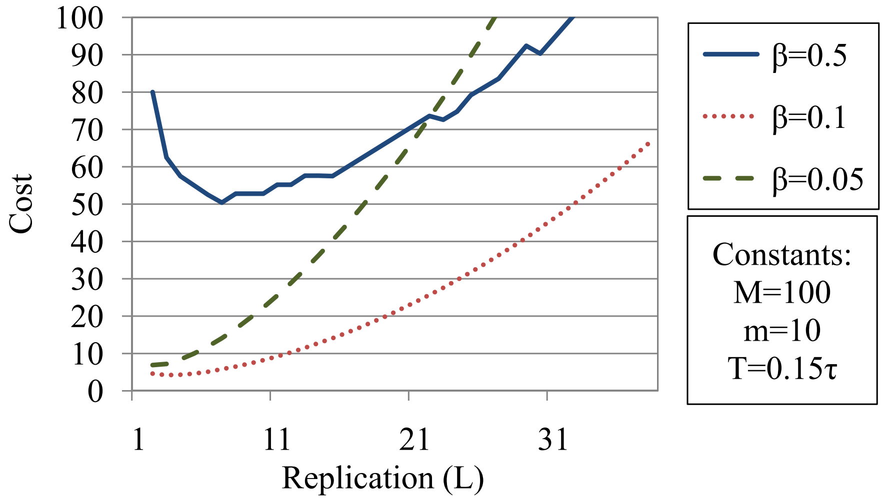


Figure 5.9 Effects of β on cost

Figure 5.9 shows how forwarding probability β affects cost evaluation. Note that the best cost is found neither at the bottom, nor at the top end of β, and this might be counterintuitive at first. In the case presented above, best cost is achieved for $\beta = 0.1$, with a significantly lower cost than for $\beta = 0.05$. Moreover, for $L < 22$, which however represents the most usable[5] range for this parameter, the cost associated with $\beta = 0.1$ is even

[5]The authors of SprayAndWait recommend L in the range of 5-20% of total network nodes

lower for the same effects than for $\beta = 0.5$. Altman et al. arrived at a similar conclusion [ABP10] by voluntarily diminishing forwarding probability in order to diminish cost, which they did with a view to saving energy. In this model, on the other hand, it is obvious that, up to a certain point, the number of copies (and implicitly cost) will decrease when no forwarding opportunities exist. In such overloaded networks, cost minimisation will then favour a high degree of redundancy over a high degree of replication. Note that the L corresponding to minimum cost decreases from $L = 8$ at $\beta = 0.5$ to a much smaller $L = 2$ corresponding to $\beta = 0.05$. All these examples confirm that the model presented in this chapter is flexible enough to handle a wide variety of cases.

5.6 Adaptation of the basic model

This section proposes an improvement of the previous model, considering now a generic distribution of delivery times instead of a slightly imprecise approximation of a logarithmic distribution between bounds, defined by T_o and T_{sw}, which was presented in section 5.3, equations 5.2 and 5.3 respectively. With this improvement, the optimisation model can be applied to any opportunistic protocol that includes controlled replication and redundancy, as long as some estimates of average latency can be obtained.

The actual distribution of delivery times $l_{i,j,L} = \{T_1, T_2, ..., T_n\}$ may be harvested from delivery history at each destination j and transmitted periodically to the relevant source i, as shown in Algorithm 2 below. Upon arrival at destination and receiving an indication of time taken to destination, node i will reconstruct the associated probability vector $p_{i,j,L} = \{p_{T_1}, p_{T_2}, ..., p_{T_n}\}$ where p_{T_n} is the probability that the n^{th} fragment reaches destination before T expires. Of course, this adds somewhat to overheads, but it can be seen as a method to extend the reach of the model to new routing protocols that have the commonality of opportunistic contacts, replication, and redundancy. Moreover, it allows a protocol to adapt to the actual network characteristics as long as past history can be assumed to have some bearing on future behaviour. Related overheads can be kept low by allowing information to be sent only when it reaches a threshold size.

In this generic case only the method of calculating probabilities is replaced, but the method of calculating n, as well as the minimisation of cost, still holds. This solution also works when a network is not homogeneous. One can also see the generic case as the second (adaptive) phase of a network operation that is initially performed using the basic model.

In the generic case, optimised parameters can be calculated as follows:

- set f for a network based on the mobility model and expected node velocities,
- for each message, calculate the number of fragments to be sent $m = \lceil s/f \rceil$,

```
n_i       //  source node
n_j       //  destination node
m_k       //  message k sent from n_i to n_j with replication factor L
l_{i,j,L} //  list of delivery times from n_i to n_j using replication
              factor L
T_k       //  delivery time of m_k from n_i to n_j
p_T       //  probability that one single message (or fragment) is
              delivered before time T

// at destination n_j
on receiving m_k from n_i do
    push T_k → l_{i,j,L};
    if size (l_{i,j,L}) ≥ threshold then
        transmit l_{i,j,L} to n_i;
        l_{i,j,L} → ∅ ;

// at source n_i
on receiving l_{i,j,L} from n_j do
    merge local l_{i,j,L} with remote l_{i,j,L};

// computing p_T at n_i for a known T
```

$$p_T(L) = \frac{countElements((l_{i,j,L})[k] \leq T)}{size(l_{i,j,L})}$$

Algorithm 1: Pseudo-code: getting p_T from history

- determine the ideal n and L corresponding to the minimum cost, according to Section 5.4.2. Assuming that delivery time distribution is transmitted from destination back to source, the ideal n and L are only a function of message time-to-live T.

5.7 Summary

This chapter has addressed several aspects of optimisation in the context of a generic, store-carry-forward protocol assuming both controlled replication and erasure coding.

The chapter has presented a mathematical model linking together controlled replication and erasure coding. By analysing the distribution of delivery times, it is possible to determine the probability for a message to be delivered using replication and erasure coding before its time-to-live expires. By applying minimisation on delivery costs, it is possible to calculate an optimised replication factor and redundancy factor for each message, taking into account message size and time-to-live.

The validity of the proposed model has been tested in simulations and a good correlation has been found between the analytical (calculated) values and those obtained in a simulation environment. The next step has been to show the effects of the various parameters involved: message time-to-live, fragmentation level, and forwarding probability. Findings have revealed that message time-to-live, forwarding probability, and to a lesser extent message size, dramatically affect the choice of replication/redundancy parameters.

Demonstrating the use of the optimisation model, this chapter has presented a set of analyses yielding the following insights: (1) an appropriate and correlated level of replication and redundancy makes a visible difference in terms of network level costs, given the same latency targets, (2) network overload, materialised as diminished available bandwidth or shortened buffer at custodians, affects cost in a significant way, for instance best cost is found at a particular value for overload, which is neither at the bottom nor at the top of the range, and (3) erasure coding a message into very small chunks - much lower values than those required by a typical contact window - negatively affects cost by ineffectively multiplying delivery paths. Initial simulations suggest that this framework is a practical tool for optimising routing parameters in a challenged network.

The model has been then enhanced from being dependent on the latency bounds calculated for a given protocol (SprayAndWait), to a generic model that adapts itself based on delivery history. In case a destination node keeps a record of delivery times and occasionally sends it to the source, this information can be used as an input for the optimisation process. Because this mechanism remains valid for an arbitrary protocol that uses replication and erasure coding, where neither upper and lower bounds, nor distribution of delivery times can be analytically determined, it represents an improvement by removing some of the model's simplifying assumptions.

Chapter 6

Vicinity resource cartography

This chapter proposes a distributed scheme for calculating the level of resources available in the vicinity of a node. A generic model is developed first, which is then applied to individual network assets, such as buffer space, energy, and bandwidth.

The model is based on a sparse network, where resources are potentially not uniformly distributed. As a consequence, there may be pockets of resources in the network, in the form of energy or buffer space, which may vary both in time and space. Incidentally, the way in which resources are distributed in a network may also be the consequence of mobility, but this model does not assume any particular mobility pattern. As an effect of sparsity, most meetings are seen as happening between two nodes.

6.1 Overview

As a rule, resources are not homogeneously distributed in real networks, so the average resource assumption usually does not hold. Real networks may be composed of handhelds, car embedded computers, laptops, fixed throwboxes and sensors, which obviously have different energy availabilities and different buffer space allowances. Moreover, neither message source nodes, nor message destination nodes need to be uniformly distributed across the node population. Therefore, network resource consumption is uneven in most practical cases. Given this resource heterogeneity, having a local estimate for the resources available in the vicinity of each node would be highly beneficial for forwarding decisions. For example, based on such estimates, a node may decide to whom to forward its messages (custodian election) and which messages should be transferred first (queue management).

In an attempt to enhance delivery performance and to minimise end-to-end delay in disconnected networks, recent studies have looked at node behaviour from several new perspectives. Some put forward the idea of social correlations between nodes, which may influence node contact distribution [YGC09], while others propose techniques based on a combination of social context and node topology information [DH07, HSL10]. These techniques select some key nodes, among those that are located more centrally in the network or are more active socially, considering them to be better suited for forwarding incoming messages. In so doing, however, they place additional strain on the resources available to these key nodes, and they will be the first to fail as a consequence of network congestion, buffer exhaustion, or battery depletion. Thus, in the absence of any metrics for the resources available to nodes, these strategies may actually jeopardise initial performance gains by losing the most important nodes through overuse.

Other studies focus on the particular movements building up higher densities around points of interest [PSDG09]. However, in opportunistic ad-hoc networks the localisation of such points of interest, acting as resource concentrators, is difficult to achieve without a priori topology knowledge. Moreover, resource concentrators may move in space and vary in time. For example, the distribution of resources in a disaster area cannot be planned in advance. Somewhat similarly, the configuration of points of interest in case of a traffic jam triggered by an accident varies in space and time as an effect of the accordion phenomenon [TLB$^+$09]. Finally, even when the geographical distribution of resources is known in advance and is constant in time, being able to detect them autonomously, that is, without central knowledge, makes a big difference in terms of protocol robustness.

Knowing the time-varying and space-varying resource distribution in a network can have a huge impact on the choice of routing strategies, even when only *approximate* knowledge is available. An important problem then is how to select custodians efficiently, i.e. depending on own resources and on resources available in the neighbourhood. For example, a node may choose not to forward a message at a particular encounter, knowing that better opportunities may show up in the near future. Another problem to consider is how to prioritise messages in a message queue so as to give them a good chance to be transferred within the limited contact window of an encounter.

As mentioned earlier, this chapter proposes a distributed scheme for estimating the resources available in the proximity of a node, with no a priori knowledge. The accuracy of this distributed scheme is studied and then validated in three different simulation settings. Space-varying and time-varying resource maps are considered separately, and validation is performed in both a random waypoint scenario and a disaster area scenario (using Bonn motion traces [AGPG$^+$07]). Then a set of policies is proposed where the estimates mentioned above are used for selecting a suitable custodian (custodian election) and for choosing the right message to transfer (queue management). Thus, such policies can be approached holistically based on the availability of the three resources under consideration in node proximity. These policies are analysed from two different perspectives: a *basic* perspective, where the number of custodians chosen cannot be defined and varies according to fluctuating system conditions, and a *controlled* perspective, where the number of custodians is determined as a fraction of the total number of encounters.

This chapter also shows that, by using information about the local availability of resources, a routing protocol may not only use up fewer resources overall, but network resources will be consumed preferentially from nodes with above average resources, while nodes with below average assets will be spared as much as possible, which helps keep the number of completely exhausted nodes at a minimum. As a result, disparities in available resources across the node population are significantly reduced, and nodes are less likely to leave the network as a consequence of resource depletion. Findings demonstrate a substantial performance enhancement, particularly in networks with heterogeneously distributed resources. They also reveal that a node choosing a fraction of about 10-20% of all the nodes encountered as custodians is a good rule of thumb for achieving best results.

To sum up, this chapter makes the following contributions: (1) proposing a distributed scheme for calculating resource availability in node vicinity; (2) validating this scheme in different scenarios, showing that its inaccuracy is below 10% in the scenarios considered; (3) proposing and then implementing custodian election and message queue strategies using all three resources in a holistic way, and (4) showing, by means of extended simulations, that the proposed strategies bring a real benefit in realistic scenarios, particularly when network resources are heterogeneously distributed.

