Message dissemination in mobile delay tolerant networks

Jingwei Miao

Thèse

Message Dissemination in Mobile Delay Tolerant Networks

Présenté devant:
L'Institut National des Sciences Appliquées de Lyon
(INSA de Lyon)

Pour obtenir:
Le grade de docteur

École doctorale:
Informatique et Mathématiques
Spécialité:
Informatique

Par:
Jingwei Miao

Soutenue le 29 March 2013 à l'INSA de Lyon

Jury:

Prof. Ernesto Damiani	University of Milan	Rapporteur
Dr. Laurent Réveillère	University of Bordeaux	Rapporteur
Prof. Vivien Quéma	INPG, Grenoble	Examinateur
Prof. Jean-Marc Nicod	University of Besançon	Examinateur
Dr. Omar Hasan	INSA Lyon	Examinateur
Prof. Lionel Brunie	INSA Lyon	Directeur de thèse
Dr. Sonia Ben Mokhtar	INSA Lyon	Co-directeur de thèse

Laboratoire d'InfoRmatique en Image et Systèmes d'information (LIRIS)

I understand that the thesis may be made electronically available to the public.

Contents

1 Introduction **1**
1.1 Overview of Mobile Delay Tolerant Networks 1
1.2 Routing in MDTNs 2
1.3 Research Issues in MDTNs 3
1.3.1 Routing Issues in MDTNs 3
1.3.2 Privacy Issue in MDTNs 4
1.4 Contributions 4
1.5 Publications 5
1.6 Thesis Outline 6

2 Background **7**
2.1 Short-range Communication Technologies 7
2.1.1 Bluetooth 7
2.1.2 802.11 WiFi 8
2.2 Infrastructure-less Networks: MANETs vs MDTNs 9
2.2.1 Mobile Ad-hoc NETworks 9
2.2.2 Mobile Delay Tolerant Networks 10
2.3 Mobility Models 11
2.3.1 Synthetic Mobility Models 11
2.3.2 Real World Traces 13
2.4 Routing in MDTNs 14
2.4.1 Classification of Routing Protocols 14
2.4.2 State of the Art in Routing Protocols 15
2.4.3 Classification of Routing Protocols 21
2.4.4 Characteristics of Routing Protocols in Each Category . . 21
2.5 Privacy Preserving Protocols in MDTNs 23

2.5.1 Classification of Privacy Objectives 23
2.5.2 Strategies for Preserving Privacy 24
2.5.3 Analysis of Strategies for Preserving Privacy 27
2.5.4 Classification of Strategies for Preserving Privacy 28
2.6 Chapter Review 28

3 An Adaptive Routing Protocol for MDTNs 30
3.1 Introduction 30
3.2 System Model 31
3.3 Overview of ARP 31
3.3.1 The Architecture of ARP 31
3.3.2 The Workflow of ARP 33
3.4 Detailed Design of ARP 34
3.4.1 Calculation of Centrality 34
3.4.2 Calculation of Regularity 35
3.4.3 Community Calculation 36
3.4.4 Normalization Function 37
3.4.5 Decision of Message Forwarding 38
3.5 Performance Evaluation 40
3.5.1 Dataset 40
3.5.2 Simulation Settings 40
3.5.3 Routing Protocols 43
3.5.4 Performance Metrics 44
3.5.5 Simulation Results 44
3.6 Chapter Review 47

4 A Delay and Cost Balancing Protocol for Efficient Routing in Mobile Delay Tolerant Networks 50
4.1 Introduction 50
4.2 System Model 52
4.2.1 A Delay Tolerant Network Model 52
4.2.2 Information Maintained by a Node 53
4.2.3 Maintenance of Information 54
4.3 Protocol Design 55
4.3.1 Overview of CAS 56
4.3.2 Design of CAS 57
4.3.3 CAS Generalizes Classes of Routing Protocols 59
4.4 Analysis 63
4.4.1 Analytical Model 63
4.4.2 Model Validation 67

4.5 Performance Evaluation 68
4.5.1 Simulation Settings 69
4.5.2 Routing Protocols 70
4.5.3 Performance Metrics 71
4.5.4 Simulation Results 71
4.6 Chapter Review 75

5 An Efficient Privacy Preserving Prediction-based Routing Protocol for Mobile Delay Tolerant Networks 77
5.1 Introduction 77
5.2 System Model 79
5.3 Privacy Preserving Prediction-based Routing 79
5.3.1 Protocol Description 79
5.3.2 Security Analysis: Correctness 81
5.3.3 Security Analysis: Privacy 82
5.4 Privacy Preserving Computation of Union 82
5.4.1 Protocol Description 82
5.4.2 Protocol Setting 83
5.4.3 Security Analysis: Correctness 85
5.4.4 Security Analysis: Privacy 86
5.4.5 Performance Analysis: Running Time 88
5.5 Experimental Evaluation 93
5.5.1 Simulation Settings 93
5.5.2 Mobility Model 94
5.5.3 Routing Protocols 95
5.5.4 Performance Metrics 96
5.5.5 Performance Results 96
5.6 Chapter Review 101

6 Perspective–Selfishness of Nodes in MDTN Routing 103
6.1 Introduction 103
6.2 Selfishness 103
6.2.1 Classification of Selfish Behavior 104
6.2.2 The Methodologies of Investigating the Impact of Selfish Behavior 104
6.2.3 The Impact of Selfish Behavior 105
6.3 Strategies for Preventing Selfish Behavior 106
6.3.1 Barter-based Strategies 106
6.3.2 Credit-based Strategies 107
6.3.3 Reputation-based Strategies 108

6.4 Experimental Analysis of Existing Strategies for Preventing Selfish Behavior . . . 110
6.4.1 Compared Strategies for Preventing Selfish Behavior . . . 110
6.4.2 Simulation Settings . . . 110
6.4.3 Routing Algorithms . . . 112
6.4.4 Performance Metrics . . . 112
6.4.5 Simulation Results . . . 112
6.4.6 Comparison of Strategies . . . 114
6.5 Chapter Review . . . 115

7 Conclusions and Future Work **116**
7.1 Conclusions . . . 116
7.1.1 An Adaptive Routing Protocol . . . 116
7.1.2 A Delay and Cost Balancing Routing Protocol . . . 117
7.1.3 A Privacy-Preserving Routing Protocol . . . 118
7.2 Future Work . . . 118

List of Figures

1.1 An example of message routing in MDTNs 2

2.1 An example of routing in MANETs 10
2.2 An example of message exchange in "store-carry-and-forward" mechanism 10
2.3 Community-based Mobility Model 13
2.4 Classification of routing protocols in DTNs 15
2.5 Categorition of privacy objectives 24

3.1 Protocol Architecture 32
3.2 An example of the workflow 34
3.3 Design space of forwarding decision maker in ARP 39
3.4 Illustration of the forwarding decision maker in ARP 39
3.5 MessageForwarding 41
3.6 Number of contacts per day in the whole MIT real ming data set . 42
3.7 Number of contacts per day in the period from Sept. 13, 2004 to Dec. 06, 2005 on MIT Real Mining dataset. 43
3.8 Delivery ratio comparison of the routing protocols 44
3.9 Delivery cost comparison of the routing protocols 45
3.10 Usage of the social properties in making forwarding decisions . . 46
3.11 Delivery ratio comparison of the routing protocols 47
3.12 Delivery cost comparison of the routing protocols 48
3.13 The impact of the settings of the encounter set on the routing performance of ARP. 48

4.1 Community Graph 54
4.2 Routing Protocol Overview 56
4.3 Algorithm:OptimizeCost 60

4.4 Protocol:RouteMessage 61
4.5 The transformation from CAS to several kinds of routing protocols, where L is the number of allocated copies for a message, V is the set of all nodes in a network, M is the set of all communities in a network. 62
4.6 The continuous time Markov chain model for modeling the flooding process in a community. States (1) to (N) are N transition states and state $(N+1)$ is the absorbing state. 64
4.7 The two-dimensional continuous time Markov chain model for modeling the message dissemination within two communities. States $(1,0)$ to $(M+1,N)$ are $(M+1)(N+1)$ transition states and state $(x,N+1)$ is the absorbing state. 66
4.8 Theoretical and simulation result comparison in the case of: (a) intra-community and (b) inter-community message exchange. The contact rates between nodes in the same community and different communities are $\lambda^{intra} = 1.703\ \mathrm{h}^{-1}$ and $\lambda^{inter} = 0.672\ \mathrm{h}^{-1}$ respectively. 69
4.9 Comparison of the routing performance of several algorithms in the single-community case. 72
4.10 Comparison of delivery ratio of several algorithms with different densities of high mobility nodes. 5% and 50% nodes are chosen as the high mobility nodes in (a) and (b) respectively. 73
4.11 Comparison of delivery cost of several algorithms with different densities of roaming nodes. 5% and 50% nodes are chosen as the roaming nodes in (a) and (b), respectively. 74
4.12 The impact of the settings of the mobility model on the routing performance of CAS 76

5.1 E3PR Protocol Overview 80
5.2 Protocol: MDTN-E3PR 81
5.3 Protocol: MDTN-Private-Union 84
5.4 The impact of m and k on the privacy, where $n = 31$ 88
5.5 The continuous time Markov model for modeling the general message dissemination in E3PR . States (1) to (N) are N transition states and state $(N+1)$ is the absorbing state. 90
5.6 Simulation Results of the running time of E3PR with a given contact rate $\lambda = 2.51\ \mathrm{h}^{-1}$. 92
5.7 Community Model 94
5.8 Delivery ratio wrt the increasing TTL of messages. 96
5.9 Delivery cost wrt the increasing TTL of messages. 97

5.10 Delivery latency wrt the increasing TTL of messages. 98
5.11 (a) delivery ratio, (b) delivery cost, and (c) delivery latency wrt the increasing size of communities. 99
5.12 The impact of the settings of the mobility model on the delivery ratio of E3PR 100
5.13 The impact of the settings of the mobility model on the delivery cost of E3PR 100
5.14 The impact of the settings of the mobility model on the delivery latency of E3PR 101

6.1 Classification of selfish behavior in DTNs 105
6.2 The routing performance in terms of delivery ratio under individual selfishness. The selfish actions of dropping and non-forwarding messages are illustrated in (a) and (b) respectively. 113
6.3 The routing performance in terms of delivery ratio under social selfishness. The selfish actions of dropping and non-forwarding messages are illustrated in (a) and (b) respectively. 114

List of Tables

2.1 Power classes of Bluetooth 8
2.2 Data transmission rate for each version of Bluetooth protocol . . . 8
2.3 Comparison of routing protocols in MDTNs 22
2.4 Comparison of strategies for preserving privacy 28

3.1 An example of the regularity table of node A; time slot is 1 hour, and time length is a week 36

4.1 Parameter settings 70

5.1 Parameter settings 94

6.1 Simulation Parameters for Strategies 111
6.2 Simulation parameters 111
6.3 Performance comparison of the selected strategies 114

Acknowledgements

I have so many people that I owe a huge debt of gratitude to. I would like to thank a few them here, but if I happened to not mention your name here, remember that what you have done for me is still fixed in my heart for ever.

First and foremost, I would like to express my sincere gratitude to my supervisor Prof. Lionel Brunie and co-supervisors Dr. Sonia Ben Mokhtar and Dr. Omar Hasan for their wise guidance, invaluable advises and powerful support throughout the course of this work. Lionel accepted me as a Ph.D. student and has been supporting me in my research through these years. His wide knowledge, personal guidance, and fatherly love have provided a good basis for this thesis. I am also very grateful to my co-supervisor Sonia for her scientific advises and knowledge and many insightful discussions and suggestions. She is my primary resource for getting my science questions and is instrumental in modifying papers and thesis. She always tried her best to help and save me when I was in the most difficult time. I also want to thank my co-supervisor and friend Omar for his advises on my studying and modifying my papers and thesis as well as his personal helps on my life in France.

I would like to thank Prof. Ernesto Damiani and Dr. Laurent Réveillère for reviewing my work and being members of the examination committee. I would also like to thank Prof. Vivien Quéma and Prof. Jean-Marc Nicod, who are also members of the examination committee.

I would like to thank my parents, Zongquan Miao and Qiuhua Song, without whom I could not have gone this far. It is the inherited gallantry and fortitude that give me the courage to confront difficulties. It is the inherited honesty and kindness that help me to get aid from such many good friends. All these properties help me to overcome all difficulties. Although we are thousands of miles apart, my heart has never left you.

I also would like to thank all staffs of LIRIS/INSA including Sylvie Calabretto, Nadia Bennani, Elod Egyed-Zsigmond, Pierre-Edouard Portier, Mabrouka Gheraissa for their help in the past years.

I also would like to thank all colleagues of LIRIS/INSA including Addissalem Negash, Sonia Lajmi, Zeina Torbey, Christian Vilsmaier, Tobias Mayer, Lyes Limam, Vanessa El-Khoury for their help in the past years.

Finally, I would like to thank Chinese Scholarship Council (CSC) for supporting my study in France, and also thank the MDPS German-French Doctoral College and the French National Research Agency (SocEDA, Grant ANR-10-SEGI-013) for funding my research in France.

Abstract

Mobile Delay Tolerant Networks (MDTNs) are wireless mobile networks in which a complete routing path between two nodes that wish to communicate cannot be guaranteed. A number of networking scenarios have been categorized as MDTNs, such as wildlife tracking sensor networks, vehicular ad hoc networks, pocket switched networks, etc.

The network asynchrony, coupled with the limited resources of mobile devices (e.g., sensors, smart phones, tablets) make message dissemination (also called routing) one of the fundamental challenges in MDTNs. In the literature, a large body of work has been done to deal with routing in MDTNs. However, most of the existing routing protocols are based on at least one of the following three assumptions: (1) all messages can be routed by relying on a single mobility property (e.g., the belonging or not to communities); (2) all messages can be routed using a single message allocation strategy (e.g., a fixed number of copies is used for all messages) (3) users are willing to disclose their mobility information and relationships to others in order to improve the quality of the routing.

We argue that the above three assumptions are not realistic because: (1) users can exhibit various social behaviors and consequently various mobility properties (e.g., they can have regular movements during week-days and exhibit non-predictable movements during week-ends); (2) some messages might need more or less copies to be delivered according to the localization of the source and the destination and to the urgency of the message; and (3) users mobility data can disclose sensitive information about the users.

In this thesis, we relieve MDTN routing from the above three restrictive assumptions.

Firstly, we propose an adaptive routing protocol for mobile delay tolerant networks. The proposed protocol can dynamically learn the social properties of nodes based on their mobility patterns, and exploit the most appropriate routing strategy each time an intermediate node is encountered. Simulations performed on real mobility traces show that our protocol achieves a better delivery ratio than existing state-of-the-art routing protocols that rely on a single mobility property.

Secondly, we present a delay and cost balancing protocol for efficient routing in mobile delay tolerant networks. The presented protocol reasons on the remaining time-to-live of a message to dynamically allocate the minimum number of copies that are necessary to achieve a given delivery probability. Evaluation results show that the protocol can achieve a good balance between message delivery delay and delivery cost, compared with most of the existing routing protocols in the literature.

Lastly, we propose an efficient privacy preserving prediction-based routing protocol for mobile delay tolerant networks. This protocol preserves the mobility patterns of a node from being disclosed by exploiting the mobility pattern of communities that node belongs to. Evaluation results demonstrate that this protocol can obtain comparable routing performance to prediction-based protocols while preserving the mobility pattern of nodes.