6.2 Related work

Node heterogeneity is the immediate intuitive consequence of the fact that mobile ad-hoc networks are made up of various nodes that are freely joining and leaving the network, without clearance from a central authority. This is also the case in open DTN networks, where heterogeneity has been recognised as a central problem early on. This problem intervenes at various levels: at network level, in relation to the social affinity between nodes, or in relation to the way in which nodes can access and contribute to network resources. Heterogeneity at network level has been studied by Rais in [Rai11], who provided DTN nodes with a transparent way to benefit from connecting heterogeneous networks such as cellular, infrastructure or ad-hoc networks. The heterogeneity of node interactions and of social affinities between nodes has been studied by Leguay [Leg07] and later by Yoneki et al. [YGC09]. Leguay argues that aggregates of inter-contact and contact times are rather irrelevant, and that only *pairwise* inter-contact and contact times can be adequately exploited by a routing protocol [CLF06]. The explanation for this is that pairwise counterparts can better capture the social affinities between nodes. However, heterogeneity in his work only refers to the social aspect, and is in no way related to resources.

In [IMC10] Ioannidis et al. have identified the following sources for heterogeneity: access to resources, user preferences (a social aspect), and mobility. They have proposed PSEPHOS, a decentralised mechanism for optimal data caching, where nodes download content for instance from the Internet, and make it available to peers via a disconnected network. The question to be answered here is: which pieces of content should be cached in each node's storage, considering the demand for content in the vicinity, as well as node heterogeneity. What is interesting about this proposal is that one of the conditions set for optimisation is the maximisation of social welfare, i.e. the maximisation of accrued utility for the whole system. However, this work only considers storage space as a resource, while energy and bandwidth are missing.

To this author's best knowledge, there is no unified proposal in the literature linking together the most relevant resources in DTN: energy, buffer space, and bandwidth, into one single abstraction. In the case presented in this chapter, past encounters are used to predict the evolution of available resources in the near-term. Prediction-based schemes have already been proposed in earlier works [NGP07, YCW09], but they mostly dealt with social affinities between nodes or attempted to estimate contact probability, thus ignoring the amount of resources that nodes are contributing to and consuming from the network. On the other hand, in a heterogeneous environment, where resources range over a wide spectrum of types and levels, estimating resource availability remains an open question.

Another prediction-based protocol, MobySpace [LFC06], proposes the construction of a high-dimensional Euclidean space from node mobility patterns, while also assuming prior knowledge (or learning) of those patterns. Euclidean space has also been used in the approach proposed earlier in this thesis, but the focus has been on the resources available in node vicinity. Moreover, the equations proposed have not been tightly constrained by one mobility model or other, as demonstrated by the use of two alternative scenarios.

Heterogeneity has also been studied by Spyropoulos et al. [STO09], who propose a utility function for the selection of appropriate custodians. The utility function provides the flexibility required for mapping various node properties. However, their paper fails to account for resources in general, or bandwidth in particular, which leads to the questionable assumption that a message can be conveyed over a meeting regardless of its size.

6.3 Resource availability as a generic abstraction

Let us consider a generic resource R for a given node. Obviously, every node knows its own resource level. Let us denote with R^o_A the own resource level available at node A and with R^v_A the estimated resource available in the vicinity of node A. Applying the distributed scheme below, every node will be able to evaluate the expected virtual value of resource R available in its vicinity.

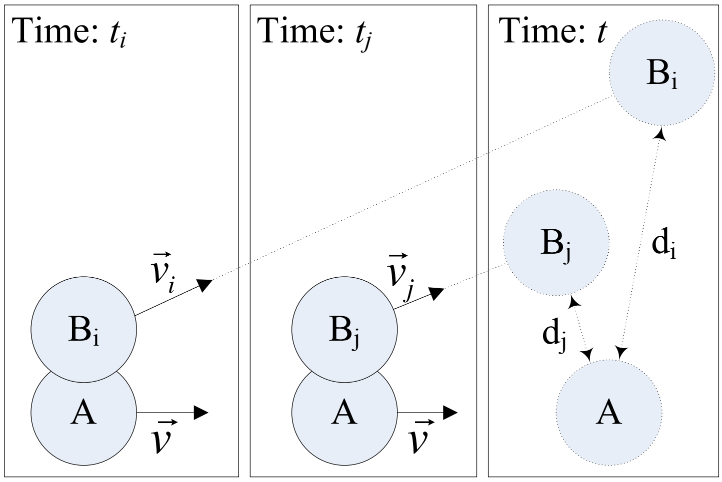


Figure 6.1 Node A meetings history within observation time span

Let us consider a node A moving at speed $\vec{v}$, and a set of nodes $B_i...B_j$ moving at speeds $\vec{v}_i...\vec{v}_j$, respectively. Every node knows its own vectorial speed and resource level, and communicates them at each meeting. For instance, when a meeting happens between node A and node B_i, node A receives vector I_i from node B_i:

$$I_i = (\vec{v}_i, R_i^o)$$

Let us consider a sequence of encounters between node A and nodes $B_i, ..., B_j$ at times $t_i, ..., t_j$ at which A collects resources $R_i^o, ..., R_j^o$ from those nodes. The assumption is that A maintains a log of maximum $j - i + 1$ encounters during a sliding time window of size τ. After meeting the last node (B_j), node A will have the following information available:

$$M_A = \begin{pmatrix} t_i & t_{i+1} & ... & t_j \\ \vec{v}_i & \vec{v}_{i+1} & ... & \vec{v}_j \\ R_i^o & R_{i+1}^o & ... & R_j^o \end{pmatrix}$$

M_A is determined by adding local time information t_k to the vector I_k received at each meeting with node B_k. The information a node needs to store is limited for two reasons: first, because the maximum number of columns of the matrix $(j - i + 1)$ is limited, and second, because all information older than τ is discarded (such as $t_j - t_i < \tau$ at any point in time).

Considering that speeds $\vec{v}$ and $\vec{v}_i, ..., \vec{v}_j$ remain constant for the short observation time span τ, the Euclidean distance between node A and nodes $B_i, ..., B_j$ at a later point in time than the encounter time can also be calculated, as shown in Figure 6.1. It is also possible to calculate the contribution of node B_k to the estimated virtual resources of A as being in reverse proportion to the distance from B_k to A. The resource footprint in the vicinity of A, denoted as R_A^v, can be computed as a function of resources met by node A during the time span τ. Finally, R_A^v is proportional to the number of meetings A had over the observation time τ. As the intention is not to limit the findings in this chapter to one particular mobility model, meeting probabilities have not been used. Instead, some derived measures have been applied. Putting all the information together, the formula below can be used to calculate the estimated available resource in the vicinity of node A:

$$R_A^v = \underbrace{\frac{n_\tau}{\omega \times \tau}}_{c_A} \times \underbrace{\frac{\sum_{k=i}^{j} \frac{R_k^o}{d_k}}{\sum_{k=i}^{j} \frac{1}{d_k}}}_{\overline{R}_A} \tag{6.1}$$

where: τ = observation time span
ω = node's average meeting frequency
n_τ = $j - i + 1$, number of nodes A actually met during observation time τ

In other words, as meetings take place and resource information is exchanged, each node builds up its own map of virtual resources, assigning greater weight to those at a shorter distance.

Equation (6.1) can be decomposed into:

- an element c_A reflecting the density of meetings in a given region. This acts as a generic factor irrespective of resource type R.
- an element $\overline{R}_A$ representing the average availability of resource R weighted by the inverse of the distance between A and the nodes met.

The validation of this formula is done in extensive scenarios in Sections 6.5 and 6.6, but the following special cases are considered here for discussion:

- if node A had no meetings over the τ time span: $\lim_{n_\tau \to 0} R_A^v = 0$ because n_τ respectively c_A is 0
- if the node actually had an average number of meetings over time span τ: $\lim_{c_A \to 1} R_A^v = \overline{R}_A$

6.4 Modelling individual resources

After having developed a generic model for calculating resources available in node vicinity, let us now move on to the second step in this modelling exercise, and refine this equation for individual network assets: buffer space, energy, and bandwidth. While buffer space and energy are properties related to one node, bandwidth is a property linking together two or more nodes. The three categories of resources are treated in an increasing order of complexity.

6.4.1 Buffer space

The buffer space case is straightforward. Equation (6.1) can be used directly for calculating buffer space by simply replacing the generic resource with buffer space in the formula. This is possible because buffer space remains constant as long as no messages are exchanged between nodes. That is, for short observation times τ and low network load, a node's buffer space at time t can be approximated with the buffer space observed at time $t_i > t - \tau$. Denoting the available buffer size with S, it is possible to replace R directly by S in Equation (6.1).

6.4.2 Energy

The energy model is more complex, because energy levels do not remain constant, even in the absence of message exchange. In case there is traffic, energy is depleted by the sending and receiving of messages at a rate approximately proportional to the size of messages exchanged. In case there is no traffic, node energy decreases simply due to network sensing. The energy level at a node, at one particular timepoint t, can be approximated by relating it to the relevant factors, as follows:

$$E^o(t) = E_{max} - \underbrace{e_s \times t}_{\text{Energy for sensing}} - \underbrace{e_m \times m}_{\text{Energy for message exchange}} \tag{6.2}$$

where: $E^o(t)$ = node's own energy at time t
E_{max} = maximum energy available for this type of node
e_s = energy factor for sensing
e_m = energy factor used for message exchange
m = total size of exchanged messages

Factors e_s and e_m can both be measured for different types of nodes in a laboratory setup [RH10]. In a simpler setup, every node can measure energy depletion as a function of time just by retrieving battery levels at 2 different times. Denoting this attenuation rate by e, the above equation is simplified as follows:

$$E^o(t) = E_{max} - \underbrace{e \times t}_{\text{attenuation with time}}$$

Thus, time variable $E^o(t)$ replaces the constant R_k^o in Equation (6.1) as a node's estimate for own energy.

6.4.3 Bandwidth

In general, for delay- and disruption-tolerant networks, the notion of bandwidth should be interpreted more generically as *connectivity*. In these networks, meeting quantity (the number of meetings between nodes) and meeting quality (contact windows during those meetings) are considered to be part of network resources.

However, for the purposes of this model, bandwidth is defined as the maximum volume of data D_N that a node N can exchange at one meeting. This model is meant to achieve a twofold objective: first, to provide an estimate for a node's capacity to send and receive messages (at one meeting or over a given time span); and second, to help determine which message to send, depending on message size and the estimated probability of success.

The theorem proposed below can be applied to estimate the volume of data exchanged between two nodes, as well as the probability of a message to pass, taking contact window estimation as a basic factor. As mentioned before, contact window denotes the time during which two nodes are in radio range of each other, and represents a critical factor for realistically evaluating bandwidth in mobile networks. Most encounters are considered to happen between only two nodes, as a result of network sparsity.

Theorem 1: For a meeting between two nodes (disk radio range with radius r), moving at a relative speed of $\vec{v}_{rel}$ and communicating over a protocol with nominal bandwidth b_n, the following elements can be calculated:

(I) the maximum volume of data exchanged during the meeting as:
$$D_{max} = \frac{2r}{|\vec{v}_{rel}|} \times b_n$$

(II) the expected volume of data exchanged during the meeting as:
$$D_{exp} = \frac{\pi r}{2|\vec{v}_{rel}|} \times b_n$$

(III) the probability of an exchange exceeding a given size:
$Pr\{D \geq pD_{max}\} = \sqrt{1-p^2}$ where $p \in [0,1]$

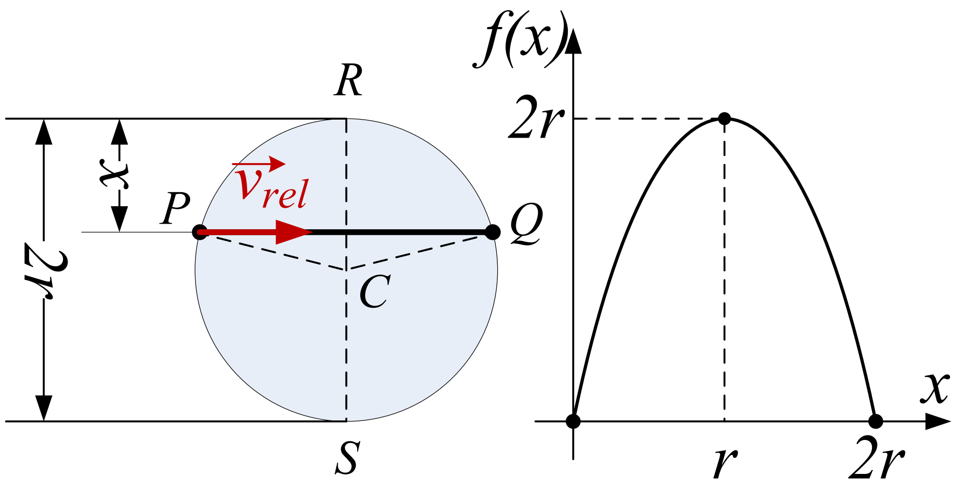


Figure 6.2 Contact window as a function of incidence point

Proof: As shown in Figure 6.2, a node crosses the radio range of another node on a trajectory $\overline{PQ}$. This trajectory is covered at a speed $\vec{v}_{rel}$ and P is the incidence point between the nodes' radio ranges. Using geometry, the trajectory between P and Q can be calculated as a function of x:

$$f(x) = \overline{PQ} = 2\sqrt{x(2r-x)}$$

If the initial assumption is that nodes meet, $x \in [0, 2r]$ and contact point P may be anywhere on the circle arc $\widehat{RPS}$. Moreover, in the generic case, x is a random variable uniformly distributed over the interval $[0, 2r]$. Thus, D_{max} can be calculated as:

$$D_{max} = \frac{\overline{PQ}_{max}}{|\vec{v}_{rel}|} \times b_n = \frac{2r}{|\vec{v}_{rel}|} \times b_n \tag{I}$$

$$D_{exp} = \frac{\overline{PQ}_{exp}}{|\vec{v}_{rel}|} \times b_n = \frac{\dfrac{\int_0^{2r} f(x)\,dx}{2r}}{|\vec{v}_{rel}|} \times b_n =$$

$$\frac{2r^2 \arctan(\dfrac{\sqrt{x}}{\sqrt{2r-x}}) + \sqrt{x(2r-x)}(x-r)|_0^{2r}}{2r|\vec{v}_{rel}|} \times b_n =$$

$$\frac{\pi r}{2|\vec{v}_{rel}|} \times b_n \tag{II}$$

The next step is to calculate the probability of having an exchange exceeding a given fraction $p \in [0, 1]$ of the maximum D_{max} as:

$$Pr\{D \geq pD_{max}\} = Pr\{f(x) \geq p2r\} =$$
$$Pr\{2\sqrt{x(2r-x)} \geq 2pr\} = \frac{|x_1 - x_2|}{2r} = \sqrt{1-p^2} \quad \text{(III)}$$

where x_1 and x_2 are the solutions of the quadratic equation $2\sqrt{x(2r-x)} = 2pr$ □

Theorem 1 can be used to estimate one particular network asset – bandwidth – characterised at one node, namely the expected data volume that this node can exchange at one particular meeting. Bandwidth can now be easily integrated into this model for generic resources in order to estimate the expected bandwidth value in the vicinity of a node, as follows. At each meeting time $t_i, ..., t_j$ a node knows its own speed as well as that of its neighbours, so it can calculate the relative speeds $v_{rel,i}, ..., v_{rel,j}$. In a given scenario, r and b_n are approximated as known constants. Therefore, by using Theorem 1.II, each node can calculate the array of expected data to be exchanged ($D_{exp,i}, ..., D_{exp,j}$) and, by using Equation (6.1), it can estimate the expected amount of data D_A^v that can be exchanged by node A over the observation time τ.

One additional application of Theorem 1 is that, if node A actively manages its message queue, it can also choose which message to send by evaluating the probability of successful message transmission given by Theorem 1.III.

6.5 Validation

This section demonstrates how the virtual resources in the neighbourhood of a node can be estimated by applying Equation (6.1) upon exchanges between nodes, as proposed in Section 6.3. The goal of validation is to show that these estimates are indeed close enough to the real resource levels in the network. The validation exercise has been organised in two sections: this section validates the model applied to buffer space in three different scenarios, while Section 6.6 validates a holistic approach that considers all three network resources taken together.

6.5.1 Space-varying environment

As shown by Hyytiä et al. [HLV06], when nodes move according to random waypoint mobility in a square, node density is maximum in the middle of the area and decreases to 0 towards the borders. Based on this observation, Equation (6.1) can be validated in a simple, yet revealing experiment where the time-varying element can be isolated from the space-varying element. Assuming constant buffer space per node (i.e. nodes do not exchange large messages, but only small amounts of information, as required for Equation (6.1)), virtual buffer distribution is given predominantly by node densities and is constant in time.

The ONE simulation environment [KOK09] was used in order to perform the validation. The network setup consists in a set of 100 nodes moving at 20 m/s according to random waypoint mobility within a space defined by a square with an edge length of 1800 m, as well as a ”spy” node A moving very slowly at 0.04 m/s on a rectilinear path starting from the middle of the northern edge and ending in the middle of the southern edge. Initially, all nodes have a fixed amount of buffer space (500 MB) and, since no message is delivered between nodes, the resource map is defined exclusively by node density. The random waypoint movement is modelled by creating 100 different movements and computing the average value of R^v_A over time, according to Equation (6.1). An observation period of $\tau = 5$ minutes is considered over a 12-hour scenario, while the number of columns M_A is limited to 40. Figure 6.3 plots the values calculated for R^v_A (as recorded by the ”spy” node A using Equation (6.1) for buffer space) versus the values given by the baseline (node density probability mass computed according to the formula presented in [HLV06]), for the same ”spy” node. As can be seen in Figure 6.3, Equation (6.1) closely follows the baseline values for buffer space in the vicinity of the ”spy” node.

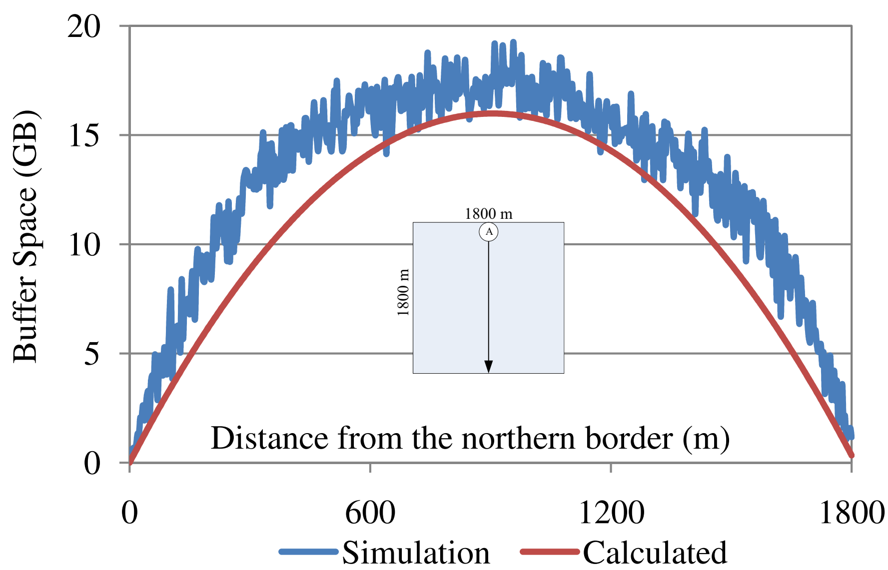


Figure 6.3 Available buffer space in the vicinity of a node, space-varying scenario

An explanation for the slight difference between the baseline and the calculated values may be that while baseline values consider a 0 radio range (calculating only node densities), values calculated using Equation (6.1) consider a radio range of 20 m.

6.5.2 Space- and time-varying environment

In the previous subsection, the spatial distribution of resources provided a simple baseline. However, choosing a baseline becomes more complicated if a space- and time-varying model, or a non-synthetic mobility model is considered. This subsection proposes two scenarios, with two different baseline alternatives:

- **future encounters**, calculated as the sum of own resources (R^o) of all peers that will be actually met by the observed node over a reference timespan[1] τ_f.
- **cell resources**, calculated as the sum of own resources (R^o) of all nodes sharing the same cell as the node itself at a given time. (Cells are obtained by dividing the simulation playground into a number of equal squares.)

Random waypoint scenario

In this scenario, the setting was again a 100-node network performing random waypoint movement in a 1800 m $\times$ 1800 m square playground. A large number of messages was then injected into this network over a short period of time, which reduced available buffer space in most nodes to a minimum level. This was followed by a period of slow recovery in buffer space, as the messages were gradually delivered and therefore deleted from the buffers.

Figure 6.4 shows the evolution in time of R_A^v for a representative node, using buffer space data, as compared to the future encounters baseline. However, comparable accuracy can be found for all the other 99 nodes. If inaccuracy I is defined as the distance between the calculated curve $c(t)$, and the baseline curve $b(t)$, the following formula can be proposed:

$$I = \frac{\displaystyle\int_0^T |c(t) - b(t)|\, dt}{\displaystyle\int_0^T b(t)\, dt} \tag{6.3}$$

where T is the simulation time (12h). By applying this formula, inaccuracy I is calculated to range between 2% and 6% for all 100 nodes in this scenario, for both baselines proposed: future encounters and cell resources.

[1]For the sake of simplicity $\tau_f = \tau$

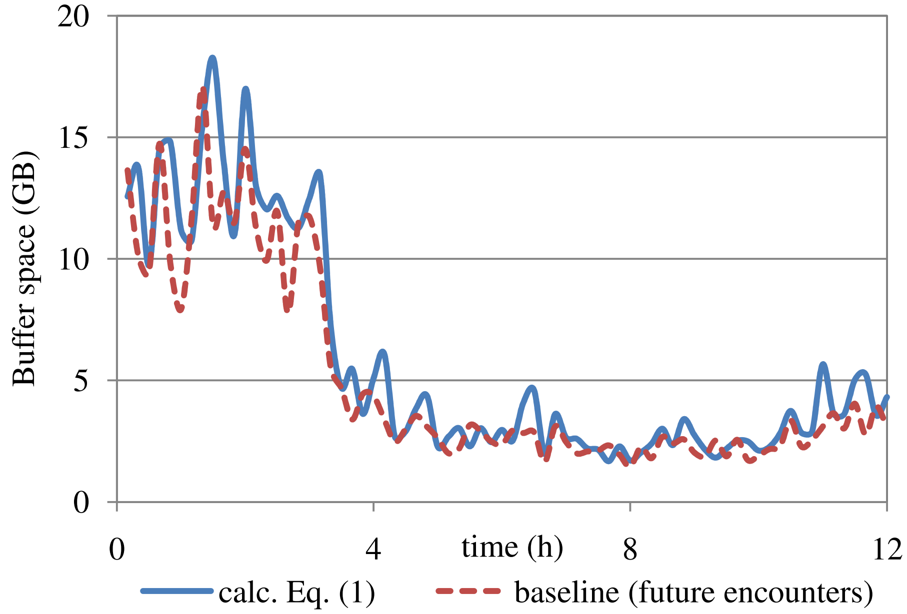


Figure 6.4 Available buffer space in the vicinity of a node, time- and space-varying scenario, random waypoint

Disaster area scenario

The next case considers a 150-node network moving according to a disaster management scenario as described earlier [AGPG$^+$07], known as Bonn Motion. In order to create some heterogeneity in the system, nodes were divided into 3 groups, each including 50 nodes. Only the first group of nodes were injecting messages over the first half of the 12 h scenario, addressing them uniformly to all nodes. Buffer space allocation was also uneven: it was 500 MB for the first two groups of nodes, and only 50 MB for the third. Message sizes were fixed, the playground was 360 m $\times$ 170 m and the cell used to calculate the cell resource baseline was 10 m $\times$ 10 m.

Figure 6.5 presents the evolution of R_A^v for one particular node, calculated for buffer space, compared to the cell resources baseline. For this scenario, maximum inaccuracy as compared to cell resource baseline is 10% and typically below 6%.

It has been demonstrated therefore that the generic notion of virtual vicinity resource can be used in two mobility contexts to estimate buffer resources with a low level of inaccuracy. This has been done considering both time-varying parameters (load and movement) and space-varying parameters (movement).