Keywords: Delay Tolerant Networks, Routing, Privacy, Mobility

Chapter 1

Introduction

1.1 Overview of Mobile Delay Tolerant Networks

In the last decade, mobile wireless networks have achieved a rapid development as the number of mobile devices increases. Simultaneously, we have witnessed the evolution of communication paradigms in mobile wireless networks. Initially, nodes (i.e., mobile devices) were communicating with each other by using long-range wireless technologies, such as GSM [82] and GPRS [6], through a pre-existing infrastructure. Subsequently, as the deployment of short-range wireless technologies (e.g., Bluetooth) in mobile devices has increased, nodes were able to directly communicate with each other by using their short-range wireless in an ad hoc manner, in which a pre-existing infrastructure has not been needed any more. This communication paradigm still depends on the assumption of a continuous end-to-end connectivity of nodes. However, due to the mobility of nodes and the limited resources of the mobile devices, the connectivity of nodes may suffer from being frequently disrupted and for a relatively long period of time. In order to enable communication between nodes in such environments, a novel form of networks, known as Mobile Delay Tolerant Networks (MDTNs), has emerged to represent a class of networks where a continuous end-to-end connectivity may not be possible.

MDTNs have a broad range of potential applications including scenarios with high delivery delay and scenarios with frequent disruptions and disconnections: military battlefields [76], vehicular communications [61], deep space communications [14].

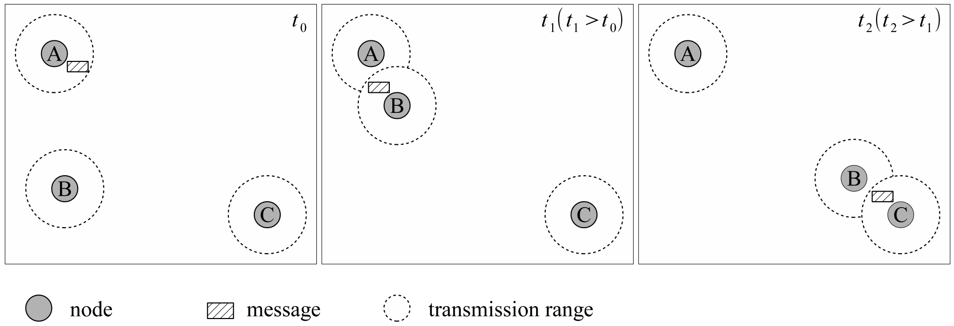


Figure 1.1: An example of message routing in MDTNs

1.2 Routing in MDTNs

Due to the intermittent connectivity in MDTNs, routing is one of the most fundamental problems in these networks. Contrary to Mobile Ad-hoc NETworks (MANETs) [104], a complete routing path between two nodes that wish to communicate cannot be guaranteed in MDTNs [30]. Therefore, the routing protocols developed for MANETs, such as Dynamic Source Routing (DSR) [77], Ad-hoc On-demand Distance Vector (AODV) [99], and Optimized Link State Routing (OLSR) [23], cannot suit the intrinsic characteristics of MDTNs.

To cope with frequent, long-term disconnections, routing in MDTNs is often performed in a "store-carry-and-forward" (also known as mobility-assisted [112], or encounter-based [28]) manner, in which a node (i.e., a mobile device) may store and carry a message for some time before forwarding the message to another node that comes within its transmission range [116]. The principle of this routing mechanism is to construct a complete routing path between the source node and the destination node of a message with the connections of intermediary nodes.

An example of message routing in a "store-carry-and-forward" manner is illustrated in Figure 1.1. The figure shows that an end-to-end path between node A and node C does not exist. At time instant t_0, node A has a message whose destination is node C. At time instant t_1 (which is later than t_0), because of their mobility, nodes A and B move within the transmission range of each other. Node A then forwards its message to node B. At time instant t_2 (which is later than t_1), the network topology is changed, nodes B and C are now in the transmission range of each other. The message is consequently forwarded from node B to node C.

1.3 Research Issues in MDTNs

In this section, we present two key research challenges with regard to message dissemination in mobile delay tolerant networks. These challenges motivate the work presented in this thesis.

1.3.1 Routing Issues in MDTNs

What the above "store-carry-and-forward" mechanism illustrates is that the communication between nodes in MDTNs, apart from utilizing a wireless network interface, also should take advantage of the nodes' mobility.

In order to better understand nodes' mobility (i.e., human mobility), various studies that collect and analyze real mobility traces have been conducted [35, 83, 10]. These studies have shown that: (1) the movement of humans in real-life is not random, but exhibits repetition in a certain extent; (2) human mobility is influenced by their social relationships. According to these observations, a number of social properties characterizing the mobility patterns of nodes have thus been defined. Among these properties, *centrality* characterizes the frequency of encounters a node has; *regularity* characterizes the probability that given encounters are repeated over time; and *community* characterizes a group of nodes that encounter frequently and regularly [128, 119].

Building on these social properties, a number of routing algorithms [47, 119, 78, 46, 48] have been proposed in the literature. However, these approaches exploit a single social property and generally evaluate their approach using mobility traces in which that property dominates. This assumption is very limitative as in reality various social behaviors can co-exist in a given environment.

Indeed, a node may have a central position in the network and be completely isolated in different time slots; it may exhibit a regular or a completely irregular mobility pattern depending on the time of the day/specific days of the week; it can be in/out a community during specific periods of time. In other words, most of the existing approaches cannot accommodate the actual dynamics of human social properties.

On the other hand, most of the existing routing protocols [117, 89, 111] in MDTNs utilize the same strategy to allocate message copies. The strategy of replicating messages utilized by these protocols is to either flood a message in the network [117] or allocate the same amount of copies for all messages [89, 111]. However, flooding messages causes a large number of redundant message copies, which issues in congestion and influences the successful message delivery. Moreover, allocating the same amount of copies for all messages cannot meet the requirements of messages with different Time-To-Lives (TTLs). For instance, compared with a message with a long TTL, a message with a short TTL generally needs a greater amount of copies to ensure to be delivered. Consequently, these routing protocols are inefficient to allocate message copies for messages with different TTLs.

Summarizing, the existing routing protocols suffer from the following two major problems: (1) using a single social property to guide message forwarding among nodes in a network, and cannot cope with the dynamics of social behaviors. (2) using the same strategy to allocate message copies for messages, and cannot deal with heterogeneous message TTLs.

1.3.2 Privacy Issue in MDTNs

Routing in MDTNs inherently depends on the participation of intermediary nodes. In order to better choose intermediary nodes, a number of routing protocols [70, 25, 78] utilize the encounter probability of nodes, which is derived from the history of nodes' encounters, to determine the routing decisions [123, 125]. Such routing protocols are known as prediction-based routing [123, 125]. It has been shown that these protocols perform better than other protocols when nodes exhibit well known mobility patterns. However, these prediction-based routing protocols implicitly assume that nodes accept to reveal their mobility patterns to other nodes. Unfortunately, such an assumption is not realistic, since this information can be used to infer private information, as demonstrated by Gambs et al. [33]. The fear that the private information is disclosed to other nodes hinders MDTNs from being largely deployed.

Summarizing, the existing prediction-based routing protocols suffer from the following problem: disclosing the private information of nodes in the message routing process.

1.4 Contributions

This thesis makes several contributions addressing the research issues outlined above. The basic features and novelties of the proposed routing algorithms are the following:

- We propose an adaptive routing protocol, in which we dynamically learn the social properties of nodes based on their mobility patterns, and exploit these properties to make forwarding decisions each time an intermediate node is encountered. As a result of the adaptation to the dynamics of the social properties of nodes, we get the advantage of a higher delivery ratio then existing state-of-the-art routing protocols that rely on a single mobility property.

- We propose a delay and cost balancing routing protocol [80], in which we reason on the remaining time-to-live of a message to dynamically distribute the minimum number of copies that are necessary to achieve a given delivery probability. Since the number of message copies are dynamically controlled based on the remaining time-to-live of a message, we achieve a good balance between message delivery delay and delivery cost, compared with most of the existing routing protocols in the literature.

- We propose a privacy-preserving prediction-based routing protocol [42], in which we exploit the mobility patterns of the communities nodes belong to. By using the community structure of nodes, we preserve the mobility pattern of individual nodes while obtaining comparable routing performance to existing prediction-based protocols.

- We investigate strategies for preventing selfish behavior in mobile delay tolerant networks [81]. First, we propose a classification of selfish behavior. Second, we review the different strategies proposed for preventing selfish behavior in MDTNs. Lastly, we conduct experiments to compare the performance of these strategies for preventing different types of selfish behavior.

1.5 Publications

The work presented in this thesis has resulted in the following publications in international journals and conferences:

International Journal Papers

[J1] **Jingwei Miao**, Omar Hasan, Sonia Ben Mokhtar, Lionel Brunie, Kangbin Yim, "An investigation on the unwillingness of nodes to participate in mobile delay tolerant network routing," *International Journal of Information Management*, vol.33, no.2, 2013.

International Conference Papers

[C1] Omar Hasan, **Jingwei Miao**, Sonia Ben Mokhtar, Lionel Brunie, "A privacy preserving prediction-based routing protocol for mobile delay tolerant networks," *In: Proceedings of IEEE International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 2013, pp. 1–8.

[C2] **Jingwei Miao**, Omar Hasan, Sonia Ben Mokhtar, Lionel Brunie, "A self-regulating protocol for efficient routing in mobile delay tolerant networks," *In: Proceedings of IEEE International Conference on Digital Ecosystems Technologies (DEST)*. IEEE, 2012, pp. 1–6.

[C3] **Jingwei Miao**, Omar Hasan, Sonia Ben Mokhtar, Lionel Brunie, Kangbin Yim, "An analysis of strategies for preventing selfish behavior in mobile delay tolerant networks," *In: Proceedings of IEEE International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*. IEEE, 2012, pp. 208–215.

[C4] **Jingwei Miao**, Omar Hasan, Sonia Ben Mokhtar, Lionel Brunie, "An adaptive routing algorithm for mobile delay tolerant networks," *In: Proceedings of IEEE International Symposium on Wireless Personal Multimedia Communications (WPMC)*. IEEE, 2011, pp. 1–5.

[C5] **Jingwei Miao**, Omar Hasan, Lionel Brunie, "Leveraging node centrality and regularity for efficient routing in mobile peer-to-peer networks," *In: Proceedings of International Conference on Data Management in Grid and Peer-to-Peer Systems*, pp. 83–94, 2011.

1.6 Thesis Outline

This thesis is organized as follows. In Chapter 2, we the state of the art related to the topics covered in the thesis. In Chapter 3, we present an adaptive routing protocol for MDTNs, which can dynamically adapt to the user's social properties at the very specific time and locations. In Chapter 4, we present a self-regulating routing protocol, which can dynamically allocate more or less message copies according to the urgency of a message. In Chapter 5, we present an efficient privacy preserving prediction-based routing protocol for MDTNs, which can achieve a comparable performance to the existing prediction-based routing protocols while preserving the privacy of nodes. In Chapter 6, we present an analysis of strategies for preventing selfish behavior in MDTNs, which is another reason that hinders users from participating in message routing. Finally, we summarize the main contributions of this thesis and point out some future directions in Chapter 7.

Chapter 2

Background

2.1 Short-range Communication Technologies

The increasing pervasiveness of mobile devices with short-range networking capability offers novel communication opportunities. Nodes can directly communicate when they come within the radio range of each other with no need for any pre-installed networking infrastructure. In this section, we give an overview of two kinds of the most predominant short-range wireless communication technologies used in MDTNs: Bluetooth and WiFi, which are widely deployed in most of mobile devices (e.g., laptops, smart phones, and tablets).

2.1.1 Bluetooth

Bluetooth was first proposed by Ericsson in 1994 as a wireless alternative to RS-232 [110] data cables. It has become the standard that aims to enable communication between devices over short distances. Nowadays, Bluetooth is widely deployed in mobile devices, such as personal digital assistants, mobile phones, laptops, or digital cameras. Bluetooth operates on the 2.45 GHz frequency band. It can operate in three power classes (see Table 2.1) with different communication ranges.

Moreover, different versions of the Bluetooth protocol have been implemented, which differ in their data transmission rate. The detailed information of data transmission rate in each version of Bluetooth protocol is illustrated in Table 2.2.

In the case of MDTNs, Bluetooth generally operates on class 2 with version 2.0 due to the following reasons: (1) class 2 model provides an acceptable ratio range ($\sim$ 10 m) for communication between nodes; (2) class 2 model can save the

Table 2.1: Power classes of Bluetooth

Class	Maximum permitted power (mW)	Maximum permitted power (dBm)	Range (m)
Class 1	100	20	~ 100
Class 2	2.5	4	~ 10
Class 3	1	0	~ 1

Table 2.2: Data transmission rate for each version of Bluetooth protocol

Version	Data rate	Maximum application throughput
Version 1.1 & 1.2	1 Mbit/s	0.7 Mbit/s
Version 2.0	3 Mbit/s	2.1 Mbit/s
Version 3.0	24 Mbit/s	

battery which is limited in mobile devices; (3) version 2.0 can provide a good data transmission rate for communication.

2.1.2 802.11 WiFi

802.11 was first proposed by the Institute of Electrical and Electronics Engineers (IEEE) [91] in 1997 as the first wireless networking standard. Nowadays, 802.11 has evolved into a class of specifications which have been created by IEEE for wireless local area networks (WLANs), such as 802.11 a/b/g/n. These specifications focus on the physical layer and the MAC (medium access) layer of the Open Systems Interconnection (OSI) model. The 802.11 family comprises several standards with different characteristics according to transmission speed and used frequency band. As an example, 802.11a (54 Mbit/s, 5 GHz) 802.11b (11 Mbit/s, 2.4 GHz) or 802.11g (54 Mbit/s, 2.4 GHz).

The term Wireless Fidelity (WiFi or Wi-Fi) was first used by US Patent and Trademark Office (USPTO) [97] in 1999. It then was used by Wi-Fi Alliance [1] to represent any WLAN products that are based on the IEEE 802.11 standards. Since most of the modern WLANs are based on these standards, the term Wi-Fi is used as a synonym for WLAN.

The most widely used protocols in the 802.11 family are the 802.11 (b/g/n). These protocols can form two types of networks: infrastructure-less networks, and infrastructure networks. In an infrastructure-less network, nodes can directly communicate with each other; while in an infrastructure network, the communication between two nodes is performed via an access point. Generally speaking, an access point is a fixed station, often connected to the Internet, that acts as a communication hub between two devices that want to communicate with each other. Thus, each message from a node to another is relayed through the access point. This approach has two advantages. First, the wireless network coverage is extended. For two nodes to communicate, they do not need to be in communication range with each other, just in communication range with the access point. Second, an access point can help mobile nodes save power by buffering frames at the access point for the mobile node. The node itself stays in power-save mode most of the time and just wakes up to receive buffered frames if available.

Communication range differs between the specific standards. For example, 802.11b spans about 150 meters (outdoors) and 802.11g only 25 meters. Both ranges are suitable for the mobile delay tolerant network environments.

2.2 Infrastructure-less Networks: MANETs vs MDTNs

According to whether an end-to-end routing path between two nodes that wish to communicate can be guaranteed or not, infrastructure-less networks can be classified into two categories: mobile ad hoc networks and mobile delay tolerant networks.