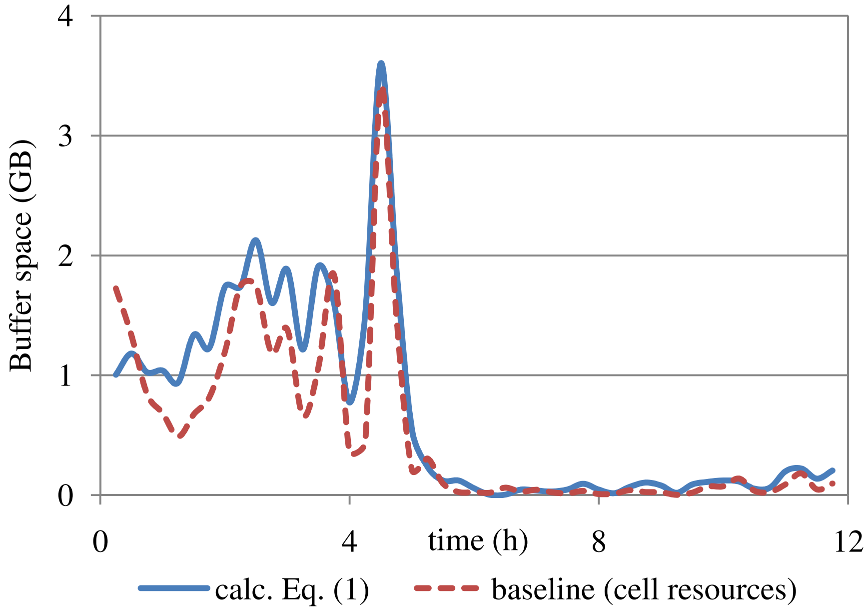


Figure 6.5 Available buffer space in the vicinity of a node, time- and space-varying scenario, disaster area

6.6 Exploiting resource heterogeneity

Subsection 6.3 suggests that every node can estimate the resource level available in its vicinity by keeping a small matrix M derived from the history of its previous encounters. These estimates were shown to be quite accurate when compared to the actual resources available in the vicinity. Basically, each node can exploit the history of its own previous n_τ encounters at each moment. In addition, whenever a reference node A meets a node B, information about the resources in B's vicinity, such as energy (E_B^v), buffer space (S_B^v) and bandwidth (D_B^v) becomes available to A.

The goal here is to show how this information can be potentially used to optimise the store-carry-forward scheme in intermittently connected networks. Since this strategy is based on an analysis of information available about three resources, the approach can be called holistic. Custodian election and message queue management policies are then proposed, as elements of strategy that can contribute to improving overall network performance. These policies can be viewed from two different perspectives: a *basic* perspective, where the number of custodians chosen cannot be defined and varies according to fluctuating system conditions, and a *controlled* perspective, where the number of custodians is determined as a fraction of the total number of encounters.

6.6.1 The basic perspective

The basic approach can be formalised as a set of custodian election (CE) and queue management (QM) policies, as shown below:

(CE) When a reference node A encounters a node B, node A elects node B as a custodian only if its relative strength (own resources versus vicinity resources) is above a specific threshold value. The whole set of resources, energy, buffer space, and bandwidth, are taken into account, as follows:

$$\text{Select node } B \text{ if:} \qquad \left(\frac{E_B^o}{E_B^v} > T_E\right) \wedge \left(\frac{S_B^o}{S_B^v} > T_S\right) \wedge \left(\frac{D_B^o}{D_B^v} > T_D\right) \tag{6.4}$$

where T_E, T_S, T_D are threshold values for energy, buffer space, and bandwidth respectively.

(QM) Once custodian B has been selected according to condition (CE) above, send message m of size s_m only if: (1) available energy at B is above the necessary level required for transmitting s_m bytes, (2) available buffer space at A exceeds s_m, and (3) available bandwidth, given a particular contact window, allows sending s_m bytes. Note that queue management policies can be described exclusively as a function of message size s_m and available resources at node B. In case one of these conditions does not hold for a message m, a smaller message should be sent to B.

Although Section 6.6.3 will demonstrate that combining queue management policies with the basic form of custodian election policies has significant benefits, it should not be overlooked that there are also two main disadvantages associated with this solution:

1) *No control over custodian election ratio*. There is no control over how many custodians are elected out of a (given) number of total meetings. This scheme will elect as many or as few custodians as required, according to the thresholds chosen and the resources available in the vicinity.

2) *Difficulty in setting threshold values*. The question remains open on how to set threshold values (T_E, T_S, T_D in Equation 6.4). Of course, these values can be set experimentally, based on a particular system. However, the system may decide at some point to change its initial behaviour. For instance, it suffices for only one of these threshold values to be set too low for a node to potentially defer choosing a custodian indefinitely. It is obvious, however, that such an indefinite deferral would not be a good option for the store-carry-forward process.

6.6.2 The controlled perspective

The novel idea presented in this section consists in limiting the number of custodians a reference node A elects to a specific *fraction* of the nodes it meets. In this approach, only the custodian election policies are modified, while the queue management policies are kept unchanged.

The controlled perspective can also be built on top of Epidemic routing, a typical case of greedy replication. However, since the number of custodians is restricted, replication in this case is no longer greedy. Intuitively, this is expected to result in lower resource usage, as controlled replication is usually seen as an effective means for reducing overhead and still achieving adequate performance. The objectives of this controlled perspective are the following:

1) To elect as custodians a fraction λ of nodes out of all node encounters.

2) To replicate the message only to custodians holding the best cumulative resources.

Let us recall from Section 6.3 that, during an observation time frame τ (sliding window), a reference node A will have approximately n_τ encounters with nodes from B_i to B_j ($n_\tau = j - i + 1$). Information on the available level of resources at nodes met is stored by node A in a matrix M_A. Provided that n_τ is big enough, it can also be considered that the resource distribution of the nodes encountered by node A will not vary significantly before and after a new node B_{j+1} is encountered. Before meeting node B_{j+1}, and based on M_A, node A calculates a resource threshold value, such that a given fraction λ of the n_τ nodes fall above this threshold value, while the rest $(1 - \lambda)$ stay below. When nodes meet, node B_{j+1} becomes a custodian for node A, provided that the own resources of B_{j+1} are above this calculated threshold value. The objective is to make sure that, over any given τ period, A will choose only $\lambda \times n_\tau$ custodians, λ being a protocol parameter, set in advance. Therefore, even though node A will meet potential custodian nodes one at a time, it will be able to calculate the selection criteria before each encounter, and also check them upon encounter with any custodian candidate.

The question remains on how to combine all relevant resources into one abstract and comparable value, thereby capturing our holistic approach to resources. For the generic case, let $u(E_B^o, S_B^o, D_B^o)$ be a function representing the utility[2] of choosing node B as a custodian for node A, where E_B^o, S_B^o, D_B^o are node B's own available levels of energy, storage space, and bandwidth respectively. The utility function represents the benefit an elector node A obtains from choosing a particular custodian B instead of some other. Of

[2]Utility is considered here only from a resource perspective; this approach does not take into account other criteria, such as proximity to destination, social impact, etc.

course, u should monotonically increase as a function of the variables E^o_B, S^o_B, D^o_B, thus indicating the fact that the choice of a node with better energy, buffer space, or bandwidth levels is more beneficial for node A. On the other hand, u should also take account of the holistic perspective, and provide adequate weight to the relative importance of energy, buffer space, and bandwidth for the system. For the performance evaluation presented in Section 6.6.3, a utility function has been used that: 1) gives equal weighting to energy, buffer space, and bandwidth, and 2) increases linearly for each of these resources:

$$u(E^o_B, S^o_B, D^o_B) = E^o_B \times S^o_B \times D^o_B \tag{6.5}$$

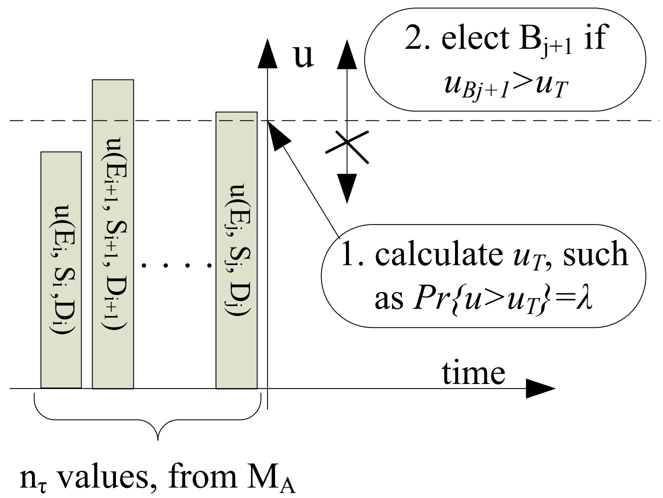


Figure 6.6 Custodian election in a controlled scenario

As shown in Figure 6.6, node A knows the last n_τ resource values corresponding to the last nodes met. Knowing function u and applying it for every node last encountered, A can determine the threshold value u_T such that, if the distribution of past resources were to be similar in the near future, approximately a fraction λ of all nodes would be elected. When A meets a potential custodian node B_{j+1}, A checks if $u_{B_{j+1}}$ is above the threshold value u_T. If it is, A will elect B_{j+1} as a custodian, otherwise it will not. Matrix M_A also provides the data structure needed to keep track of the nodes elected and ensures that a fraction close to λ is elected. As M_A is refreshed at every new encounter, and only a sliding window of size n_τ is kept, the system is capable of adapting dynamically to variable network conditions.

Note that the utility function may be more complex in different practical scenarios. For example, either 1) some resources may be given a greater weight than others, for instance nodes with energy resources may be preferred over nodes with buffer space, and/or 2) the function may vary in a nonlinear fashion, for instance utility may decrease more abruptly

Table 6.1 Simulation setup

Group	# nodes	Initial energy E_{in}(kJ)	Initial Buffer (MB)	Initiate messages
1	50	20	500	no
2	50	50	500	yes
3	50	20	50	no
Energy factor for sensing (e_s):			0.1J/s	
Energy factor for message exchange (e_m):			0.1J/kB	
Message size (fixed):			10MB	
Transmission speed:			0.1MB/s	
Simulation period:			12h	
Node's radio transmission range:			10m	
Playground size:			360m x 170m	

for resources close to depletion. Under such a scenario, at some point in time a node may value the availability of energy more (as it is aware that its battery is running down, and the next custodian may be its only hope to send a message on), while at some other point in time the same node may value the availability of bandwidth more (since it has a particularly heavy payload in its buffer).

6.6.3 Performance evaluation and discussion

With a view to establishing the end-to-end impact of the policies presented in Sections 6.6.1 and 6.6.2, these policies have been implemented on top of the Epidemic routing protocol [VB00]. Epidemic routing has been chosen due to its simplicity and the absence of any a priori queue management and custodian election policies, which makes differences easier to spot. However, it would also be possible to modify some other, potentially more complex, store-carry-forward protocols, as long as the original protocol policies can be combined with integrated information about resource availability.

Bonn motion mobility traces [AGPG+07] and the network configurations described in Table 6.1 above are used in this implementation. Messages originate from only one of the 50-node groups and are intended for all the 150 nodes in the setup, and they are injected into the system only during the first hour of simulation in order to create the preliminary heterogeneity that is then exploited using the proposed scheme.

The energy model implemented in this simulation corresponds to the theory presented in Section 6.4.2. Available energy at each node diminishes in time as an effect of 1) radio scanning, and 2) message transmission. While the former factor is proportional to elapsed

Table 6.2 Various curves in the simulation environment

id		description
E	=	Epidemic (used as baseline)
b:C	=	Epidemic + custodian election policy, as described in the *basic* perspective (Section 6.6.1). $T_E = T_S = T_B = 1.5$ has been used for this simulation
b:CQ	=	Epidemic + custodian election + queue management as described in the *basic* perspective (Section 6.6.1)
$c : \lambda$	=	Epidemic + custodian election policy, as described in the *controlled* version (6.6.2). It elects as custodians a λ fraction of nodes out of all nodes met. For instance, c:9 represents the case where only $\lambda = 9\%$ custodians out of all meetings are elected. λ is varied in a set of values of 9%, 15%, and 35%

time, the latter is proportional to message size, as described by Equation 6.2. Initial energy for one node is set between 20kJ and 50kJ, which corresponds roughly to the fully charged battery of a modern smartphone. Because batteries are not recharged during the 12h run, some nodes will be exhausted, and therefore become useless for transmitting messages.

Note that the energy related parameters in this evaluation, presented in Table 6.1, are not modelled based on measurements performed on a specific device. However, the parameters selected for this evaluation are corroborated by findings in some recent studies [PFS$^+$09]. Of course, device characteristics, the radio interface used, and data bursts will affect the depletion model. Yet, it has been found that running the same scenario with various radio and energy parameters in several runs largely reproduces the same qualitative results.

This simulation involves a comparison of several curves that have been described in detail in Table 6.2 above.

Figure 6.7 shows how nodes gradually become useless for the network as their energy is depleted over time. As a consequence of using the highest energy footprint, the Epidemic protocol quickly exhausts most nodes, and the simulation ends with only about 15% of the initial set of nodes. This figure also shows that implementing queue management and custodian election policies is a good way to reduce resource consumption (energy, in this particular case).