2.2.1 Mobile Ad-hoc NETworks

Mobile Ad-hoc NETworks (MANETs) are self-configuring infrastructure-less networks, in which nodes (i.e., mobile devices) are connected via a short-range wireless interface [44]. Nodes in MANETs are free to move and organize themselves arbitrarily. As a result, the network topology may change unpredictably and frequently.

Even though the network topology in MANETs may arbitrarily and frequently change, routing in MANETs is still based on the model of connectivity, that is, they assume that there is an end-to-end routing path between two nodes that wish to communicate. Depending on the number of nodes in their routing path, routing in MANETs can be divided into two types: one-hop routing and multi-hop routing. In one-hop routing, message is transferred from a node to another node at the time it comes in the transmission range of the former; while, in the multi-hop routing, intermediate nodes are utilized to connect two nodes that wish to communicate and are not directly in the transmission range of each other. It is worth noting that a contemporaneous end-to-end path is established via the intermediate nodes. An example of routing in MANETs is illustrated in Figure 2.1.

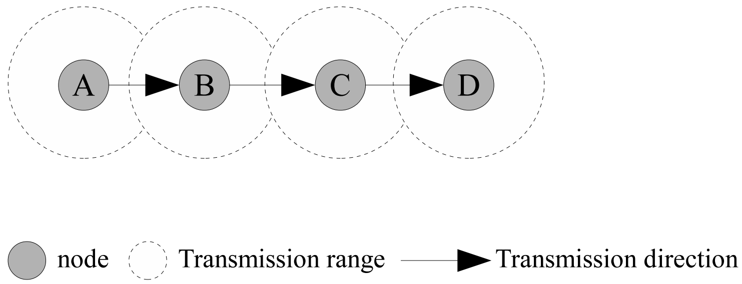


Figure 2.1: An example of routing in MANETs

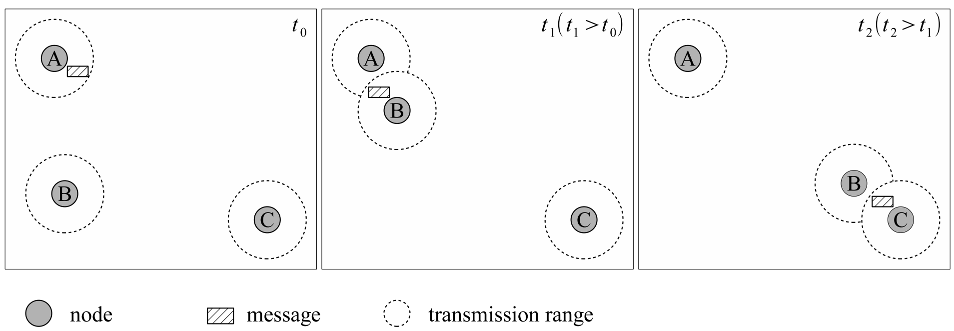


Figure 2.2: An example of message exchange in "store-carry-and-forward" mechanism

2.2.2 Mobile Delay Tolerant Networks

Routing in MANETs is based on the assumption that a complete routing path between two nodes that wish to communicate can be guaranteed. However, such assumption may be violated due to the mobility of the nodes and the limited resources of the mobile devices. Consider such a case where end-to-end delays can be of several minutes, or even hours, which is much longer than protocols such as TCP can handle, or where a completely connected end-to-end path through the network rarely, or never, exists between two nodes wishing to communicate. Such communication networks are commonly classified as Delay Tolerant Networks (DTNs).

The key difference between DTNs and MANETs is that a contemporaneous routing path between two nodes that wish to communicate cannot be guaranteed in DTNs, but the union of network snapshots over time can present an end-to-end routing path. An example of routing in DTNs is illustrated in Figure 2.2.

The forms that a DTN show can be classified into two modes [98]: infrastructure-less model or infrastructure-based model. In the former model, nodes can communicate when they come within the communication range of each other; while, in the latter model, nodes can communicate via an infrastructure. In the literature, the term Mobile Delay Tolerant Networks (MDTNs) are utilized to express the former kind of DTNs. This is the kind of DTNs considered in this thesis.

2.3 Mobility Models

As discussed in Chapter 1.2, routing in MDTNs takes advantage of nodes' mobility, which is represented by the term *mobility model*. Generally speaking, mobility models are divided into two categories [72]: synthetic models and real-world traces [86]. In a synthetic model, the nodes' mobility is generated by a mathematical model, while in a real-world trace, the nodes' mobility is collected from a trace of human movements in a real-life scenario. In the following sections, we introduce the most widely utilized mobility models in the literature.

2.3.1 Synthetic Mobility Models

Random WayPoint Mobility Model

The Random WayPoint (RWP) mobility model was first presented by Johnson et al. in [52]. In this model, each node is initially assigned a random position (called waypoint) within a given area. At the same time, each node is specified a random waypoint as its destination, and it then moves towards the destination with a random speed. The speed is chosen according to a uniform distribution in $[v_{min}, v_{max}]$, where v_{min} and v_{max} are the minimum and maximum speed of nodes respectively. When the node reaches the destination, it stays there for a certain period of time, called pause time. When the pause time expires, it randomly chooses a new waypoint as the destination, and repeats the above process.

However, according to the study in [88, 63], it has been observed that the stationary distribution of the location of a node, sampled at random time instants, is more concentrated near the center of the simulation area. Density waves in the average number of neighbors thus are produced by the RWP mobility model. A density wave is the clustering of nodes in one part of the simulation area. In order to avoid the initialization problem of RWP, Camp et al. [18] introduced the following three solutions: (1) save the position of nodes after a simulation that has executed long enough to be past this initial high variability, and use this position file as the initial position of nodes in the future simulations; (2) initially distribute nodes in a manner that maps to a distribution more common to the model; (3) discard the initial 1000 s of simulation time produced by the RWP mobility model in each simulation. The third solution is widely utilized [81].

Random Direction Mobility Model

The Random Direction (RD) mobility model [103] was introduced for modeling users movement within cellular systems [43, 39]. The RD mobility model is defined as follow. As in the RWP model, each node is initially assigned a random position, which is uniformly distributed within a simulation area. Each node then chooses a direction φ uniformly distributed in $[0, 2\pi]$. It also chooses a speed according to the uniform distribution in $[v_{min}, v_{max}]$, where v_{min} and v_{max} are the minimum and maximum speed of nodes respectively. It then moves in the direction φ until it reaches the borders of the simulation area. When a boundary of the simulation area is reached, it stays there for a certain period of time (i.e., a pause time). When the pause time expires, it randomly chooses a new direction uniformly distributed in $[0, \pi]$ and a random speed. It then repeats the above process.

Many variants of the RD mobility model have been proposed. We briefly introduce three variants of the RD mobility model, which are widely used. The first variant of the RD mobility model was presented by Royer et al. [103]. In [103], each node is also assigned a travel duration before it starts to move with a random speed in a random direction. It then stops and changes its direction if the assigned travel duration expires before it reaches the borders of the simulation area. In such a case, a new direction is uniformly chosen in $[0, 2\pi]$. The second variant of the RD mobility model was presented by Haas [40]. In [40], when a node reaches a boundary of the simulation area, it instantaneously re-enter into the simulation area from the opposite boundary. I then continues to move with the previous speed and in the previous direction. The third variant of the RD mobility model was presented by Bettstetter [5]. In [5], when a node reaches a boundary of the simulation area, it bounces off and continues to move with the previous speed in a new direction, which is $\pi - \varphi$.

Unlike the RWP mobility model, the stationary distribution of the nodes' positions, which is sampled at random time instants, has a uniform distribution.

Community-based Mobility Model

Recent studies [10, 53, 45] on the spatial characteristics of human mobility based on real world traces demonstrate that nodes (i.e., humans) in real-life tend to visit some locations more frequently than others. For instance, Kim et al. in [59] conducted the investigation of the spatial characteristics of human mobility by analyzing the mobility traces of students in the campus of Dartmouth College. The investigation shows that students spend most of their school time at several specific locations in the campus such as the cafeteria, the library, and the study halls. Moreover, this investigation also demonstrates that human movement is driven by

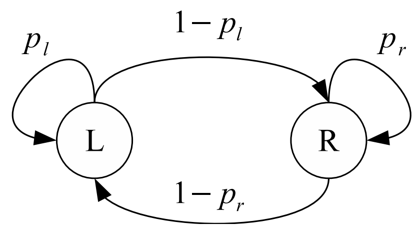


Figure 2.3: Community-based Mobility Model

social relationships. For instance, the engineering students tend to stay in the engineering hall most of their time, while the students in computer science visit the computer science hall most often [25]. These observations are consistent with the research works in [10, 84].

Building on the above observations, Spyropoulos et al. in [112] presented a community-based mobility model. It has been widely utilized to evaluate the routing performance of community-based routing protocols [114, 25], since it is motivated by the traces from real world and proved to better resemble human movement in real-life [114].

In this mobility model, each node is associated with a geographical area as its local region. The local region of a node is the region it always prefers to visit. Depending on whether a node moves inside its local region or not, the movement of a node can be defined to consist of a sequence of *local* and *roaming* epochs. More specifically, a local epoch of a node is a random direction movement restricted inside its local region, while a roaming epoch of the node is a random direction movement inside the entire network. If the previous epoch of a node was a local one, the next epoch is a local one with probability p_l, or a roaming epoch with probability $1-p_l$. Similarly, if the previous epoch of the node was a roaming one, the next epoch is a roaming one with probability p_r, or a local one with probability $1-p_r$. The state transition between local and roaming epochs is shown in Figure 2.3. The nodes whose local regions are the same consist of a community, due to the above observations.

2.3.2 Real World Traces

Reality Mining Trace

The Reality Mining trace [27] was collected by the Reality Mining project group from MIT Media Labs. It is an experimental study involving about 97 people for the duration of 9 months. Each person was given a Nokia 6600 cell phone with a software that continuously logs data about the location and contacts of the cell phone. The logged data from all the cell phones total around 350K hours of monitoring time and fit into a database of 1GB size.

Cambridge Trace

The cambridge trace [106] was collected by Scott et al. in Cambridge university. This trace includes the contacts about 36 people for 3 days. Each people was asked to carry the mobile devices (i.e., iMotes) with them at all times for the duration of the experiment. In addition, a number of stationary nodes were deployed in various locations that we expected many people to visit such as grocery stores, pubs, market places, and shopping centers in and around the city of Cambridge, UK. A stationary iMote was also placed at the reception of the Computer Lab, in which most of the experiment participants are students.

2.4 Routing in MDTNs

2.4.1 Classification of Routing Protocols

In the literature, a variety of routing protocols have been proposed in MDTNs. The existing routing protocols can be classified from different aspects [125, 19]. In the following paragraphs, the existing routing protocols in DTNs are classified with respect to their strategies of controlling message copies and their strategies of making the forwarding decision.

According to the number of destination nodes of a message, routing protocols can be classified into three categories: unicast routing, multicast routing, and broadcast routing. In unicast routing, there is a single destination for each message. In multicast routing, there is a group of destination nodes for each message. In broadcast routing, all the nodes in the network are the destination nodes for each message.

Moreover, depending on whether a message is partitioned into multiple fragment packets by the source node or not, the existing routing protocols can be divided into two categories: coding-based or coding-free. The technology of message coding is generally utilized to achieve the following two goals: protecting the routed messages and routing big data. For instance, Jansen and Beverly in [51] proposed a coding-base privacy-preserving strategy, in which each message is partitioned into multiple packets which are separately routed to the destination node via multiple independent paths.

Depending on the number of message copies utilized in the routing process, protocols can be classified into two categories [115, 114, 28, 31]: single-copy and multi-copy. In single-copy routing protocols, only a single copy for each message

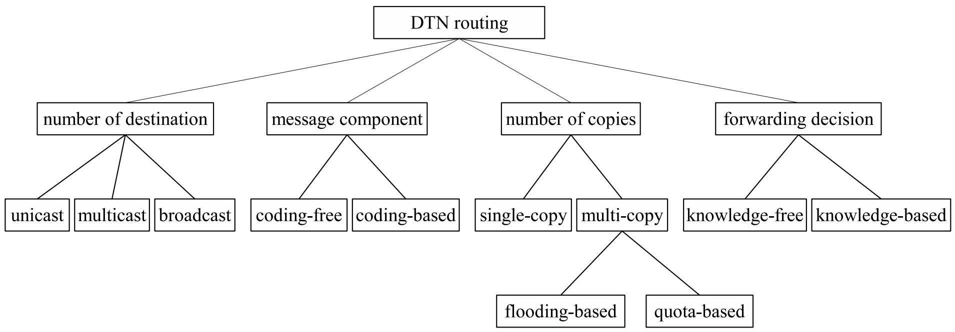


Figure 2.4: Classification of routing protocols in DTNs

exists in the network at any instant; while in multi-copy routing protocols, multiple copies of a message can be distributed into the network. Moreover, based on whether the number of message copies of a message is limited or not, multi-copy routing protocols can be further divided into flooding-based [117] and quota-based [89, 111]. Flooding-based routing protocols forward a copy of each message to as many nodes as possible, whereas quota-based routing protocols intentionally limit the number of message copies.

In addition, according to whether the forwarding decision is based on the knowledge derived from the nodes' encounters or not, protocols can as well be classified into two categories: knowledge-free and knowledge-based. Knowledge-free routing protocols make forwarding decision without relying on the nodes' capacity of delivering a message to the destination node, while knowledge-based routing protocols utilize some knowledge, such as encounter probability of nodes and community structure, to make the forwarding decision.

Figure 2.4 shows a classification of existing routing protocols based on the above aspects. It is worth noting that a routing protocol may be classified into multiple categories according to different aspects.

2.4.2 State of the Art in Routing Protocols

The contributions of this thesis on the routing issues involve the following two aspects: (1) how to make the forwarding decision, and (2) how to allocate message copies. Therefore, we introduce the work related to the above two aspects in the following sections.

Direct Delivery [38]

In Direct Delivery routing, the source node of a message keeps the message in its buffer until it encounters the destination node or the Time-To-Live (TTL) of the message expires.

It is worth noting that Direct Delivery is traditionally considered to achieve the lowest delivery probability and consume the fewest cost in terms of the number of message copies.

First Contact [38]

In First Contact routing, after receiving (or generating) a message, a node forwards the message to the first node that it encounters, and then deletes the message. Such forwarding process is repeated until the message is delivered to the destination node or the TTL of the message expires.

Seek and Focus [115]

In Seek and Focus routing, at the beginning, a message is randomly forwarded to an encountering node which does not yet hold the message until it reaches a node with a high enough utility. The concept of utility indicates how efficiently a node can deliver the message to the destination. Afterwards, the message is forwarded based on the utility of the encountering nodes until it reaches the destination or the TTL of the message expires. Note that only one copy for each message exists in the routing process.

Two-Hop Forwarding [38]

Grossglauser and Tse [38] presented a simple routing protocol named Two-Hop Forwarding. In Two-Hop Forwarding, messages are relayed through only one intermediate node, that is, the number of hops of message forwarding is limited to two hops. More specifically, the source node of a message randomly sends the message to an encountering node without the message. This latter then keeps the received message until it encounters the destination node or the TTL of the message expires.

Epidemic Routing [117]

In Epidemic routing, the messages in the network diffuse similarly to viruses by pairwise encounters of nodes. Specifically, when two nodes encounter each other, each of them complements the missing messages according to the messages in the other's buffer. A node is "infected" if it accepts and relays a message.

Note that Epidemic is traditionally considered to achieve the highest delivery probability and consume the most cost in terms of the number of message copies.

Gossiping [41]

Haas et al. in [41] presented a randomized flooding routing protocol, named Gossiping routing. Unlike the message propagation in Epidemic, in Gossiping, a message is replicated to an encountering node without the message with some probability $p < 1$. Therefore, it is also known as Probabilistic routing. Moreover, compared to Epidemic routing, Gossiping routing can reduce the delivery cost but with potentially lower delivery ratio and higher delivery latency.

(p,q)**-Epidemic Routing [79]**

Matsuda and Takine [79] presented a routing protocol, named (p,q)-Epidemic Routing, on the basis of the randomized flooding routing. In (p,q)-Epidemic routing, messages are forwarded in a probabilistic manner, with p (respectively q) being the probability of a relay node (respectively source node) transmitting a message to another node when they meet. It is also demonstrated that, according to the different settings regarding p and q, several previous algorithms are special cases of (p,q)-Epidemic Routing, such as Direct Delivery ($p = 0$, $q = 0$), Probabilistic routing ($0 < p = q < 1$), Two-Hop forwarding ($p = 0$, $q = 1$), and Epidemic routing ($p = 1$, $q = 1$).

Spray and Wait (SW) [111]

Spyropoulos et al. [111] presented a quota-based routing protocol, named Spray and Wait (SW). In Spray and Wait (SW), each message is associated with some forwarding tokens, which indicate message copies. Each message has two phases: the spray phase and the wait phase. A message is in the spray phase, if there is more than one forwarding token left. During the spray phase, the forwarding tokens of a message can be sprayed to an encountering node without the message. If there is only one forwarding token left, the message is in the wait phase. During the wait phase, a message can only be forwarded to the designation node.

The author presented two versions of Spray and Wait: Source Spray and Wait (SSW) and Binary Spray and Wait (BSW). In SSW, when the source node of a message encounters a node without the message, it forwards a copy of the message with one forwarding token to the encountering node. In BSW, when a node carrying a message in the spray phase encounters another node without the message, the node forwards a copy of the message with half forwarding tokens to the encountering node, and keep the message with the remaining forwarding tokens.

Encounter Based Routing (EBR) [89]

Nelson et al. in [89] presented a quota-based routing protocol, named EBR. Similar to SW, the source node of a message in EBR disseminates a predetermined number of forwarding tokens for each message. The difference between EBR and SW is that the number of message copies forwarded to an encountering node is dynamically computed based on a windowed degree centrality, that is, the number of nodes that encountered the node in a given period of time. More specifically, in EBR, the percentage of message copies forwarded from a node to an encountering node is proportional to the ratio of the centrality value of the encountering node and that of the node. Therefore, compared with SW routing, EBR is more efficient by spraying the copies of a message to better intermediate nodes which have a high capability of delivering the message to the destination node.

Spray and Focus (SF) [113]

Spyropoulos et al. in [113] presented a quota-based routing protocol, named Spray and Focus (SF). Similar to SW, depending on the number of forwarding tokens associated to a message, each message has two phases: the spray phase and the focus phase. The meaning of the spray phase and the focus phase is the same as that in SW. SF has a more complex second phase than SW. Specifically, in the focus phase, a message is forwarded from a node with lower utility value to a node with higher utility value. Therefore, SF exploits all the advantages of quota-based routing, but is also able to identify appropriate forwarding opportunities that could deliver the message faster.

SAURP [28]

Elwhishi et al. in [28] introduced a quota-based routing protocol, called Self Adaptive Utility-based Routing Protocol (SAURP). As the same as other quota-based routing protocols [111, 89, 113], SAURP also allocates a predetermined number of forwarding tokens for each message, at the time the message is generated by a source node. SAURP similarly to SF allows the message with only one forwarding token left to be forwarded to better intermediary nodes. SAURP differs with SF in the computation of the number of message copies forwarded to an encountering node. In SAURP, the percentage of message copies forwarded from a node to an encountering node is proportional to the ratio of the average inter-contact time with the destination of the encountering node and that of the node. Note that the average inter-contact time is similar to the encounter probability. For instance, a short average inter-contact time between two nodes indicate a high encounter probability.

Multiperiod Spray and Wait (MSW) [12]

Bulut et al. [12] presented a Multiperiod Spray and Wait (MSW) routing protocol for delay tolerant networks. MSW aims to minimize the average copy count used per message until the delivery while maintaining the predefined message delivery rate by the given deadline. In CEMS, the number of copies for each message is allocated in multiperiod. At the beginning of each period, some number of additional copies of a message are allocated to the source node. MSW computes the number of additional copies of the message, depending on the urgency of achieving the delivery rate by the given deadline for that message. The allocation process of message copies is repeated until the message is delivered to the destination node or the TTL of the message expires.

PRoPHET [70]

Similar to Gossiping routing, Lindgren et al. in [70] presented a Probabilistic ROuting Protocol using History of Encounters and Transitivity (PROPHET). PROPHET takes advantage of the encounter probability of nodes to guide the message forwarding. When two nodes encounter each other, they exchange a delivery probability vector, which contains the encounter probability of each known node of them. The encounter probability is computed based on previous encounters of nodes and is subject to an ageing factor. Based on the direct encounter probability, the delivery probabilities of nodes which have never directly encountered each other can be computed. Following the vector exchange, a message is forwarded from the node with the lower delivery probability to the node with the higher delivery probability. The forwarding process is repeated until the message is forwarded to the destination node or the TTL of the message expires.

MaxProp [13]

Burgess et al. in [13] introduced a flooding-based routing protocol, named MaxProp. Similar to Epidemic routing, when two nodes encounter, each of them tries to complement the missing messages according to the messages in the others buffer. Unlike Epidemic routing, the message forwarding decision depends on the probability that the two nodes will encounter soon the destination. The encounter probability is utilized as well to determine which messages should be deleted when a node's buffer space is almost full.

RANK [47]

In RANK, the main concept is that each node maintains a windowed degree centrality value, which indicates how many nodes it can encounter in a given period of time. The centrality of nodes is then employed to guide the message forwarding. More specifically, a message is forwarded from the node with lower centrality value to the node with higher centrality value until it reaches the destination node or the TTL of the message expires.

SBR [29]

Fabbri and Verdone in [29] presented a Sociability-Based Routing (SBR) scheme for DTNs. SBR can be considered as an extension of RANK. SBR differs with RANK in the calculation of the centrality value of nodes. In SBR, the computation of the centrality value of nodes does not only include the direct encounters between nodes but also the multiple hops indirect encounters between nodes through intermediate nodes.

LABEL [46]

In LABEL, it is assumed that each node possesses a label indicating its community. The label of the nodes (i.e., the community structure) is then utilized to guide the message forwarding. Specifically, a source node keeps a message until it encounters the destination or a node in the same community as the destination node. After wards, the message is flooded inside the community of the destination node until the TTL of the message expires. Therefore, LABEL can be considered as a combination of Direct Delivery and Epidemic.

BUBBLE [48]

Hui et al. in [48] presented a community-based protocol that utilizes social information about nodes, such as their centrality and the community to which they belong. In this protocol, a message is forwarded based on the global rankings of two encountering nodes, until it reaches a node in the same community as the destination node. After that, the message is forwarded based on the local rankings of two encountering nodes, until it either reaches the destination node or the TTL of the message expires.

IFR [66]

In IFR, a message is forwarded from a node with lower centrality value to a node with higher centrality value until it reaches the destination node or a node in the same community of the destination node. After that, the message is flooded inside the community of the destination node until the TTL of the message expires. Therefore, IFR can be considered as a combination of RANK and Epidemic.

Clustering Routing [25]

Dang and Wu in [25] introduced a community-based routing protocol, named Clustering routing. The encounter information of nodes is synchronized and processed to identify the community structure and the gateway nodes. A gateway node connecting its community to another community is the node that has the highest probability of encountering any node in the latter community. In this protocol, Direct routing is employed to route a message to a gateway node or the destination node if the message is in the destination nodes community.

Habit [78]

Mashhadi et al. in [78] presented a regularity-based routing protocol, called Habit. In Habit, each node maintains a regularity table, which records the encounter probability of a pair of two nodes in time slots. When a source node generates a message, it looks up the local regularity table to find the optimal routing path along which it can achieve the best delivery probability before the TTL of the message expires. The message is then forwarded based on the pre-determined routing path. Note that Habit is a single-copy routing protocol.

3R [119]

Vu et al. in [119] proposed a regularity-based routing protocol, named 3R. In 3R, when a message carrier (i.e, a node) encounters another node, each of them calculates the cumulative probability of encountering the destination node in the time slots covering the remaining TTL of the message. The message is then forwarded to the encountering node if it has a higher probability of encountering the destination node. Such forwarding process is repeated until the message is delivered to the destination or the TTL of the message expires.

SimBet [24]

Daly and Haahr in [24] proposed a routing protocol based on social network analysis, named SimBet. SimBet utilizes similarity and centrality at the same time. In the context of SimBet, similarity is defined as the ratio of common neighbors between individuals in social networks. It is thus similar to the concept of community. In SimBet, message forwarding is guided by a utility value, which is the sum of the values of centrality and similarity multiplied by their weights. The utility value of a node indicates its capacity of delivering a message to a destination node.

Friendship-Based Routing (FBR) [11]
In [11], Bulut and Szymanski presented the Friendship-Based Routing (FBR) protocol for delay tolerant networks. FBR takes advantage of community and regularity to guide the message forwarding. In the context of FBR, the friendship between nodes indicates the frequency and duration of the connectivity between nodes. The friendship between nodes is thus utilized to detect community structure of nodes and guide message forwarding. Moreover, unlike the previous works [48, 46] where a single community structure maintained by each node, in [11], Bulut and Szymanski considers that the community structure of nodes evolves periodically. Therefore, each node maintains a community structure for a given time slots according to the history of its encounters.

After the strength of friendship and community structure of nodes are computed and constructed, a node carrying a message forwards the message to an encountering node, if and only if the encountering node has a stronger friendship with the destination node and the encountering node's community structure in the current time slot contains the destination.

2.4.3 Classification of Routing Protocols

A classification of the above routing protocols is given in Table 2.3 according to the established criteria and sorted by their year of publication.

2.4.4 Characteristics of Routing Protocols in Each Category

In the above sections, we have introduced and classified the state of the art routing protocols. In the following sections, we will analyze the advantages and disadvantages of the routing protocols in each category.

In single-copy routing protocols, such as Direct Delivery [115] and First Contact [50], only a single copy for each message exists in the network at any instance. Therefore, these routing protocols achieve the minimum transmission overhead in terms of number of message copies. However, due to the inherent characteristic of frequent and long-term network partitions in MDTNs, these protocols often suffer from low delivery ratio and long delivery latency [38] even if a message is eventually delivered to the destination.

In order to improve the routing performance of delivery ratio and delivery latency, multi-copy protocols distribute multiple copies for each message to the network. Since multi-copy routing protocols can be further divided into flooding-based and quota-based, we analyze their characteristics respectively.

Flooding-based routing protocols, such as Epidemic [117], achieve the optimal routing performance in terms of delivery ratio and delivery latency at the

Table 2.3: Comparison of routing protocols in MDTNs

Protocol	Number of Message Copies	Forwarding Decision	Year
Epidemic [117]	multi-copy	Knowledge-free	2000
Direct Delivery [38]	single-copy	Knowledge-free	2002
First Contact [38]	single-copy	Knowledge-free	2002
Two-Hop Forwarding [38]	multi-copy	Knowledge-free	2002
PRoPHET [70]	multi-copy	Encounter Probability	2003
SW [111]	multi-copy	Knowledge-free	2005
Gossiping [41]	multi-copy	Knowledge-free	2006
MaxProp [13]	multi-copy	Encounter Probability	2006
SF [113]	multi-copy	utility-based	2007
LABEL [46]	multi-copy	Community	2007
(p, q)-Epidemic [79]	multi-copy	Knowledge-free	2008
RANK [47]	multi-copy	Centrality	2008
Seek and Focus [115]	single-copy	utility-based	2008
EBR [89]	multi-copy	Centrality	2009
Habit [78]	single-copy	Regularity	2009
SimBet [24]	multi-copy	Centrality & Community	2009
Clustering [25]	multi-copy	Community	2010
MSW [12]	multi-copy	Knowledge-free	2010
SBR [29]	multi-copy	Centrality	2011
BUBBLE [48]	multi-copy	Centrality & Community	2011
IFR [66]	multi-copy	Centrality & Community	2011
3R [119]	multi-copy	Regularity	2011
SAURP [28]	multi-copy	Encounter Probability	2012
FBR [11]	single-copy	Regularity & Community	2012

cost of huge resource consumption. Unfortunately, the resources (e.g., bandwidth and memory) in mobile devices are traditionally limited and scarce. In addition, a massively redundant messages can cause network congestion, which in turn can severely degrade the routing performance [3, 100].

Quota-based routing protocols [89, 111] can get rid of network congestion by restricting the number of message copies for each message. Specially, the same amount of message copies is allocated for all messages in these protocols. However, allocating the same amount of message copies is a two-edged sword for quota-based routing protocols. Besides the number of copies for a message, the successful delivery of a message is also related to its TTL. It is obvious that a message with a short TTL needs more copies than that with a long TTL to ensure the successful delivery. Therefore, quota-based routing protocols generally suffer from dynamically and reasonably allocating message copies [31].

Knowledge-free routing protocols, such as Epidemic [117] and Spray-and-Wait [111], are characterized by their simple routing processes. Thus, the routing performance of these protocols are easily modeled by mathematics models, such as Markov Chain and Ordinary Differential Equations. However, these protocols assume that all nodes have the same capability of visiting every region covered by the network. However, reality is different. Recent studies [10, 53, 45] on the spatial characteristics of human mobility from real world traces demonstrate that humans usually roam in some relatively small regions rather than the whole space. As a result, knowledge-free routing protocols generally suffer from poor routing performance in real-life scenarios.

Knowledge-based routing protocols exploit the knowledge derived from the history encounters of nodes, such as mobility pattern and social properties of nodes, to better choose the intermediary nodes. Owing to the utilization of such knowledge, knowledge-based routing protocols achieve a better routing performance in terms of delivery ratio or delivery cost, compared with knowledge-free routing protocols. However, existing knowledge-based routing protocols, such as PRoPHET [70] and BUBBLE [48], depends on the assumption that the utilized social property is the most prominent one for all the nodes in a network. However, the most prominent social property may vary during the message dissemination process.

2.5 Privacy Preserving Protocols in MDTNs

In this section, we focus and classify privacy preserving protocols for MDTN routing according to their specific privacy objectives.

2.5.1 Classification of Privacy Objectives

As mentioned in Chapter 1.2, messages in MDTNs are relayed by intermediary nodes. Apart from selfishness, mobile device carriers can be unwilling to participate in the routing process due to their concern about privacy. Recent years have seen considerable research works addressing the issues of privacy in MDTNs. The protocols in the literature are mainly concerned with preserving the privacy of one or more of the following sensitive user aspects: (1) identity, (2) location, (3) message content, and (4) relationships. We can thus classify the existing privacy preserving protocols according to their privacy objectives. Please refer to Figure 2.5 for an illustration of this classification. We discuss each of these privacy objectives in the following section along with some solutions proposed in the literature for achieving these objectives.