When comparing a set of store-carry-forward protocols applied to a scenario where nodes have various levels of available resources, the ideal protocol should satisfy the following intuitive requirements: 1) it should use a small resource footprint to deliver a large number of messages with low latency, and 2) during operation, it should attempt to level out disparities between left-over resources available to various nodes. While the first requirement is systematically taken into account in most related studies, the second requirement

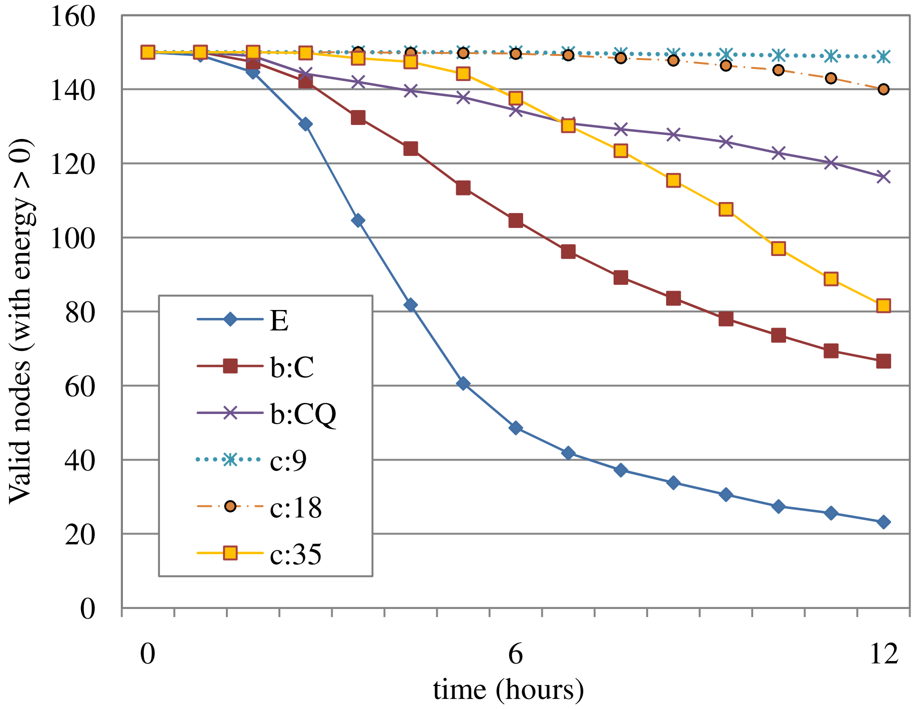


Figure 6.7 Valid and exhausted nodes over time

is somewhat more subtle. The suggestion would be that an ideal protocol should do its best to preserve as many nodes available in the network as possible. In other words, it should make careful use of nodes with scarce resources, and make preferential use of nodes with abundant resources, thereby maximising the number of valid nodes available in the network. On the other hand, the network may suffer if resources are concentrated only in a few nodes, while the remaining nodes undergo exhaustion as an effect of battery depletion, buffer exhaustion, or network congestion. Formally, in order to meet the second requirement, the distribution of available resources should be maintained as uniform as possible across the node population.

In order to achieve a better understanding of the various behaviours of the proposed schemes, a coefficient of variation (CV) on resources has also been considered. The case of available energy was analysed first, knowing that other resources, such as buffer space and bandwidth, can receive a similar treatment.

$$CV_E = \frac{\sqrt{\frac{1}{N}\sum_{1}^{N}(E_i - \overline{E})^2}}{\overline{E}} \tag{6.6}$$

where:
CV_E = coefficient of variation for energy
N = number of nodes (150 in this case)
E_i = energy of node i
$\overline{E}$ = energy mean value

Figure 6.8 plots the coefficient of variation for energy as a function of time, for all the six curves. This figure shows that custodian election policies, particularly in the controlled version, show smaller energy variance across the node population. Although not shown here, similar shapes can also be found when plotting the coefficient of variation for buffer space and bandwidth across the node population. This means that custodian election policies will use network resources especially from nodes with above average amounts of resources, and deal gently with nodes whose available assets are below average.

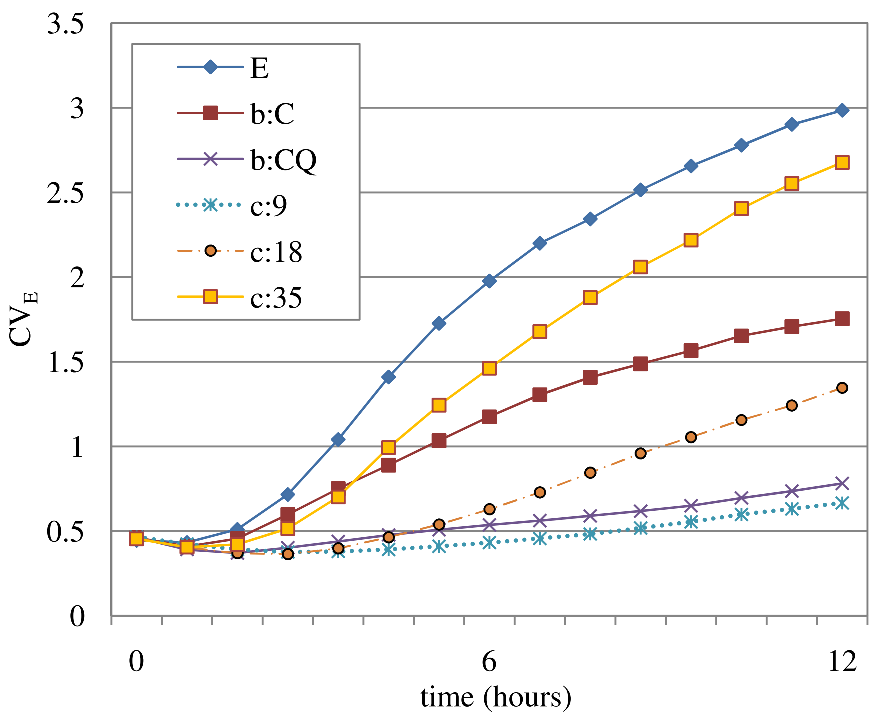


Figure 6.8 Variation coefficient for energy available to nodes

Based on Figures 6.7 and 6.8 it can be concluded that the proposed schemes not only use fewer resources, but also maintain a narrow variation of the resources available across the node population by avoiding node starvation as a consequence of resource exhaustion.

Finally, other network metrics also show significant improvements when queue management and custodian election schemes are used. In order to validate the approach at various loads, the number of messages sent was gradually increased by a factor of 1 to 8, and the results were plotted in Figure 6.9. The delivery ratio proves to be significantly better (around 30%) and there is no significant increase in latency (less than 3%) when best performing curves are used (c:9 and c:18).

A legitimate question concerns the costs incurred (i.e. added overhead) for information exchange between nodes. Costs can be approximated as follows:

- **network costs**. In order to implement custodian election and queue management policies, custodian candidates need to send E^o_A/E^v_A, S^o_A/S^v_A and D^o_A/D^v_A at each meeting. For calculating the resource level available in the vicinity (E^v_A, S^v_A and D^v_A), nodes need to exchange the vector $I = (\vec{v}, E^o, S^o)$ at each meeting. Considering that each scalar value is expressed by 2 bytes, and each vectorial value by 4 bytes, it can be considered that data exchanged by one node at each encounter is 14 bytes.

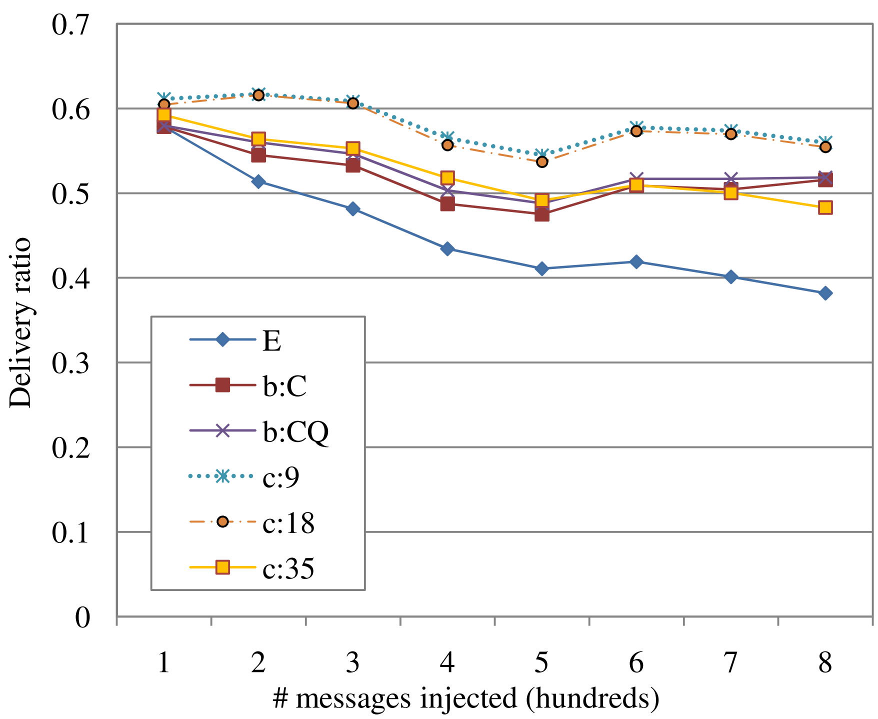


Figure 6.9 Delivery ratio on load

- **storage costs**. Each node stores a matrix sized $n_\tau \times size_I$. This translates into storage requirements of 320 bytes in the current setting.

These are simple metrics that can be easily compared with aggregated network workload (for all nodes and the whole duration of the experiment) and buffer size. In the settings considered, aggregated network workload was about 10^6 times above network costs and node average buffer was about 10^6 times above storage costs.

6.7 Summary

This chapter has shed some light on how nodes can estimate the amount of resources they may encounter in their vicinity autonomously and without central knowledge. The first step in the analysis was to select the resources relevant for a store-carry-forward scheme. A node needs some buffer space in order to be able to store a message, and it also needs energy to keep node power on and to feed energy into the communication interface. Finally, it also requires bandwidth to be able to forward messages to destination.

One of the motivations behind writing this chapter is the observation that, in real networks, participating nodes will differ significantly in terms of their levels of buffer space, energy, or bandwidth resources. Routing algorithms should make the most efficient use of these resources while making sure that those in limited supply are not wasted. These resources can be either represented as a generic abstraction, or they can be modelled as individual resources, with an emphasis on their specificities.

This chapter has shown that various resources: energy, bandwidth, and buffer space can be modelled as a generic resource. This resource abstraction paves the way for a holistic interpretation of resources, insisting on the fact that all three resources are required in order to allow a store-carry-forward routing protocol to operate successfully. Moreover, representing energy, bandwidth and buffer space as a generic resource facilitates the quantitative evaluation of these resources in the vicinity of a node. This chapter has proposed a distributed scheme that uses only past and current encounters in order to evaluate the resources available in the vicinity of a node. This evaluation takes into account the resources present in the vicinity of all nodes met, as well as the mobility parameters of those nodes, but it does not take account of any central knowledge. The validity of the proposed distributed scheme has been tested in simulations and a good accuracy of the values calculated has been demonstrated using the proposed scheme as compared to various theoretical or experimental baselines.

However, what is most important about the possibility of calculating the amount of resources available in a network is that the data thus obtained may benefit a generic routing protocol. A generic store-carry-forward protocol can be conceptually divided into two modules: a custodian election module, and a queue management module. Custodian election is responsible for selecting one or several nodes from the list of available potential custodians that are considered dependable enough to carry messages towards their destination. On the other hand, queue management is responsible for choosing the messages to be forwarded and the order in which they should be transmitted to a custodian or even deleted in case storage buffer runs short.

By letting the information on available resources interfere with custodian election and queue management policies, an optimisation approach has been proposed that can accommodate any store-carry-forward protocol. Two types of custodian election policies have been proposed. The first uses the relative strength of a node's own resources as compared to the resources found in its vicinity as a basis for choosing a particular node as a custodian. The second uses past encounters and allows a node to select only a given fraction of nodes as custodians out of the total number of encounters of that particular node. Assuming this fraction is a protocol parameter, this policy proposal can be seen as a version of controlled replication. Moreover, this controlled replication is also resource-aware, as it consists in selecting only the best-fitted custodians in the vicinity of a node, i.e. those that have the highest level of resources available.