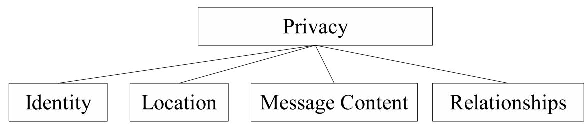


Figure 2.5: Categorition of privacy objectives

2.5.2 Strategies for Preserving Privacy

Identity privacy

In the category of identity privacy, the identity of nodes participating in message delivery is considered as a private information.

Kate et al. [55] presented an anonymous communication architecture for DTNs using Identity-Based Cryptography (IBC) [107]. This is one of the first anonymous communication solutions specifically for DTNs. Kate et al. use a construct called DTN gateways, which are entities assumed to be trusted and to be aware of user identities. In the routing process, a DTN gateway replaces the identity of a source node with a pseudonym unlinkable to the identity. The advantage of the protocol is that there is not much overhead for routing. However, the protocol relies on the assumption that trusted DTN gateways are present, which is a strong assumption for MDTNs.

Le et al. [62] proposed a privacy preserving infrastructure called Privacy-Enhanced Opportunistic Networks (PEON) based on onion routing [102]. In PEON, nodes are clustered into groups. Nodes in the same group share public keys. Before sending a message, a source node determines the routing path, which contains a certain number of node groups. The message is then encrypted by the public keys of the destination node and the determined groups in an inverse order. Thus, each relay node can only be aware of the next hop (i.e., a node group) in the routing path and remains unaware of the identity of the source node. Compared to classic onion routing, the routing performance of PEON in terms of delivery ratio and delivery latency is enhanced due to the utilization of multicasting inside a group. However, node groups are randomly clustered, which may result in the inefficient dissemination of messages inside a group. In addition, the assumption of a Public Key Infrastructure (PKI) rarely holds in MDTNs [55].

Lu et al. [74] presented a social-based privacy-preserving packet forwarding protocol (named SPRING) for Vehicular DTNs. In SPRING, Road Side Units (RSUs) are assumed to be trusted and uncompromisable. Similar to [122], RSUs are strategically deployed at some highly-social intersections to temporarily buffer the messages as relays. Due to the utilization of RSUs, an adversary cannot find out the identity of the source and the destination nodes. However, the private information of nodes is disclosed, if any RSU in the network is compromised. Additionally, all RSUs in SPRING are managed by a single management authority, which results in inflexibility.

Location privacy

In the category of location privacy in MDTNs, the discovery of the user location by the adversary is considered as the main privacy threat. In an untrusted network, the mobile device owners do not want others to know their positions for personal security reasons [75].

In [75], Lu et al. proposed the Anti-Localization Anonymous Routing (ALAR) protocol for MDTNs. In ALAR, each message is divided into k segments and each segment is then encrypted and sent to n different neighbors. Therefore, an adversary may receive several copies of a segment at different times from different relay nodes. Even if the adversary collects these segments, they cannot localize the source node with high probability. The disadvantage is that the routing performance is influenced by the setting of the parameters k and n. Specifically, the routing performance in terms of delivery ratio and delivery latency is degraded as the two parameters increase.

Zakhary and Radenkovic [124] presented a location privacy protocol that is based on the utilization of social information of nodes. In this protocol, each node maintains a social profile, which includes n profile attributes. The social relationship between nodes are inferred by the matching of profile attributes. For each message, the forwarding is guided by the obfuscated attributes in the first k hops. After that, the message can be routed by any routing protocols. Therefore, an adversary cannot distinguish the location of the source node from the other k relay nodes. However, nodes that have strong social relationships are generally considered to be frequently co-located. Thus, the adversary can still detect the approximate location of the source node. Moreover, the routing performance is degraded, due to the extra k forwarding hops.

Message content privacy

Since messages are relayed by intermediary nodes in MDTNs, the content of messages can be unintentionally disclosed to these nodes in the routing process. Thus, in the category of message content privacy, the content of messages is considered as private information.

Jansen and Beverly [51] proposed a Threshold Pivot Scheme (TPS) based on the technique of secret sharing [92]. In TPS, a message, considered as the secret, is divided into multiple shares by the technique of secret sharing. The shares are delivered to the destination node via multiple independent paths. The content of a message is thus protected from individual intermediary nodes. At the destination node, the message can be reconstructed by the knowledge of any τ shares. The disadvantage of this protocol is that if an adversary successes n monitoring a sybil attack, it can create multiple pseudonymous nodes and then intercept sufficient number of shares.

Shi and Luo [109] proposed an anonymous communication mechanism called ARDEN based on onion routing [102], multicast dissemination and Attribute-Based Encryption (ABE) [36]. In ARDEN, before sending a message, the source node determines a path of disjoint groups, one of which includes the destination node. The message is then encrypted by the keys of the destination node and the grouping keys. Compared with the traditional onion routing, the advantage of ARDEN is that it encrypts messages with the keys of groups rather than the keys of individual intermediate nodes. The performance in terms of delivery ratio and delivery latency can be improved, since all nodes in the same group can participate in message forwarding. On the other hand, the arbitrary group partitioning manner may result in performance degradation in terms of delivery ratio and delivery latency.

Relationships privacy

As mentioned in the introduction, the mobility pattern of nodes plays an important role in the routing process. A number of proposed routing protocols exploit the encounter probability [25, 70] and social relationship of nodes [25, 48] to guide the message forwarding decision. However, such information is considered as personal and private [96] thus users may hesitate in participating in such protocols.

Hasan et al. [42] proposed a Privacy Preserving Prediction-based Routing (3PR) protocol for MDTNs. A prediction-based routing protocol for MDTNs works by forwarding a message from one intermediate node to another if the latter has higher probability of encountering the destination node. However, this process compromises the privacy of the nodes by revealing their mobility patterns. 3PR forwards messages by comparing information about communities of nodes instead of individual nodes. Specifically, it compares the maximum probability that a node in the community of a potential intermediate node will encounter the destination node. Simulations on a community-based mobility model demonstrate that the protocol has comparable performance to existing prediction-based protocols.

Parris and Henderson [96] presented the Privacy-enhanced Social-network Routing protocol. This protocol takes advantage of obfuscated social information rather than accurate social information to guide the message forwarding. The original social information of a node is obfuscated by the following two approaches: (1) modifying the friend list, i.e., adding or removing some items into or from the friend list, or (2) using a Bloom filter [8] to hash the friend list. The advantage of the protocol is that the presence of a public key infrastructure is not necessary. However, message routing may be guided erroneously due to the utilization of obfuscated social information. Moreover, in the case of modifying the friend list of a source node, an adversary can approximately determine the source node's friends by collecting the messages from the source node. In the second approach, the probability of false positives increases as the Bloom filter becomes more full, due to the characteristics of Bloom filter.

2.5.3 Analysis of Strategies for Preserving Privacy

Criteria for comparison

The criteria for comparison of the above privacy preserving protocols are described in the following sections.

Adversarial models

We identify two adversarial models, which characterize the behavior of dishonest users. The models are: Semi-Honest, and Malicious. A privacy preserving protocol is considered secure under one of these models if it can show correctness and meet its privacy requirements under the given model.

Semi-Honest. In the semi-honest model, the users do not deviate from the specified protocol. In other words, they always execute the protocol according to the specifications. The adversary abstains from wiretapping and tampering of the communication channels. However, within these constraints, the adversary passively attempts to learn the inputs of honest users by using intermediate information received during the protocol and any other information that it can gain through other legitimate means.

Malicious. Malicious users are not bound to conform to the protocol. Users under a malicious model may deviate from the protocol as and when they deem necessary. They actively attempt to achieve their objectives. They may participate in extra-protocol activities, devise sophisticated strategies, and exhibit arbitrary behavior. A malicious adversary has the following objectives: 1) learn the inputs of honest users, and 2) disrupt the protocol for honest users. The reasons for disrupting the protocol may range from gaining illegitimate advantage over honest users to completely denying the service of the protocol to honest users.

Collusion

A dishonest user may act alone or multiple dishonest users may act in agreement to achieve their ulterior motives. When multiple dishonest users work together, it is referred to as collusion. Privacy preserving protocols either consider that collusion can take place between users or consider that collusion does not take place.

Security building blocks

The privacy preserving protocols for MDTN routing are generally built using security building blocks such as Identity-Based Cryptography (IBC) [107], Public Key Infrastructure (PKI), Onion routing [102], Secret Sharing, Attribute-Based Encryption (ABE) [36], and Bloom filter [8].

2.5.4 Classification of Strategies for Preserving Privacy

A classification of the above privacy preserving strategies is given in Table 2.4 according to the established criteria.

Table 2.4: Comparison of strategies for preserving privacy

Protocol	Privacy Objective	Collusion	Attack Model	Building Blocks
[55]	Identity	Group	Semi-Host	IBC
PEON [62]	Identity Content	Group	Semi-Host	Onion Routing PKI
SPRING [74]	Identity	Group	Semi-Host	
ALAR [75]	Location	Individual	Semi-Host	Secret Sharing
[124]	Location	Individual	Semi-Host	
TPS [51]	Content	Individual	Semi-Host	Secret Sharing
ARDEN [109]	Identity Content	Group	Semi-Host	Onion Routing ABE
3PR [42]	Relationships	Group	Semi-Host	Secret Sharing
[96]	Relationships	Individual	Semi-Host	Bloom Filter

2.6 Chapter Review

In this chapter, we first introduced the wireless technologies used in mobile wireless networks and two major forms of mobile wireless networks. We then introduced two types of mobility models that are widely utilized in the evaluation. Afterwards, we proposed a classification of the existing routing protocols, briefly introduced the state of the art routing protocols, and summarized the characteristics of the routing protocols in each category according to the above classification.

Lastly, we investigated the privacy issues in mobile delay tolerant network. More specifically, we proposed a classification of the privacy objectives and introduced the strategies for each of the privacy objectives. We then classified the existing preserving privacy strategy, according to several building blocks.

Chapter 3

An Adaptive Routing Protocol for MDTNs

3.1 Introduction

As discussed in Chapter 1.3.1, various studies that collect and analyze real mobility traces have been conducted [35, 83, 10], in order to better understand human mobility. These studies have shown that: (1) human mobility is influenced by social relationships; (2) such social relationships are relatively stable over time. According to these observations, a number of social properties characterizing the mobility patterns of nodes have thus been defined. Among these properties, *centrality* characterizes the frequency of encounters that a node has; *regularity* characterizes the probability that given encounters are repeated over time; and *community* characterizes a group of nodes that encounter frequently and regularly [128, 119].

Building on these social properties, a number of routing algorithms [47, 119, 78, 46, 48] have been proposed in the literature. However, most of these routing protocols assume that all messages can be routed by relying on a single mobility property (e.g., the belonging or not to communities). This assumption is very limiting, since in reality users can exhibit various social behaviors and consequently, various mobility properties, in a given environment. For instance, a node may have a central position in the network or be completely isolated in given time slots, it may exhibit a regular or a completely irregular mobility pattern during specific times of the day/specific days of the week, and it can be in/out a community during specific periods of time. In other words, the exiting approaches cannot accommodate the dynamics of human social properties.

In order to accommodate this dynamics, we present the first Adaptive Routing

Protocol (ARP) for MDTNs that dynamically adapts to the user's social properties at the very specific time and location. To this end, our protocol locally monitors the contacts between nodes. When two nodes encounter each other, each of them utilizes the protocol to calculate the utility values of delivering a given message to the destination node, based on different social properties. The forwarding decision is made based on a comparison of the computed utility.

The remainder of this chapter is organized as follows. In Section 3.2, we introduce the system model and the information maintained by each node. In Section 3.4, we describe our proposal in detail. The simulations and results are presented in Section 3.5. Finally, we conclude this chapter in Section 3.6.

3.2 System Model

We consider a set $\mathbb{A}$ of N mobile devices which can freely roam in a physical environment. Each mobile device is denoted as a node with a unique identifier. We assume that each node is equipped with a short-range radio interface (e.g., Bluetooth or WiFi) for communication. For the sake of simplicity, we make the same assumption as in [12, 114, 69, 68] that the transmission range of all nodes is the same. Nodes are said to encounter (or meet, contact) when they come into the transmission range of each other. Two encountering nodes can directly exchange messages (i.e., communication) with each other.

We define a community C as a set of nodes which frequently co-exist and meet in a common space. Thus, the meeting frequency of nodes in the same community is considered to be much higher than that of nodes in different communities. We assume that each community has a unique identifier. The set of nodes in a community C will be indicated hereafter by $C = \{a_1, a_2, \ldots, a_n\}$, where $n = |C|$.

3.3 Overview of ARP

In this section, we first present the architecture of ARP. We then explain how it works.

3.3.1 The Architecture of ARP

Our aim is to improve the routing performance by dynamically adapting the protocol to the social properties of the nodes at the very specific time and location. To enable this kind of self-adaptive behavior, we need a protocol that supports: (1) the analysis of the social properties of nodes, (2) the capability of comparing different

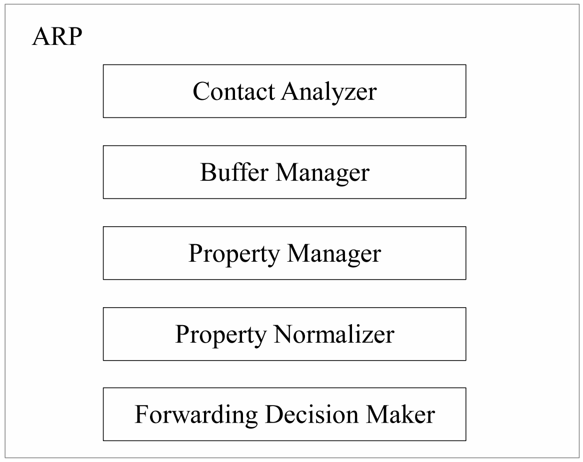


Figure 3.1: Protocol Architecture

social properties, and (3) the management of the priority of the messages that are to be forwarded.

The architecture of ARP is illustrated in Figure 3.1. In the following, we introduce the components of ARP and their functions.

- **Contact Analyzer**: It records its host's contacts with other nodes. Based on the contact history, the contact analyzer can infer the social properties of its host (e.g., centrality, regularity, and community).

- **Buffer Manager**: It stores and manages all the messages that are taken by its host.

- **Property Manager**: It stores the property values of the last k nodes that its host encountered.

- **Property Normalizer**: It utilizes a utility value, which is a real number in $[0,1]$, to represent the value of a given social property. Thus, the utility value of a property indicates its host's delivery capability through the property.

- **Forwarding Decision Maker**: Forwarding decision maker decides to forward a message to an encountering node, based on the utility values of the two encountering nodes. For all the messages which the encountering node has a better delivery capability, forwarding decision maker sorts them by the delivery capability in a decreasing order.

3.3.2 The Workflow of ARP

In this section, we use an example (See Figure 3.2) to demonstrate how ARP works. Without loss of generality, let's consider two encountering nodes named a_1 and a_2. The example is described from the viewpoint of node a_1 in the following steps:

1. After neighbor discovery service of the radio interfaces, node a_1 can identify when a_2 encounters it; it then adds the contact with node a_2 into the contact analyzer.

2. Node a_1 searches its buffer and selects all messages whose destination is node a_2. Node a_1 then sends these messages to node a_2.

3. Node a_1 sends its centrality value, regularity table and community ID to node a_2.

4. Assume that node a_1 still holds other messages whose destination node is not node a_2. Node a_1 then sends a summary list of these messages in the format of $\langle$ message ID, destination node ID, expiration time $\rangle$ to node a_2.

5. Using this information for the messages in the summary list, node a_2 computes the property values of itself and the last k encountered nodes for each message in the summary list, and returns them to node a_1 in the format of $\langle$ message ID, node ID, centrality value, regularity value, community value $\rangle$. Note that the community value is 1, if a_2 is in the same community as the destination node; otherwise, it is 0.

6. For each message in the summary list, node a_1 merge all the property values returned from node a_2 with its last k encounters' property values. Node a_1 then normalizes the property values of nodes a_1 and a_2.

7. Based on the normalized property values, node a_1 prepares a list of messages for which node a_2 has a higher deliver capability. Node a_1 then sorts the prepared message list in decreasing order of the delivery capability of node a_2. Afterwards, node a_1 forwards the messages in the prepared list to node a_2.

Without loss of generality, in steps 3-7, we only describe how node a_1 determines which messages should be forwarded to node a_2. Actually, node a_2 also performs similar steps.

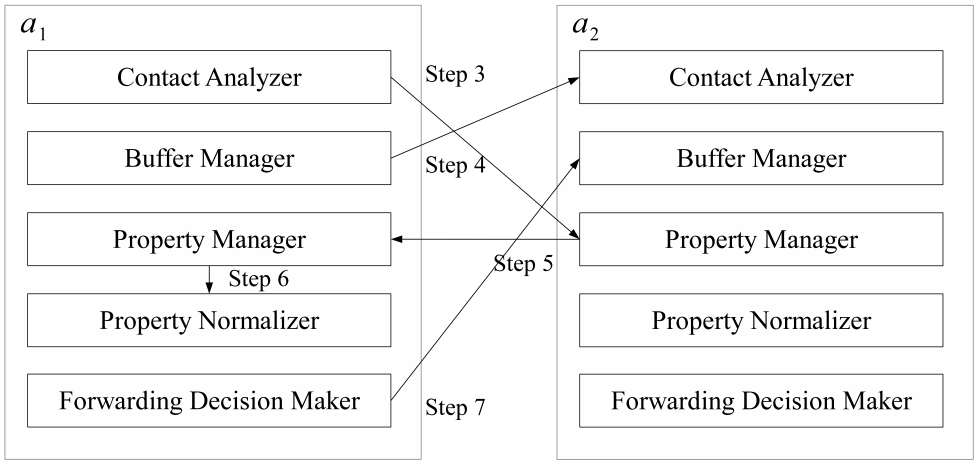


Figure 3.2: An example of the workflow

3.4 Detailed Design of ARP

In this section, we describe the computation of the property values, and the normalization function.

3.4.1 Calculation of Centrality

As mentioned in the introduction, centrality is a metric that calculates the relative importance of a node in a network. Hui et al. in [48] presented that the centrality of nodes in real-life environments is heterogeneously distributed. Moreover, Hui et al. also pointed out that popular nodes (i.e., nodes with high centrality values) instead of unpopular nodes should be utilized as relays in the design of efficient routing strategies.

In the literature, a number of centrality metrics have been proposed, such as betweenness centrality [32], closeness centrality [105], degree centrality [90]. These centrality metrics are based on graph theory. However, these metrics are not suitable for a distributed environment, such as DTNs. Hui et al. in [48] presented that the degree of a node per time slot (e.g., the number of unique nodes encountered by the node in per 6 hours) is able to approximate the pre-calculated centrality (e.g., betweenness centrality) quite well. We thus utilize this approach [48] to calculate the centrality of nodes.

In our protocol, each node (e.g., node u) holds an encounter set denoted as ξ, which records the nodes encountered by node u in a given time slot. The encounter between node u and a given node v in time slot i is expressed as Equation (3.1).

$$\zeta_{uv}^{i} = \begin{cases} 1, & \text{if } u \text{ and } v \text{ meet in time slot } i \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

Let ξ' be the set of nodes encountered by node u in the last time slot. The number of unique nodes encountered by node u in the last time slot is then presented as Equation (3.2). The superscript *cen* indicates the centrality property.

$$\Delta P_{u}^{cen} = \sum_{v \in \xi'} \zeta_{uv}^{'} \tag{3.2}$$

Generally, there are two ways to calculate the centrality of nodes based on time slot: S-Window and C-Window. In S-Window manner, the centrality of a node is the number of unique nodes that it met in the previous time slot. Meanwhile, in C-Window manner, the centrality of a node is the accumulation of the number of unique nodes that it met in each of the previous windows. Let $P_{u}^{cen'}$ be the centrality value of node before the previous time slot. The centrality of node u is then expressed as Equation (3.3).

$$P_{u}^{cen} = \begin{cases} \Delta P_{u}^{cen}, & \text{S-Window} \\ P_{u}^{cen'} + \Delta P_{u}^{cen}, & \text{C-Window} \end{cases} \tag{3.3}$$

3.4.2 Calculation of Regularity

As mentioned in the introduction, the regularity of two nodes is defined as the probability that the two nodes meet each other in a given time slot over a given time length. For example, time slots can be considered as 1 hour intervals and the time length can be considered as a week. When the size of the time slot equals to the time length, the regularity of two nodes in a time slot becomes to the general encounter probability, which is widely utilized in DTNs [25, 70]. Assume the duration of the contact history is 10 weeks. In the contact history, node A has met node B for 7 times in the time slot from Mon. 0 AM to Mon. 1 AM. In this case, the regularity between node A and node B from Mon. 0 AM to Mon. 1 AM is 0.7. Each node contains a regularity table that describes the regularity between it and its fellow nodes in given time slots. The regularity table is constructed by tuples which contain the time slot and the regularity. The number of such tuples is the ratio of the time length (e.g., a week, a month) divided by the size of the time slot. An example of regularity table of a node is illustrated in Table 3.1.

The utilized calculation of regularity between nodes is similar to [119]. Let p_{uv}^{i} be the regularity between two nodes u and v in a given time slot i, that is the probability that nodes u and v encounter each other in time slot i. The probability

Table 3.1: An example of the regularity table of node A; time slot is 1 hour, and time length is a week

Time slot	B	C	D	E
Mon.[0 AM, 1 AM)	0.7	0.6	0.5	0.3
Mon.[1 AM, 2 PM)	0.1	0.2	0.6	0.4
...	...	...	...	...
Sun.[11 PM, 0 AM)	0.1	0.2	0.6	0.4

that nodes u and v encounter each other in the k^{th} time slot and not in the first $k-1$ time slots can be presented as Equation (3.4).

$$P_{uv}^{k} = \{\prod_{i=1}^{k-1}(1-p_{uv}^{i})\}p_{uv}^{k} \tag{3.4}$$

Consider a node u and a message whose destination is node d and the remaining TTL is T. Let $\psi(t)$ be the time slot to which a given time t belongs. Let t' indicate the current time. Since node u may encounter node d in each of the time slots from the current time slot $\psi(t')$ to the time slot $\psi(t'+T)$ in which the TTL of the message expires, we should accumulate the encounter probability between node u and d in each of these time slots. The probability that node u will encounter node d before the TTL of the message expires can thus be expressed as Equation (3.5). The superscript *reg* indicates the regularity property.

$$P_{ud}^{reg} = \sum_{i=\psi(t')}^{\psi(t'+T)} P_{ud}^{i} \tag{3.5}$$

3.4.3 Community Calculation

Studies utilizing Social Networks Analysis (SNA) on human mobility from real world traces have proved that people do not act alone in real life, but tend to belong to communities [48, 9, 49]. Since human mobility is driven by social relationships [2, 10], nodes which frequently co-exist in a common location are generally considered to consist of a community [48, 46, 9, 25]. In addition, since the encounter frequency of nodes in the same community is generally considered to be much higher than that of nodes in different communities, community structure is widely utilized in the design of routing protocols [48, 9, 25]. Furthermore, these works also proved that the utilization of the community can improve the routing performance in terms of delivery ratio and delivery cost [48, 9, 46].

Hui and Crowcroft in [49] introduced some distributed community detection methods for DTNs, such as SIMPLE, k-CLIQUE and MODULARITY. The description of these algorithms is out of the scope of this chapter; the reader is suggested to refer to [49] for further study of community detection. Hui and Crowcroft also pointed out that k-CLIQUE is a good choice compared with SIMPLE and MODULARITY, according to the computational complexity and accuracy. Moreover, Palla et al. in [94] demonstrated that the centralized k-CLIQUE shows stable results for different types of human contact traces. Thus, in this chapter, we utilize the distributed k-CLIQUE algorithm to detect communities.

Without loss of generality, consider a node u and a community C. The relationship between node u and community C is then expressed in Equation (3.6). The superscript *com* indicates the community property.

$$P_{u,C}^{com} = \begin{cases} 1, & \text{if } u \in C \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

3.4.4 Normalization Function

The goal of the work in this chapter is to efficiently utilize the social properties of nodes to steer the message forwarding towards the destination node as fast as possible. However, the routing decision is made by comparing social properties with different meaning and domain values. Thus, we should use the relative values of social properties rather than the absolute values of these properties. To this end, the value of social properties are normalized by a normalization function.

Let $\mathscr{N}_u$ be the set of nodes which are utilized to normalize the property values of a node u. When node u encounters a node v, the set (denoted as $\mathscr{N}$) of nodes utilized to normalize their property values is updated as follows.

$$\mathscr{N} = \mathscr{N}_u \cup \mathscr{N}_v \tag{3.7}$$

Let p_u be the value of a property (i.e., centrality or regularity) p of a node u. Let the sample set of the property values of nodes in $\mathscr{N}$ be $\Omega_p = \{p_i | 1 \leq i \leq n\}$, where $n = |\mathscr{N}|$. Let $m(p)$ and $\Delta(p)$ be the mean value and standard deviation of the sample set Ω_p of property p, respectively. Let p'_u be the normalized value of the property p of node u. The normalization function is then expressed in Equation 3.8 [4].

$$p'_u = \begin{cases} 1, & \text{if } p_u - m(p) > 2\Delta(p) \\ 0, & \text{if } p_u - m(p) < -2\Delta(p) \\ 0.5 + \frac{p_u - m(p)}{4\Delta(p)}, & \text{otherwise} \end{cases} \tag{3.8}$$

The formalized property values vary in $[0,1]$, which presents a node's relative capability of delivering a given message among a set of nodes. It is called as centrality utility, regularity utility, etc. A message is then transferred from a node with a lower utility to a node with a higher utility value.

3.4.5 Decision of Message Forwarding

In this section, we first present the design space of the forwarding decision marker. We then present how the forwarding decision marker works.

Design Space of Forwarding Decision Marker

The decision of message forwarding in ARP is mainly based on a node's properties values which indicate the node's capability of delivering a message to the destination node. The forwarding decision is made based on the maximum difference between the properties values of two encountering nodes.

Figure 3.3 shows the design space of the forwarding decision maker in ARP. The vertical axis represents the regularity of nodes, which indicates the mobility pattern of nodes. The two horizontal axes represent the social information. More specifically, the community structure indicates the relationship between nodes, while the centrality indicates the node's importance in the network.

The forwarding decision maker in ARP is a combination of RANK [47], 3R [119] and LABEL [46], each of which uses a single social property of centrality, regularity and community to guide the message forwarding. It utilizes LABEL to identify the community of a destination node if the destination node belongs to a community. It uses RANK and 3R to forward a message closer and closer to (the community of) the destination node.

Our protocol is inspired by the protocol presented by Hui et al. [48]. However, our protocol does not depend the assumption made in [48] that each node belongs to at least one community. Our protocol thus can suit to more environments. Moreover, compared with the centrality property, the regularity property is more accurate to indicate the contacts between a given pair of two nodes. Thus, our protocol can improve the routing performance of the delivery ratio.

Design of Forwarding Decision Marker

A routing example is depicted in Figure 3.4. This figure shows that a source node s wants to send a message to a node d that belongs to the community C. Suppose the distance between two nodes indicates the encounter frequency of them. That is, the encounter frequency of two nodes which are far from each other is lower than

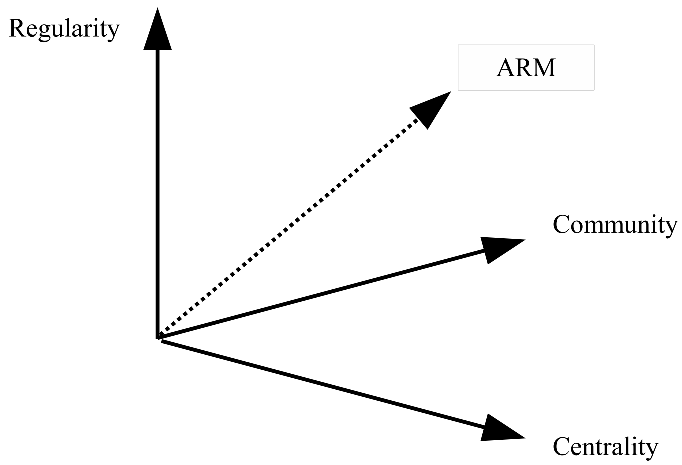


Figure 3.3: Design space of forwarding decision maker in ARP

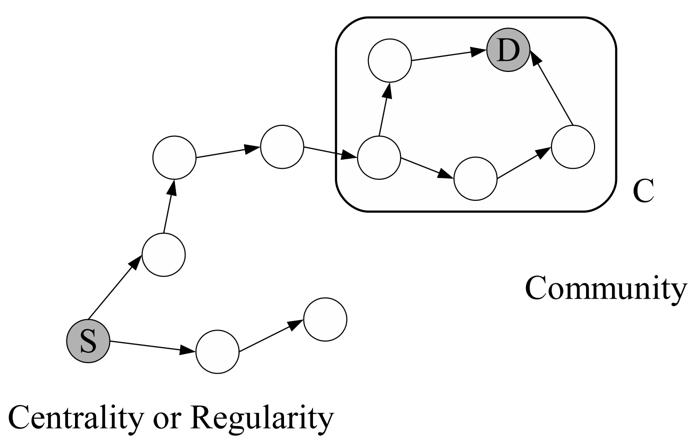


Figure 3.4: Illustration of the forwarding decision maker in ARP

that of two nodes which are near to each other. Consequently, when the message is at the node which is far from the destination node, the value of the regularity property between the node and the destination node is low. Thus, in such a case, the message is forwarded to popular nodes, which have high centrality values. When the message is close to the destination node, the regularity property becomes the prominent property. The message is then forwarded to the nodes which have a high regularity property with the destination node, until the message reaches a node that belongs to the same community as the destination node. After entering the destination node's community, the forwarding of the message is guided by the property of centrality or regularity inside the community, since the community property is not the prominent property among the community members. The forwarding process is repeated until the message reaches the destination node or the TTL of the message expires. The pseudo code of the forwarding strategy is illustrated in Figure 3.5.

3.5 Performance Evaluation

In this section, we present an evaluation of our adaptive routing protocol by means of simulations. Specifically, we first introduce the data set utilized to conduct the simulations. We then present the simulation settings and the performance metrics. After that, we briefly describe the compared routing protocols. Finally, we compare the performance of our proposal with the compared routing protocols.