It is shown in this chapter that, by implemening these policies, a routing protocol may not only use up fewer resources overall, but may also consume resources preferentially from nodes with higher resource levels, sparing those with limited supplies when possible. As a result, disparities in available resources across the node population can be significantly reduced, and nodes are less likely to leave the network as a consequence of resource depletion. The solutions proposed are particularly beneficial in networks with heterogeneously distributed resources.

These schemes do not take account of one resource only, but rather three: available storage space, energy, and bandwidth. Therefore this approach has been called holistic, the availability of these resources being seen as a whole. Simulations have demonstrated that the benefits of these policies are substantial when combined with an Epidemic routing baseline in a disaster management scenario. Performance evaluations have shown that resource-aware custodian election and queue management policies can yield significant benefits without any noticeable disadvantage. Network costs have also been analysed and shown to be relatively low as compared to the benefits.

Most improvements have been found to be the result of applying custodian election policies, according to which a custodian is chosen based on the result of a comparison between a node's own available energy, buffer space, and bandwidth, and the values of the same parameters found in its vicinity. A threshold value has been used as a scaling factor to compare own resources with resources available in the neighbourhood. Queue management policies have a more subtle role to play in these improvements, as they affect message latency in particular, and are more apparent at higher loads.

Chapter 7

Conclusion and future work

This chapter summarises the insights and findings of this thesis, and outlines potential directions for further research on this topic.

7.1 Conclusion

Delay and disruption-tolerant networking (DTN) has been a hot topic of interest over the last decade and has emerged as a potential solution to extending current Internet architecture to support partitioned networks. A full end-to-end path from source node to destination node may never exist in such networks. Moreover, links between nodes work intermittently, and may have highly asymmetric data rates, large propagation delays and/or high error rates. Routing in this type of environment is a challenging undertaking, as messages have to be delivered by nodes possessing extremely limited advance information. Routing algorithms make use of a store-carry-forward paradigm in order to route messages towards their destination. In most typical scenarios, nodes have limited amounts of resources available. Therefore routing algorithms should make the most efficient use of these resources while making sure that those in limited supply are not wasted.

This thesis has proposed several contributions to DTN resource-aware routing, addressing the following issues in particular:

- *Utility-aware queue management schemes:* Understanding how operations on messages, such as message replication and message deletion, can be performed using utility functions and message differentiation. The goal has been to optimise the accrued global utility calculated for messages delivered, and the usage of resources, such as buffer space.

- *Variable contact window as an effect of mobility:* Modelling the effects of mobility on link duration and proposing strategies to deploy when message size exceeds the largest transferable unit over the expected contact window. The goal has been to adapt the sending process to the contact windows available in order to avoid partial transmissions or, alternatively, to divide messages into smaller chunks that can be sent over a given contact opportunity.

- *Hybrid DTN/infrastructure schemes:* Proposing a cross-network hybrid scheme where most of the traffic is routed via a DTN network, while some last bits may be pulled using residual infrastructure. The goal has been to minimise infrastructure usage and maximise overall delivery rate.

- *Combined evaluation of erasure coding and replication:* Understanding how these considerably different but redundant mechanisms of data aggregation can be used in parallel, in a resource-conscious way. Optimisation has aimed at using a minimum amount of resources in order to obtain a particular network latency.

- *Mapping available resources in the vicinity:* Defining a scheme to gather information on the level of resources available in the vicinity of a node by using only information from current and past encounters. All three relevant resources, i.e. energy, bandwidth, and buffer space were considered while validation has been done by focusing on buffer space.

- *Custodian election strategies based on resource availability:* Selection of a custodian node for a message, based on the resources available at node level, as compared to the resources found in node vicinity. This strategy adopts a holistic view of various resources and aims at preserving available resources at each node by consuming predominantly only those available in abundance.

Chapter 3 of this thesis has presented a new **O**pportunistic **R**outing algorithm using **W**indow-aware **A**daptive **R**eplication (ORWAR). This routing protocol exploits the store-carry-forward mechanism in delay-tolerant networks and combines controlled replication and delivery acknowledgements with two novel features in the DTN context: (1) utility-aware replication by using message utility as a parameter in the selection of replicated messages, followed by deletion from the message queue, and (2) contact window estimation aimed at optimising the number and size of messages sent in relation to the estimated contact opportunity. This routing scheme is shown to outperform a number of well-known routing protocols such as SprayAndWait [SPR05], Maxprop [BGJL06] and PROPHET [LDS04], particularly when messages to be sent are relatively big in size and may exceed a given transmission opportunity. The use of message differentiation helps

ORWAR outperform other routing schemes especially when networks are subject to message overload. The adaptive character of ORWAR makes it well-suited for a wide range of message sizes, including big-sized messages that would have generated a large number of partial transmissions in case an ordinary routing protocol, with no contact-window estimation, had been used for their transfer. It should be said, however, that for the time being only high level simulations have been performed for the validation of results (using ONE [KOK09] simulator). On the other hand, some effort is required in order to extend the wireless model and move on to a more realistic approach, discarding the simplistic assumption that radio ranges are perfect circles characterised by contiguous communication opportunities and constant bandwidth.

Another method for coping with big-sized messages to be transferred over limited contact windows would be the fragmentation of original messages into smaller data units. Chapter 4 has analysed the effects of adding erasure coding (fragmentation and redundancy) to the ORWAR protocol, and has demonstrated an improvement in the performance of the original protocol. The chapter has also included an analysis of the relation between fragment size and mobility, and has presented an experimental evaluation of fragmentation size, thus confirming the theory.

Fragmentation has been shown to be useful also in relation to hybrid DTN/Infrastructure schemes. The purpose of such a hybrid scheme would be to route most of the traffic over the DTN network, and to pull the last few fragments via the expensive but more reliable infrastructure network. The advantages of this hybrid scheme may be considered from two perspectives. First, store-carry-forward schemes have generally been presented as a better-than-nothing solution when all else fails, that can only achieve low delivery ratios. However, users usually expect all the data units they send to be delivered from source to destination. In order to do this, the proposed hybrid DTN/Infrastructure scheme uses fragmentation as an instrument to minimise the load on the infrastructure, then transfers most messages over the DTN network, whose costs are lower, and finally pulls the missing bits over the infrastructure. As a result, all data units are eventually delivered, using one network or the other, with a net preference for the DTN network. Second, even if cost were not an issue, sending all the traffic over the infrastructure network would not be a feasible solution either. For example, when communicating in a disaster area, the attempt to handle all data over the infrastructure network would lead to unsustainable overloading and communication failure. An intelligent split between a cost-conscious DTN network and the more reliable infrastructure may not only be more cost-effective, but may also improve network survivability.

Chapter 5 has formulated a proposal for an optimisation framework where both controlled replication and erasure coding are used within a store-carry-forward scheme. The purpose of the analysis has been to demonstrate how these different, but somewhat redundant mechanisms of data aggregation can be used in parallel, thus leading to a minimisation of resource usage. Out of the various optimisation criteria that could apply, optimisation of buffer space has been chosen for the demonstration contained in this chapter. By applying minimisation criteria on delivery costs, Chapter 5 has shown how optimised replication and redundancy factors can be calculated for each message, also taking into account message size and time-to-live. The analysis performed has led to the following insights: (1) given a latency target, it is possible to calculate an appropriate level of replication and a correlated level of redundancy that represents the best (minimum) network cost for a given resource, in particular for storage space, (2) fluctuation in available bandwidth at nodes translates into significant variations of best cost, corresponding to a particular value of available bandwidth, which is usually neither at the bottom nor at the top of the range, and (3) overfragmentation of messages - using much smaller fragment sizes than those required by a typical contact window - negatively affects cost by ineffectively multiplying delivery paths. The model has then been enhanced from being dependent on the latency bounds calculated for a given protocol (SprayAndWait) to a generic model that can adapt itself based on delivery history. In case the destination node keeps a record of delivery times and occasionally sends it to the source, this information can be used as an input for the optimisation process. Because this mechanism is supposed to remain valid for an arbitrary routing protocol that uses replication and erasure coding, where neither upper and lower bounds, nor distribution of delivery times can be analytically determined, it represents an improvement, as it removes some of the model's simplifying assumptions. However, this is still to be demonstrated by means of real-life implementations or at least by extended simulations.

Chapter 6 has shown that calculating the amount of resources available in the vicinity of a node and using this information for routing decisions may substantially increase routing performance by preventing resource exhaustion. This strategy is beneficial especially for heterogeneous environments, that contain various types of nodes, or where message injection is unbalanced. For the first time, this chapter has brought together various resources - energy, bandwidth, and buffer space - presenting them as a generic, abstract resource. This abstraction calls for a holistic interpretation of resources, underlining that all three resources need to be available in order to allow a store-carry-forward routing protocol to work successfully. Moreover, representing energy, bandwidth, and buffer space as a generic resource facilitates the quantitative evaluation of these resources. This evaluation does not need any central knowledge and relies exclusively on an exchange of information about resources or mobility parameters that nodes can gather autonomously. The validity of the proposed distributed scheme has been tested in simulations and a good accuracy of the calculated values has been demonstrated, using both theoretical and experimental baselines. By letting the information on available resources interfere with the custodian election and queue management policies, an optimisation approach has been proposed that can accommodate any store-carry-forward protocol. This approach uses the relative strength of a node's own resources as compared to the resources found in its vicinity as a basis for choosing a particular node as a custodian. As mentioned earlier, this bidding scheme does not take into account one resource only, but rather three: available storage space, energy, and bandwidth. Given that the availability of these resources has been considered as a whole, this approach has been appropriately called holistic. Simulations have demonstrated that the benefits of these policies are substantial when combined with an Epidemic routing baseline in an emulated disaster management scenario. The cost of implementing the proposed distributed scheme between nodes has been estimated to be low, i.e. a fraction of $10^{-}6$ from the network capacity and total buffer availability respectively. However, neither computational costs, nor the lower networking layers have been taken into account, as the modelling process has been kept at a relatively high level of abstraction.

7.2 Future work

The evaluation of ORWAR performance presented in this thesis has been based on simulations using the ONE network simulator [KOK09]. This is one of the very few network simulators built specifically for delay-tolerant networks, therefore it constitutes an abstraction class from protocols below the bundle layer. While this assumption is essentially correct, and highlights the fact that DTNs should work with a multitude of lower level protocols, even over exotic links such as those found in outer space or underwater scenarios, a legitimate question would be: what would performance be if ordinary protocols, such as IP over IEEE 802.11 links, were used? In order to shed light on this question, Frederik Herbertsson has studied this protocol in his master thesis [Her10] using a *ns-3* network simulator [NS310] which implements UDP/IP over IEEE 802.11 links. Herbertsson finds performance measurements overall in line with those presented in Chapter 3 of this thesis, but he points out that, because topology may change in an unpredictable way, beaconing mechanisms are needed for discovering potential neighbours. As a result, beacon rates would also have to be optimised, knowing that higher rates consume resources, and lower rates miss contact opportunities. Other practical mechanisms at lower level should also be implemented, such as avoiding checks of the neighbours' connectivity using beacons in case data has been received at one node. Again, this is not a DTN issue, but should be taken into account in using a cross-layer design in a real-life implementation. Cross-layer design may therefore be a future line of work.

ORWAR has also been the subject of another master thesis [Anz10] by Davide Anzaldi, who has implemented this protocol for the Android platform [And11]. In that work only the ORWAR version without fragmentation has been implemented. In his thesis, Anzaldi points out that, in real-life cases, radio ranges may not be circular and obstacles may cause significant deviations from the theoretical assumptions. This calls for more empirical studies to check to what extent the range of real measurements of contact windows differ from calculated ones. Similary, the bandwidth available between nodes may not be constant over the whole contact window, so this may also need to be reconsidered in a physical test bed. Actually ORWAR has been tested on Android devices using only a very limited number of nodes (more exactly two), and that for practical reasons. Therefore testing ORWAR in a real-life implementation, with a significant number of nodes and relevant mobility, would be another exciting direction of future work.

There are also other potential directions for future work that are closer to the ORWAR theoretical framework. For example, it would be interesting to know how message time-to-live may play a role in the utility function of a message and thus take active part in the utility-aware replication process. For instance, considering that delivery of a message with a closer deadline would increase accrued utility, it may be possible for a node to increase the replication rate accordingly. However, this would be a move out of the controlled replication domain, because no maximum total number of copies spread could then be determined. This may also call for a new optimisation framework between replication and erasure coding, as controlled replication would no longer hold.