3.5.1 Dataset

The dataset that we utilized to evaluate our protocol is the MIT Reality Mining dataset [27]. The contacts between nodes (i.e., students) are illustrated in Figure 3.6. It can be seen that these contacts concentrate in two periods: from the middle of September 2004 to the middle of December 2004 and from the middle of January 2005 to the middle of May 2005. The two periods are the academic semesters, the contacts between nodes are thus relatively stable. In addition, the contacts in the first period are much more than that in the second period. We thus selected the period from Sept. 13 2004 to Dec. 06 2004, which contains $\frac{49183}{90534} \approx 54.33\%$ contacts of the whole dataset, to conduct the experiments.

3.5.2 Simulation Settings

The simulations have been conducted on the Opportunistic Network Environment (ONE) simulator [57] with the MIT Reality Mining dataset refined as explained afore. The dataset contains the contacts of nodes over 12 weeks. The first 4 weeks

Protocol: MessageForwarding
Participants: Node u and node v, where $u, v \in \mathbb{A}$.
Input: (1) m, a message. (2) d, the destination node of message m. (3) C_d, the community of node d. (4) ε, the threshold of maximum difference of properties.
Notation: (1) P^u_{cen}, the centrality value of node u. (2) P^v_{cen}. (3) P^u_{reg}, the regularity value of node u. (4) P^v_{reg}. (5) P^u_{com}, the community value of node u. (6) P^v_{com}.
Output: Message m is forwarded to node v if $v = d$, or node v has a higher capability of delivering the message to the destination node.
Setup: Node u has a message m whose destination is node d. Node v does not have message m.
Events and Associated Actions:

node u **encounters a node** v

if $v = d$ **then**
  node u sends message m to node v
else
  each node calculates and normalizes its property values, according to Equation 3.3,3.5,3.6,3.8
  $P^{diff}_{cen} \leftarrow P^v_{cen} - P^u_{cen}$
  $P^{diff}_{reg} \leftarrow P^v_{reg} - P^u_{reg}$
  $P^{diff}_{com} \leftarrow P^v_{com} - P^u_{com}$
  $P^{max} \leftarrow max\{P^{diff}_{cen}, P^{diff}_{reg}\}$
  if $P^{diff}_{com} \geq 0$ **and** $P^{max} \geq \varepsilon$ **then**
    node u forwards message m to node v
  end if
end if

Figure 3.5: MessageForwarding

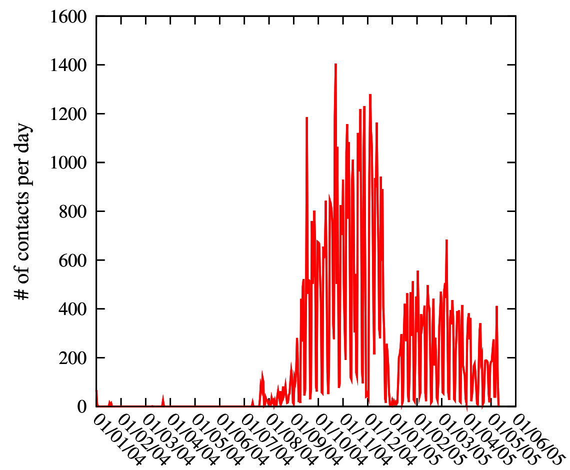


Figure 3.6: Number of contacts per day in the whole MIT real ming data set

are selected as a warm-up period to initialize the properties of the nodes, i.e., centrality, regularity and community. In the following 4 weeks, as in [78], a pair of nodes are randomly selected as the source node and the destination node of a message every hour. Therefore, there are 672 messages generated in each simulation. Each message contains the identifiers of the source and the destination nodes, and a given TTL. The last 4 weeks are utilized to deliver the messages which are last generated.

In addition, we analyze the contacts of nodes, in order to detect the regularity pattern of nodes' contacts. Figure 3.7 shows the number of contacts between nodes per day in the dataset. It can be seen that the contacts of nodes exhibit a high periodicity. That is, there are about 4 periods in each month. We thus map the contacts of nodes into a week map to extract the regularity pattern of nodes.

In order to detect the community structure of nodes, we adopted the same setting as in [48]. Specifically, the community detection is based on the contact duration of each pair of nodes. The threshold for the contact duration is chosen as 129600 seconds in the dataset, i.e., the contact duration of each pair of nodes in a community is greater than the threshold. The threshold was obtained from assuming 3 lectures per week and a total trace duration of 12 weeks. Research students in the same office may stay together all day, so their contact duration threshold could be very large. For students attending lectures, this estimation can be reasonable.

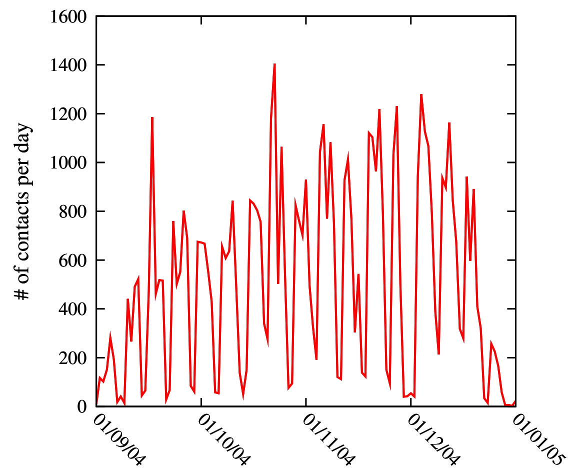


Figure 3.7: Number of contacts per day in the period from Sept. 13, 2004 to Dec. 06, 2005 on MIT Real Mining dataset.

3.5.3 Routing Protocols

Based on the above settings, we conducted our experiment with the following routing protocols.

RANK: In [47], each node maintains a windowed degree centrality value, which indicates how many nodes it can encounter in a given period of time. The centrality of nodes is then employed to guide the forwarding of messages. More specifically, a message is forwarded from a node with a lower centrality value to a node with a higher centrality value until it reaches the destination node or the TTL of the message expires. In the simulation, the C-Window strategy is used to compute the centrality value of a node. It accumulates the number of unique nodes that encountered the node in each of the previous time windows. The window size is set to 4 hours.

3R: In [119], each node maintains a regularity table which records the probability of encountering each of other nodes in different time slots. The regularity table is then utilized to calculate the probability that a node will encounter the destination node within the TTL of a message. Specifically, each message is forwarded from a node to the destination node or the nodes which have a higher encounter probability with the destination node.

Label: In [46], it assumes that each node possesses a label indicating its community. The label of nodes is then used to guide the forwarding of messages. More specifically, each message is forwarded to the nodes with the same label as

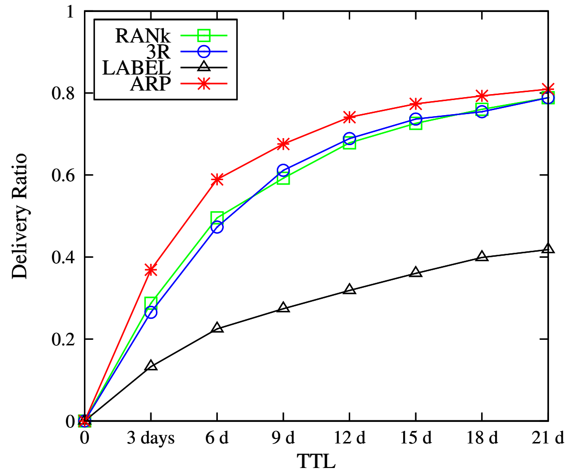


Figure 3.8: Delivery ratio comparison of the routing protocols

the destination node until it reaches the destination node or the TTL of the message expires.

3.5.4 Performance Metrics

To evaluate ARP we used two well known metrics: the delivery ratio and the delivery cost defined as follows.

Delivery ratio: is the proportion of messages that have been delivered out of the total unique messages created.

Delivery cost: is the total number of messages transmitted in the simulation. To normalize this, we divide it by the total number of unique messages created.

3.5.5 Simulation Results

We performed three experiments. First, we compare the performance of ARP against the protocols introduced above, with respect to the two introduced performance metrics. We then analyze the impact of the threshold value on the performance of ARP. Finally, we investigate the impact of the size of the encounter set on the performance of ARP.

Performance comparison of routing protocols

Figure 3.8 shows the delivery ratio of the compared routing protocols. We can observe that ARP always achieve the best delivery ratio for all values of TTL.

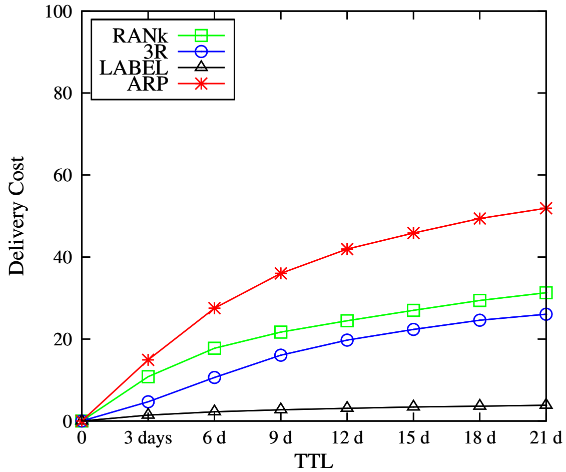


Figure 3.9: Delivery cost comparison of the routing protocols

LABEL always achieve the worst delivery ratio. This is because the messages whose destination nodes do not belong to any community can only be delivered by LABEL when the source nodes directly encounter the destination nodes. We also observe that the delivery ratio of RANK is very close to that of 3R.

Figure 3.9 shows the delivery cost of the compared routing protocols as a function of the TTL of the generated messages. We can observe that ARP has the highest delivery cost, whatever the value of TTL. This is because ARP make multiple properties to make forwarding decision, and the properties of nodes are diversified. LABEL always achieve the lowest delivery cost. This is because the messages whose destination nodes do not belong to any community can only be delivered by LABEL when the source nodes directly encounter the destination nodes. We also observe that the delivery cost of 3R is lower than that of RANK when the TTL is shorter than 1 week, since the regularity property is more accurate to indicate the contacts between a given pair of two nodes.

In this work, we focus on improving the delivery ratio by adapting to the dynamics of social properties. In the future, we would restrict the number of copies for each message to reduce the deliver cost of the protocol.

Demonstrating Routing Protocol Behavior

In this section, we demonstrate the usage of multiple social properties to make the forwarding decisions. To this end, for each property, we record the number of forwarding decisions that are made by using the property. To normalize this, we

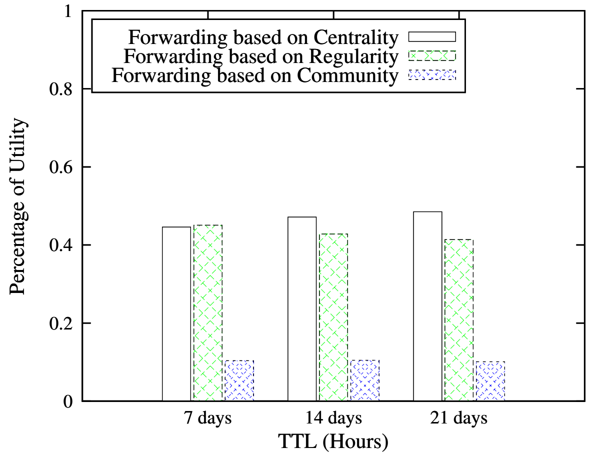


Figure 3.10: Usage of the social properties in making forwarding decisions

divide it by the total number of forwarding decisions in each simulation. In the following simulations, the TTLs of messages are chosen as 7 days, 14 days, and 21 days.

Figure 3.10 shows the percentage of the usage of each property to make the forwarding decisions. It can be seen that most of the forwarding decisions are made based on the centrality and regularity properties. It is because that the community property only works when a message carrier encounters a node belonging to the same community as the destination node of the message. Moreover, the percentage of the usage of centrality property increases as the TTL increases, while the percentage of the usage of regularity property decreases as the TTL increases. This is because the centrality property in this dataset is more heterogeneous than the regularity property.

Impact of the Threshold Value

In this section, we investigate the impact of the settings of the threshold on the routing performance of ARP. In the following experiments, we vary the value of the threshold from 0.1 to 0.3 with step by 0.1.

Figure 3.11 shows the delivery ratio of ARP with different threshold values. We can observe that ARP achieves a higher delivery ratio with a lower threshold value when the TTL is less then 12 days. This is because ARP can forwards more copies of a message to encountered nodes with a lower threshold. The more nodes taking a copy of a message can in turn result in increasing the delivery probability of the message. When the TTL is greater than 12 days, ARP achieves almost the

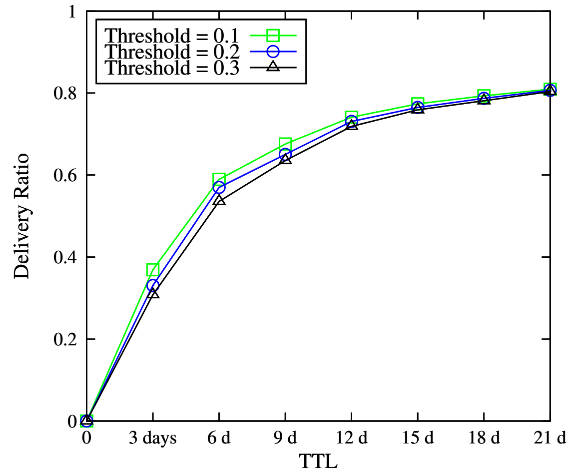


Figure 3.11: Delivery ratio comparison of the routing protocols

same delivery ratio with different threshold values.

Figure 3.12 shows the delivery cost of ARP with different threshold values. It can be seen that the delivery cost increases as the TTL increases. This is because message copies of a message can be forwarded to more nodes with a longer TTL. Moreover, we can observe that the delivery cost increase as the threshold value decreases. This is because ARP can forwards more copies of a message to encountered nodes with a lower threshold.

Impact of the Normalization Function

In this section, we investigate the impact of the size of the encounter set, which is utilized to normalize the property values, on the routing performance of ARP. In the following experiments, we vary the value of the size of the encounter set from 5 to 15 with step by 5.

Figure 3.13 (a) and (b) show the delivery ratio and the delivery cost of ARP with different sizes of the encounter set. We can observe that ARP achieves almost the same routing performances with different sizes of the encounter set.