Another enhancement could be done by considering a time-to-live for vaccine messages (messages known to be delivered), which would minimise space taken up in node buffers or would reduce the number of overall transmissions involving vaccine messages. Some other enhancements may also concern the fragmentation framework. Chapter 3 and 4 have proposed a network-wide constant fragment size calculated by taking into account maximum node speed. This framework can be enhanced by transferring calculations to the nodes themselves, using exclusively distributed mechanisms and no global knowledge. In fact, fragment size may not even be constant for a whole network, but rather it may vary for various node pairs (source, destination). Piggy-backing vaccine messages would be a practical and resource-friendly method to add more information about the mobility parameters between source and destination.

The optimisation framework involving replication and erasure coding may be enhanced in an interesting way by replacing the optimisation criteria used. While the current criterion is total buffer space available in the network, the new criteria could be minimising the number of transmissions or total bandwidth used. The replication/erasure coding framework may also have a different application than routing, more concretely for the study of privacy in a delay-tolerant network. Splitting the initial bundle into fragments and then having each fragment take a different path also means that the initial data avoids eavesdropping for the majority of intermediary nodes. Only a minority of intermediary nodes, those that have got a sufficient number of fragments in order to reconstruct the initial bundle, can break data privacy. This would be another idea for further work that might lead off to a different research domain.

Finally, vicinity resource cartography is a domain that could be further studied in a number of ways. First of all, it would be possible to study the optimisation of the sliding time window during which information about the nodes met during transfer is kept in memory. This size has been empirically set in Chapter 6, but it is very likely that window size could affect the accuracy of measurements. Moreover, the calculation of bandwidth in the vicinity of a node has been done assuming that nodes would meet only in pairs (this assumption corresponds to a very sparse network). It would be interesting to know how an encounter between more than two nodes, a situation that may occur in a denser network, would affect bandwidth calculation. Another direction of study would be the propagation in space and time of an inconvenient event affecting resources, such as the filling up of node buffers, or energy exhaustion, upon massive message injection. Such a study could elaborate some viable resource allocation strategies, similar to flow control and congestion control in fully-connected networks.

Bibliography

[ABP10] E. Altman, T. Başar, and F. D. Pellegrini, "Optimal monotone forwarding policies in delay tolerant mobile ad-hoc networks," *Elsevier Performance Evaluation*, vol. 67, no. 4, pp. 299–317, April 2010. Cited on pages: 74, 81, and 87.

[ABR08] A. Andronache, M. Brust, and S. Rothkugel, "HyMN-injection-based multimedia content distribution in hybrid wireless networks," *Springer Multimedia Systems Journal*, vol. 14, no. 3, pp. 145–153, August 2008. Cited on page: 59.

[AGPG+07] N. Aschenbruck, E. Gerhards-Padilla, M. Gerharz, M. Frank, and P. Martini, "Modelling Mobility in Disaster Area Scenarios," in *Proceedings of the* 10^{th} *International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. ACM, October 2007, pp. 4–12. Cited on pages: 22, 92, 103, and 108.

[And11] "Android Inc," 2011. [Online]. Available: http://developer.android.com Cited on page: 122.

[Anz10] D. Anzaldi, "ORWAR: a Delay-Tolerant Protocol Implemened on the Android platform," Master's thesis, Linköping University, December 2010. Cited on pages: 55 and 122.

[APM04] I. F. Akyildiz, D. Pompili, and T. Melodia, "Challenges for efficient communication in underwater acoustic sensor networks," *ACM SIGBED Review - Special issue on embedded sensor networks and wireless computing*, vol. 1, no. 2, pp. 3–8, July 2004. Cited on page: 14.

[ASL08] A. Agarwal, D. Starobinski, and T. D. C. Little, "Analytical Model for Message Propagation in Vehicular Ad Hoc Networks,," in *Proceedings of the IEEE Vehicular Technology Conference (VTC-Spring '08)*. IEEE, May 2008, pp. 3067–3071. Cited on page: 73.

[Aus01] *A Review of Automotive Radar Systems - Devices and Regulatory Frameworks*, Australian Communication Authority, 2001. [Online]. Available: http://acma.gov.au/webwr/radcomm/frequency_planning/spps/0104spp.pdf Cited on page: 39.

[BCF07] S. Bandyopadhyay, E. J. Coyle, and T. Falck, "Stochastic Properties of Mobility Models in Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 11, pp. 1218–1229, November 2007. Cited on page: 76.

[Bet01] C. Bettstetter, "Mobility modeling in wireless networks: Categorization, smooth movement, and border effects," *ACM Mobile Computing and Communications Review*, vol. 5, no. 3, pp. 55–66, July 2001. Cited on page: 22.

[BG10] P. Bellavista and C. Giannelli, "Internet Connectivity Sharing in Multipath Spontaneous Networks: Comparing and Integrating Network- and Application-Layer Approaches," in *Proceedings of the* 3^{rd} *International ICST Conference on MOBILe Wireless MiddleWARE (MOBILWARE'10), Operating Systems, and Applications*, June 2010. Cited on page: 59.

[BGJL06] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for Vehicle-Based Disruption- Tolerant Networks," in *Proceedings of the* 25^{th} *IEEE International Conference on Computer Communications (InfoCom'06)*, April 2006, pp. 1–11. Cited on pages: 19, 36, 45, 54, and 118.

[BHT$^+$03] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking: an approach to interplanetary Internet," *IEEE Communications Magazine*, vol. 41, no. 6, pp. 128–136, June 2003. Cited on page: 13.

[BJS06] S. Burleigh, E. Jennings, and J. Schoolcraft, "Autonomous Congestion Control for an Interplanetary Internet," in *Proceedings of the AIAA SpaceOps Conference 2006*, June 2006. Cited on pages: 3 and 15.

[BLM99] J. W. Byers, M. Luby, and M. Mitzenmacher, "Accessing Multiple Mirror Sites in Parallel: Using Tornado Codes to Speed Up Downloads," in *Proceedings of the* 18^{th} *IEEE International Conference on Computer Communications (InfoCom'99)*, August 1999, pp. 275–283. Cited on pages: 19 and 60.

[BLMR98] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 4, pp. 56–67, 1998. Cited on page: 58.

[BLV07] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "DTN Routing as a Resource Allocation Problem," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 373–384, August 2007. Cited on pages: 19, 21, 29, 36, and 38.

[BMJ+98] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the 4^{th} annual ACM/IEEE international conference on Mobile computing and networking (MobiCom'98)*, 1998. Cited on page: 11.

[CBH+07] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss., "Delay-tolerant networking architecture (RFC4838)," April 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4838.txt Cited on pages: 12, 14, 28, and 34.

[CE08] H. Cai and D. Y. Eun, "Toward Stochastic Anatomy of Inter-meeting Time Distribution under General Mobility Models," in *Proceedings of the 9^{th} ACM international symposium on Mobile ad hoc networking and computing (MobiHoc'08*, May 2008, pp. 273–282. Cited on page: 23.

[CHC+05] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Pocket Switched Networks: Real-world mobility and its consequences for opportunistic forwarding," University of Cambridge, Computer Laboratory, Tech. Rep. UCAM-CL-TR-617, February 2005. [Online]. Available: http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-617.pdf Cited on page: 23.

[CLF06] V. Conan, J. Leguay, and T. Friedman, "The heterogeneity of inter-contact time distributions: its importance for routing in delay tolerant networks," Arxiv, Tech. Rep. cs.NI/0609068, October 2006. Cited on page: 93.

[CNT08] C. Curescu and S. Nadjm-Tehrani, "A Bidding Algorithm for Optimised Utility-based Resource Allocation in Ad hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 12, pp. 1397–1414, December 2008. Cited on pages: 29 and 36.

[DH07] E. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proceedings of the 8^{th} ACM international symposium on Mobile ad hoc networking and computing (MobiHoc'07)*, September 2007. Cited on page: 92.

[DH10] E. Daly and M. Haahr, "The challenges of disconnected delay-tolerant MANETs," *Elsevier Ad Hoc Networks Journal*, vol. 8, no. 2, pp. 241–250, August 2010. Cited on pages: 3 and 12.

[DJ10] R. J. D'Souza and J. Jose, "Routing approaches in delay tolerant networks: A survey," *International Journal of Computer Applications*, vol. 1, no. 17, pp. 8–14, February 2010, foundation of Computer Science. Cited on page: 24.

[DTN03] "Delay-Tolerant Networking Research Group (DTNRG)," 2003. [Online]. Available: http://www.dtnrg.org Cited on pages: 9 and 12.

[DTN07] "Delay-Tolerant Networking Research Group. DTN reference implementation, Version 2.5," 2007. [Online]. Available: http://www.dtnrg.org/docs/code Cited on page: 35.

[DUP02] A. Doria, M. Uden, and D. P. Pandey, "Providing connectivity to the Saami nomadic community," in *Proceedings of the Developement by Design Conference 2002*, IEEE, Ed., June 2002. Cited on page: 24.

[EKKO08] F. Ekman, Keränen, J. Karvo, and J. Ott, "Working Day Movement Model," in *Proceedings of the* 1^{st} *ACM SIGMOBILE Workshop on Mobility Models for Networking Research*, May 2008. Cited on pages: 22 and 23.

[Fal03] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," in *Proceedings of the ACM conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM'03)*, August 2003. Cited on pages: 10, 11, and 14.

[FC06] S. Farrell and V. Cahill, *Delay and Disruption Tolerant Networking*. Artech House, October 2006. Cited on page: 20.

[FF08] K. Fall and S. Farrell, "DTN: an architectural retrospective," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 828–836, June 2008. Cited on pages: 12, 18, and 21.

[FM08] R. Fracchia and M. Meo, "Analysis and Design of Warning Delivery Service in Intervehicular Networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 7, pp. 832–845, August 2008. Cited on page: 73.

[GNK05] R. Groenvelt, P. Nain, and G. Koole, "The message delay in mobile ad hoc networks," *Elsevier Journal of Performance Evaluation*, vol. 62, no. 1–4, pp. 210–228, October 2005. Cited on pages: 24, 74, and 76.

[GT02] M. Grossglauser and D. N. C. Tse, "Mobility Increases the Capacity of Ad Hoc Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477–486, August 2002. Cited on page: 17.

[HCS+05] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket Switched Networks and Human Mobility in Conference Environments," in *Proceedings of the ACM SIGCOMM workshop on Delay-tolerant networking (WDTN'05)*. ACM, September 2005, pp. 244–251. Cited on pages: 12 and 59.

[Her10] F. Herbertsson, "ORWAR: a Delay-Tolerant Protocol Implemened on the ns-3 platform," Master's thesis, Linköping University, December 2010. Cited on pages: 54 and 121.

[HLV06] E. Hyytiä, P. Lassila, and J. Virtamo, "Spatial node distribution of the random waypoint mobility model with applications," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 680–694, June 2006. Cited on pages: 22, 100, and 101.

[HS06] Z. J. Haas and T. Small, "A New Networking Model for Biological Applications of Ad Hoc Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 14, no. 1, pp. 27–40, February 2006. Cited on page: 73.

[HSL10] T. Hossmann, T. Spyropoulos, and F. Legendre, "Know Thy Neighbor: Towards Optimal Mapping of Contacts to Social Graphs for DTN Routing," in *Proceedings of the* 29^{th} *IEEE International Conference on Computer Communications (InfoCom'10)*, March 2010. Cited on page: 92.

[IMC10] S. Ioannidis, L. Massoulié, and A. Chaintreau, "Distributed Caching over Heterogeneous Mobile Networks," in *Proceedings of the ACM international conference on Measurement and modeling of computer systems (SIGMETRICS'10)*, June 2010. Cited on page: 94.

[JDPF05] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using Redundancy to Cope with Failures in a Delay Tolerant Network," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 109–120, October 2005. Cited on pages: 19 and 58.

[JFP04] S. Jain, K. Fall, and R. Patra, "Routing in a Delay Tolerant Network," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 145–158, August 2004. Cited on pages: 11 and 21.

[JHF$^+$08] M. Johnsson, J. Huusko, T. Frantti, F.-U. Andersen, T.-M.-T. Nguyen, and M. P. de Leon, "Towards a New Architectural Framework The Nth Stratum Concept," in *Proceedings of* 4^{th} *International Mobile Multimedia Communications Conference*, July 2008. Cited on page: 13.

[JLSW07] E. Jones, L. Lily, J. K. Schmidke, and P. Ward, "Practical routing in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 8, pp. 943–959, August 2007. Cited on pages: 20, 21, and 35.

[KK98] R. Kravets and P. Krishnan, "Power Management Techniques for Mobile Communication," in *Proceedings of* 4^{th} *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98)*, 1998, pp. 157–168. Cited on page: 35.

[KNG$^+$04] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental evaluation of wireless simulation assumptions," in *Proceedings of the* 7^{th} *ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems (MSWiM'04)*, 2004, pp. 78–82. Cited on page: 41.

[KNT08] E. Kuiper and S. Nadjm-Tehrani, “Geographical Routing in Intermittently Connected Ad Hoc Networks,” in *Proceedings of the first IEEE International Workshop on Opportunistic Networking*, March 2008, pp. 1–11. Cited on pages: 18 and 30.

[KOK09] A. Keränen, J. Ott, and T. Kärkkäinen, “The ONE Simulator for DTN Protocol Evaluation,” in *Proceedings of the* 2^{nd} *International Conference on Simulation Tools and Techniques (SIMUTools’09)*. ICST, 2009. Cited on pages: 45, 54, 65, 78, 82, 101, 119, and 121.

[LD07] A. Lindgren and A. Doria, “Probabilistic Routing Protocol for Intermittently Connected Networks,” April 2007. [Online]. Available: http://tools.ietf.org/id/draft-lindgren-dtnrg-prophet-03.txt Cited on page: 24.

[LDS04] A. Lindgren, A. Doria, and O. Schelén, “Probabilistic routing in intermittently connected networks,” *Lecture Notes in Computer Science*, vol. 3126, pp. 239–254, January 2004. Cited on pages: 24, 25, 30, 37, 45, 46, 54, 73, and 118.

[Leg07] J. Leguay, “Heterogeneity and Routing in Delay Tolerant Networks,” Ph.D. dissertation, Pierre & Marie Curie University (Paris VI), 2007. Cited on page: 93.

[LFC06] J. Leguay, T. Friedman, and V. Conan, “Evaluating Mobility Pattern Space Routing for DTNs,” in *Proceedings of the* 25^{th} *IEEE International Conference on Computer Communications (InfoCom’06)*, April 2006. Cited on pages: 25 and 94.

[LLL08] Y. Lin, B. Li, and B. Liang, “Efficient Network Coded Data Transmissions in Disruption Tolerant Networks,” in *Proceedings of the* 27^{th} *IEEE International Conference on Computer Communications (InfoCom’08)*, April 2008, pp. 1508–1516. Cited on page: 74.

[LPO10] J. Lakkakorpi, M. Pitkänen, and J. Ott, “Adaptive Routing in Mobile Opportunistic Networks,” in *Proceedings of the* 13^{th} *ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems (MSWiM’10)*, Bodrum, Turkey, 2010. Cited on page: 60.

[LZYZ07] W. Li, Y. Zhang, J. Yang, and J. Zheng, “UCC: Update-Conscious Compilation for Energy Efficiency in Wireless Sensor Networks,” in *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI’07)*, June 2007. Cited on page: 2.

[Muk06] O. Mukhtar, “Design and Implementation of Bundle Protocol Stack for Delay-Tolerant Networking,” Master’s thesis, Helsinki University of Technology, August 2006. Cited on page: 13.

[NGP07] H. Nguyen, S. Giordano, and A. Puiatti, "Probabilistic Routing Protocol for Intermittently Connected Mobile Ad hoc Networks (PROPICMAN)," in *Proceedings of the* 1^{st} *IEEE Workshop on Autonomic and Opportunistic Communications (WoWMoM'07)*, Helsinki, 2007. Cited on pages: 25 and 94.

[NN03] D. Niculescu and B. Nath, "Trajectory Based Forwarding and Its Applications," in *Proceedings of* 9^{th} *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'03)*, September 2003. Cited on page: 18.

[NS310] "The Network Simulator, Version 3," 2010. [Online]. Available: http://www.nsnam.org Cited on pages: 54 and 121.

[OE08] H. Ochiai and H. Esaki, "Mobility Entropy and Message Routing in Community-Structured Delay Tolerant Networks," in *Proceedings of the* 4^{th} *ACM Asian Engineering Conference (AINTEC'08)*, November 2008. Cited on page: 18.

[OK05] J. Ott and D. Kutscher, "A disconnection-tolerant transport for drive-thru Internet environments," in *Proceedings of* 24^{th} *IEEE International Conference on Computer Communications (InfoCom'05)*, vol. 3, March 2005, pp. 1849–1862. Cited on page: 28.

[Pal10] J. Palmer, "Supercomputers 'will fit in a sugar cube', IBM says," November 2010. [Online]. Available: http://www.bbc.co.uk/news/technology-11734909 Cited on page: 2.

[PBGL08] A. Piyatumrong, P. Bouvry, F. Guinand, and K. Lavangnananda, "Trusted spanning tree for delay tolerant manets," in *Proceedings of IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC'08)*, vol. 2, December 2008, pp. 293–299. Cited on page: 22.

[PBGL09] A. Piyatumrong, P. Bouvry, F. Guinand, and K. Lavangnananda, "A Study of Token Traversal Strategies on Tree-Based Backbones for Mobile Ad Hoc - Delay Tolerant Networks," in *Proceedings of the International Conference on Ultra Modern Telecommunications (ICUMT'09)*, 2009, pp. 1–8. Cited on page: 22.

[PFS$^+$09] G. P. Perrucci, F. H. Fitzek, G. Sasso, W. Kellerer, and J. Widmer, "On the Impact of 2G and 3G Network Usage for Mobile Phones' Battery Life," in *Proceedings of the* 15^{th} *European Wireless Conference (EW'09)*. IEEE, November 2009, pp. 255–259. Cited on page: 109.

[PKO08] M. Pitkänen, A. Keränen, and J. Ott, "Message Fragmentation in Opportunistic DTNs," in *Proceedings of International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'08)*, June 2008. Cited on pages: 18 and 73.

[Pla97] J. S. Plank, "A tutorial on Reed-Solomon coding for fault-tolerance in RAID-like systems," *Software Practice & Experience, John Wiley & Sons*, vol. 27, no. 9, pp. 995–1012, September 1997. Cited on pages: 19 and 60.

[PO07] M. Pitkänen and J. Ott, "Redundancy and Distributed Caching in Mobile DTNs," in *Proceedings of the* 2^{nd} *ACM/IEEE international workshop on Mobility in the evolving Internet architecture (MobiArc'07)*, August 2007. Cited on page: 58.

[PR99] C. E. Perkins and E. M. Royer, "Ad hoc On-Demand Distance Vector Routing," in *Proceedings of the* 2^{nd} *IEEE Workshop on Mobile Computing System and Applications*. IEEE, February 1999, pp. 90–100. Cited on page: 11.

[PSDG09] M. Piórkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "A Parsimonious Model of Mobile Partitioned Networks with Clustering," in *Proceedings of the International Conference on Communication Systems and Networks (COMSNETS'09)*. IEEE, January 2009. Cited on pages: 23 and 92.

[Rai11] R. N. B. Rais, "Communication mechanisms for message delivery in heterogeneous networks prone to episodic connectivity," Ph.D. dissertation, University of Nice, Sophia Antipolis, France, 2011. Cited on page: 93.

[RH10] A. Rice and S. Hay, "Decomposing power measurements for mobile devices," in *Proceedings of the* 8^{th} *IEEE Conference on Pervasive Computing and Communications (PerCom'10)*, April 2010. Cited on page: 97.

[SHCD06] J. Scott, P. Hui, J. Crowcroft, and C. Diot, "Haggle: A networking architecture designed around mobile users," in *Proceedings of the* 3^{rd} *Conference on Wireless On-demand Network Systems and Services (WONS'06)*. IEEE, January 2006. Cited on page: 59.

[SNT08] G. Sandulescu and S. Nadjm-Tehrani, "Opportunistic Routing with Window-aware Adaptive Replication," in *Proceedings of the* 4^{th} *Asian Conference on Internet Engineering (AINTEC'08)*. ACM, November 2008, pp. 103–112. Cited on page: 54.

[SPR05] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking (WTDN'05)*, August 2005, pp. 252–259. Cited on pages: 17, 19, 36, 37, 39, 45, 54, 65, 73, and 118.

[SPR07] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility," in *Proceedings of the* 5^{th} *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom'07)* , March 2007, pp. 79–85. Cited on pages: 29, 36, and 73.

[SPR08a] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 77–90, February 2008. Cited on pages: 17, 18, 24, 72, and 76.

[SPR08b] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The single-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 63–76, February 2008. Cited on pages: 18, 24, 72, and 76.

[SRT+10] T. Spyropoulos, R. N. B. Rais, T. Turletti, K. Obraczka, and A. Vasilakos, "Routing for disruption tolerant networks: taxonomy and design," *Wireless Networks*, vol. 16, no. 8, pp. 2349–2370, Septembre 2010. Cited on page: 15.

[Sta02] T. Starner, "Thick Clients for Personal Wireless Devices," *Computer, IEEE*, vol. 5, no. 1, pp. 133–135, July 2002. Cited on pages: xi and 27.

[STO09] T. Spyropoulos, T. Turletti, and K. Obraczka, "Routing in Delay Tolerant Networks Comprising Heterogeneous Populations of Nodes," *IEEE Transactions on Mobile Computing*, vol. 8, no. 8, pp. 1132–1147, August 2009. Cited on page: 94.

[TLB+09] P.-U. Tournoux, J. Leguay, F. Benbadis, V. Conan, M. D. de Amorim, and J. Whitbeck, "The Accordion Phenomenon: Analysis, Characterization, and Impact on DTN Routing," in *Proceedings of the* 28^{th} *IEEE International Conference on Computer Communications (InfoCom'09)*, April 2009. Cited on page: 92.

[TNB+10] N. Thompson, S. C. Nelson, M. Bakht, T. Abdelzaher, and R. Kravets, "Retiring Replicants: Congestion Control for Intermittently-Connected Networks," in *Proceedings of the* 29^{th} *IEEE International Conference on Computer Communications (InfoCom'10)*, March 2010. Cited on page: 15.

[Tøn04] A. Tønnesen, "Implementing and extending the Optimized Link State Routing Protocol," August 2004. [Online]. Available: http://www.olsr.org/docs/report.pdf Cited on page: 11.

[VB00] T. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke University, CS-2000-06, July 2000. Cited on pages: 16, 36, 45, 54, 71, 73, and 108.

[War03] F. Warthman, "Delay-Tolerant Networks (DTNs), A Tutorial, V1.1," May 2003. [Online]. Available: http://www.dtnrg.org/docs/tutorials/warthman-1.1.pdf Cited on pages: 10, 12, 14, and 34.

[WB05] J. Widmer and J.-Y. L. Boudec, "Network Coding for Efficient Communication in Extreme Networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking (WTDN'05)*, 2005, pp. 284–291. Cited on page: 58.

[WC09] J. Whitbeck and V. Conan, "HYMAD: Hybrid DTN-MANET Routing for Dense and Highly Dynamic Wireless Networks," in *Proceedings of International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'09)*, IEEE, Ed., June 2009. Cited on pages: 59 and 60.

[WJMF05] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure-Coding Based Routing for Opportunistic Networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking (WTDN'05)*, 2005. Cited on page: 58.

[XLN06] Y. Xue, B. Li, and K. Nahrstedt, "Optimal Resource Allocation in Wireless Ad Hoc Networks: A Price-Based Approach," *IEEE Transactions On Mobile Computing*, vol. 5, no. 4, pp. 347–364, April 2006. Cited on page: 29.

[Yan04] W.-L. Yang, "Optimal and heuristic algorithms for quality-of-service routing with multiple constraints," *Elsevier Journal of Performance Evaluation*, vol. 57, no. 3, pp. 261–278, January 2004. Cited on page: 3.

[YCW09] Q. Yuan, I. Cardei, and J. Wu, "Predict and Relay: An Efficient Routing in Disruption-Tolerant Networks," in *Proceedings of the* 10^{th} *ACM international symposium on Mobile ad hoc networking and computing (MobiHoc'10)*, 2009. Cited on page: 94.

[YGC09] E. Yoneki, D. Greenfield, and J. Crowcroft, "Dynamics of Inter-Meeting Time in Human Contact Networks," in *Proceedings of the 2009 International Conference on Advances in Social Network Analysis and Data Mining (ASONAM'09)*, July 2009. Cited on pages: 23, 92, and 93.

[Zha06] Z. Zhang, "Routing in Intermittently Connected Mobile Ad Hoc Networks and Delay tolerant networks: Overview and Challenges," *IEEE Communication Surveys*, vol. 8, no. 1, pp. 24–37, 2006. Cited on pages: 15 and 21.

[ZNKT07] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 51, pp. 2867–2891, July 2007. Cited on pages: 17, 24, 73, and 76.