3.6 Chapter Review

In this chapter, we presented a novel adaptive routing protocol for mobile delay tolerant networks, named ARP. ARP can dynamically learn the social properties of nodes based on their mobility patterns, and exploit the most appropriate routing

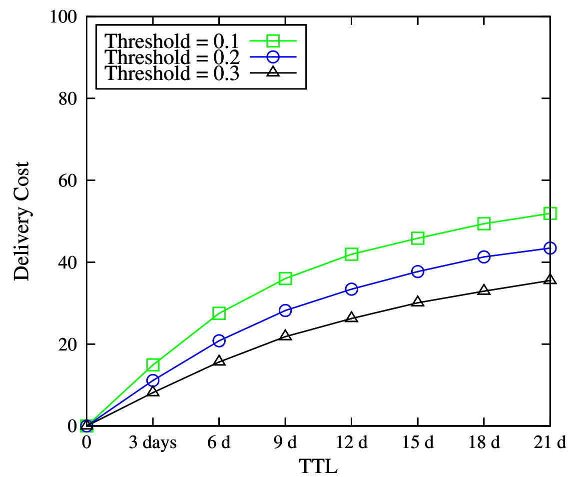
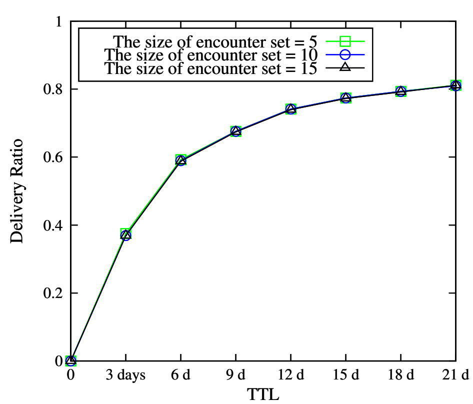
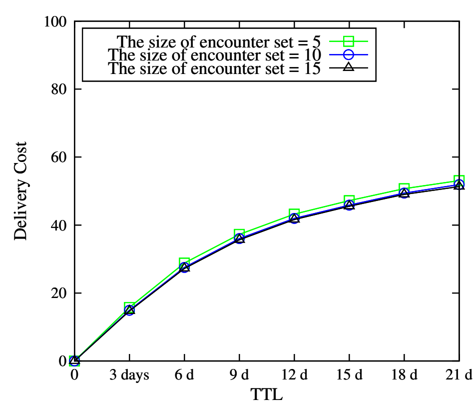


Figure 3.12: Delivery cost comparison of the routing protocols



(a) Delivery Ratio



(b) Delivery Cost

Figure 3.13: The impact of the settings of the encounter set on the routing performance of ARP.

strategy each time an intermediate node is encountered. Our simulations based on a real mobility trace demonstrated that our protocol achieves a better delivery ratio than existing state-of-the-art routing protocols that rely on a single mobility property.

Chapter 4

A Delay and Cost Balancing Protocol for Efficient Routing in Mobile Delay Tolerant Networks

4.1 Introduction

In the literature, a number of routing protocols have been proposed in MDTNs. Nevertheless, most of them are inefficient from the point of view of message replication. For instance, flooding-based routing protocols [117], which refers to those protocols that rely on an unlimited number of message copies to route a message, cause a mass of redundant message copies, which obfuscates the (limited) resources of mobile devices. Quota-based routing protocols [89, 111] instead, allocate the same amount of message copies for messages with different Time-To-Lives (TTLs). The rigidity of the latter approaches makes them often inefficient, as a fixed number of message copies can not suit all the routing situations. As a result, the dynamic allocation of message copies according to the urgency of messages is still an open issue in MDTNs.

To the best of our knowledge, this issue has been addressed only by Bulut et al. in [12]. However, the protocol presented in [12] assumes that all nodes move similarly in all the regions of the network. Unfortunately, reality is different. Recent studies [10, 53, 45] on the spatial characteristics of human mobility from real world traces demonstrate that humans in real-life tend to roam in some relatively small regions rather than the whole network space. In addition, the allocation of message copies in [12] relies on the assumption that the source node is aware of the successful delivery of a message at anytime. However, such assumption rarely holds in reality, due to the frequent and long-term network partitions in MDTNs.

In this chapter, we present a delay and cost balancing routing protocol in MDTNs, called Community-based Adaptive Spray (CAS) routing protocol. The goal of this protocol is to allocate the minimum number of message copies while still achieving a high delivery ratio in a practical manner. Our protocol exploits the community structure of the nodes as it has been demonstrated in the literature that often nodes belong to such structures [48].

Our protocol decomposes into two major parts. First, a sub-protocol responsible for gathering mobility information about nodes upon encountering each other. This sub-protocol aims at learning/synchronizing the topologies of communities in the network. Second, a sub-protocol responsible for the routing process. Routing is organized around the notion of *gateways* between communities. Specifically, a gateway towards a community C is the node in a given community that has the highest probability to encounter any node in C. In our protocol, to route a message towards a given destination node, the source of a message uses the community topology to pre-compute the multi-hop path traversing the minimal number of communities through their gateway nodes and that has the highest delivery probability. Furthermore, once the routing process is engaged, our routing protocol allocates a given number of message copies at each hop according to the remaining TTL of the message.

The contributions of the work in this chapter are twofold:

- We propose a novel routing protocol that dynamically allocates message copies according to the TTL of each message.
- The analysis of our protocol has shown that it is the generalization of many protocols including Direct [115], Epidemic [117], Spray-and-wait [111], and some community based routing protocols [25]. By generalization, we mean that our protocol can dynamically decide to behave like one of these algorithms in order to better suit the current situation.

The remainder of the chapter is organized as follows. In Section 4.2, we introduce the system model and the information maintained by each node. In Section 4.3, we describe our proposal in detail. In Section 4.4, we develop and evaluate an analytical model of our routing protocol, which is utilized to estimate the cover time of the topology of communities. The simulations and results are presented in Section 4.5. Finally, we conclude this chapter in Section 4.6.

4.2 System Model

In this section, we first present the network model. We then introduce the information maintained by each node. Finally, we explain how the information is maintained by each node.

4.2.1 A Delay Tolerant Network Model

In order to make the study more systematic, we start with modeling a delay tolerant network.

Node. We define a node as a mobile device. We assume that each node is equipped with a radio interface (e.g., Bluetooth) for short-range communication, and that the transmission range of all nodes is the same. Two nodes are considered to meet, if they are in the transmission range of each other. We assume that a node can only communicate with one other node at the same time. Two meeting nodes can exchange messages with each other. We also assume that each node has a unique identifier.

Community. We define a community C as a set of nodes which frequently coexist and meet in a common space. Recent studies [10, 84] have indeed shown that human mobility was actually driven by social relationships. Thus, the meeting frequency of nodes in the same community is considered to be much higher than that of nodes in different communities. We also assume that each community has a unique identifier. The set of nodes in a community C will be indicated hereafter by $C = \{u_1, u_2, \ldots, u_n\}$, where $n = |C|$. Moreover, we assume that a node can only belong to one community.

Inter-contact time between two nodes. The inter-contact time (also known as inter-meeting time) between two nodes is the time interval between two successive encounters between them.

Mobility Model. Mobility models are generally characterized by the inter-contact time between two nodes [12, 114]. Karagiannis et al. in [53] demonstrated that under a large class of mobility scenarios in real life, the inter-contact time follows a power-law in a finite range, and then exhibits an exponential decay. It is consistent with the suggestion made by Gonzalez et al. in [35] that a power law with an exponential decay is a very good approximation of human mobility patterns. Additionally, Chaintreau et al. in [20] pointed out that the exponential decay eliminates the issue of infinite message forwarding delay. Building on these previous studies, the inter-contact time of nodes is assumed to be exponentially distributed or have an exponential tail characterized by a contact rate λ (the inverse of the expected inter-contact time of any pair of nodes). This is a widely accepted assumption in MDTNs [12, 114, 69, 37].

Inter-contact time between a node and a community. The inter-contact time between a node u and a community C is the time interval between any two successive contacts between node u and any node member of the community C.
Gateway. A node w in a community C_i is defined as the gateway connecting to another community C_j, if node w's average inter-contact time with the nodes of the community C_j is the shortest among the other nodes of its community C_i.
Network. Let the set of all the nodes in the environment be given as the set V. Let the set of all communities in the environment be $\mathbb{M} = \{C_i | 1 \leq i \leq k, V = \bigcup_{i=1}^{k} C_i\}$, where k is the number of communities in the network. Edges $e_{i,j}$ and $e_{j,i}$ exist between two communities C_i and C_j (where $1 \leq i \leq k$, $1 \leq j \leq k$, $i \neq j$), if some nodes from the two communities have encountered at least once. The weight of the edge $e_{i,j}$, denoted as $\varpi_{i,j}^{w}$, is the average inter-contact time of the gateway w of community C_i with the nodes of the community C_j. Notice that the gateway from community C_i to the community C_j is not the same node as the gateway from C_j to C_i, so the community graph is a directed graph. Let $E = \{e_{i,j} | 1 \leq i \leq k, 1 \leq j \leq k, and i \neq j\}$. We will represent the MDTN formed by all the nodes of the network by their community graph $G(\mathbb{M}, E)$.
Message. A message is represented as a tuple $\langle S, D, I, C, L, T \rangle$, where S is the source node, D is the destination node, I is the intermediate target node, which is the gateway node connecting the current community to another community (or the destination node if the message is yet in the community of the destination node). C is the identifier of the community connected by the gateway node (or null if the intermediate target node is the destination node). L is the number of copies allocated to route the message to the intermediate target node in the current community, T is the message TTL.

4.2.2 Information Maintained by a Node

Each node in the network, e.g., node u, maintains six types of information: its node ID u, its community ID C^u (the exponent here indicates the fact that this is the community to which u belongs), a community table, a gateway table, a contact table, and a community graph.
Community table. The community table of node u holds the node ID and the community ID of all nodes who have encountered the nodes of u's community. Node u also maintains a timestamp which indicates the time when the table was last updated.
Gateway table. The gateway table of node u contains the following fields for each known community C_i and for each gateway w linking C_i to another community C_j: the community ID of C_i, the node ID of the gateway w, the community ID of C_j, and the average inter-contact time between the gateway w and the community C_j

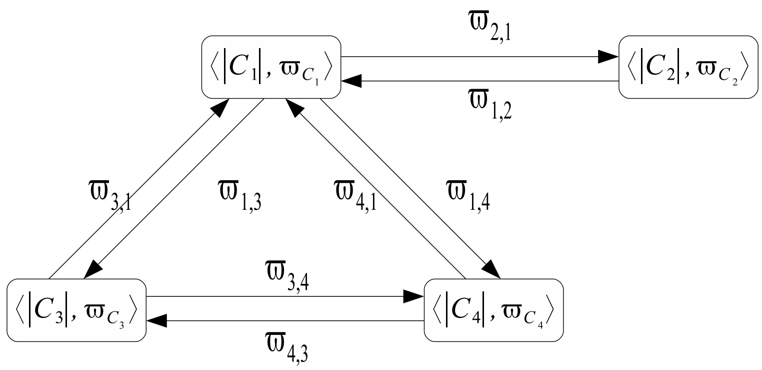


Figure 4.1: Community Graph

given as $\varpi^{w}_{i,j}$. Node u also maintains a timestamp which records the last updated time of this table.

Contact table. The contact table of node u maintains the following fields for each encountered node v: the node ID of v, the sum of inter-contact times between the nodes u and v, denoted as $\tau_{u,v}$, the number of encounters $\sigma_{u,v}$ (the latter two quantities are used to compute the average inter-contact time), the end time of the last encounter $t^e_{u,v}$, and the start time of the ongoing encounter $t^b_{u,v}$ (if there is one).

Community graph. In the community graph of node u, each vertex denotes a known community e.g., community C_i. Each vertex is labeled by a tuple $\langle |C_i|, \varpi_{C_i} \rangle$, where $|C_i|$ is the number of nodes in community C_i, ϖ_{C_i} is the average inter-contact time between the member nodes of the community C_i. The weight of the directed edge e_{ij} denoted by $\varpi_{i,j}$ is the average inter-contact time between the gateway in community C_i and the community C_j. The community graph of node u locally reflects the topology of the communities of the network*as it is known by u*, which is illustrated in Figure 4.1.

4.2.3 Maintenance of Information

The maintenance of the above information is driven by events. There are three kinds of events in the protocol: 1) *connect event*, 2) *disconnect event*, and 3) *update event*. We assume that nodes in the network honestly synchronize the maintained information.

Connect Event. It happens at the moment when two nodes enter the transmission range of each other. When a connect event takes place, two meeting nodes honestly exchange and update their corresponding values in each field of their community table, community graph, and gateway table, according to the value of their timestamps. They also set the timestamps.

Disconnect Event. It happens at the moment when two nodes u and v go out of the transmission range of each other. Let $t^e_{u,v}$ be the end time that node u encounters node v. Let $t^b_{u,v}$ be the begin time that node u encounters node v of the ongoing encounter. Thus, the last inter-contact time between node u and v is set to $\Delta\tau_{u,v} = t^b_{u,v} - t^e_{u,v}$. The sum of inter-contact time between nodes u and v is $\tau_{u,v} = \tau^{'}_{u,v} + \Delta\tau_{u,v}$, where $\tau^{'}_{u,v}$ is the sum of inter-contact time between node u and v before last encounter. Then, the number of encounters $\sigma_{u,v}$ increases by 1.

Consequently, the average inter-contact time between two nodes u and v is expressed as follows:

$$\varpi_{u,v} = \begin{cases} +\infty, & \sigma_{u,v} = 1 \\ \frac{\tau_{u,v}}{\sigma_{u,v}-1}, & \sigma_{u,v} > 1 \end{cases} \tag{4.1}$$

The average inter-contact time between two nodes is the major parameter used by methods of community detection. Specifically, a node joins (or leaves) a community if it is (or not) qualified to be in the community. Since the community detection is out of the scope of this chapter, the reader is suggested to refer to [49] for further information.

Moreover, as in [69, 68], we assume that the nodes in the same community have the same average inter-contact time. $\varpi_{C_i} = \varpi_{u,v}$, for all nodes u and v in the same community C_i.

Update Event. The update event is periodically invoked synchronously by all nodes every γ time units. Let $C^{'}_j$ be the set of nodes which belong to a community C_j and have met node u by more than once. Then the average inter-contact time between node u and community C_j is computed as follows.

$$\varpi^u_{i,j} = \frac{\sum_{v\in C^{'}_j} \tau_{u,v}}{\sum_{v\in C^{'}_j} (\sigma_{u,v} - 1)} \tag{4.2}$$

If node u achieves an average inter-contact time with community C_j that is shorter than the one of the current gateway w (that is $\varpi^u_{i,j} \leq \varpi^w_{i,j} - \varepsilon$, where ε is a threshold), then the relevant fields are updated, i.e., the node ID of the gateway from C_i to C_j, the inter-contact time and timestamp are updated to u, $\varpi^u_{i,j}$ and the current time respectively.

4.3 Protocol Design

In this section, we first briefly describe the mechanism of CAS. We then present the design of CAS in detail. Finally, we demonstrate that our protocol is the gen-

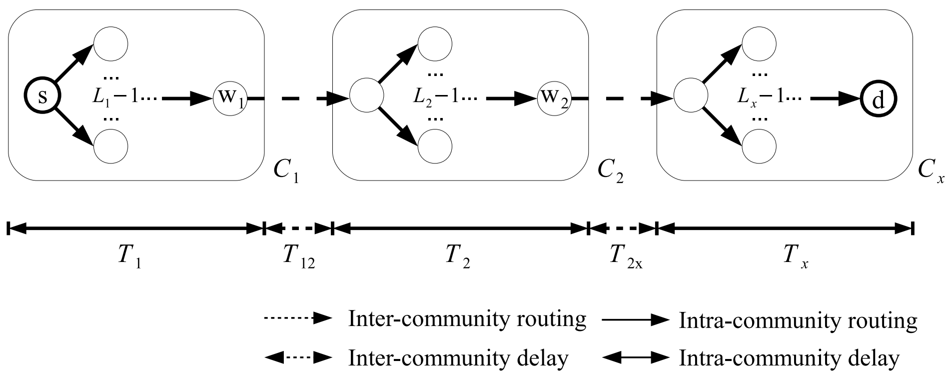


Figure 4.2: Routing Protocol Overview

eralization of several existing routing protocols.

4.3.1 Overview of CAS

In this section, we give an overview of our routing protocol. A routing example is depicted in Figure 5.1. This figure shows a number of nodes belonging to three communities C_1, C_2, and C_x. A source node s that belongs to the community C_1 wants to send a message to a node d that belongs to the community C_x.

In CAS, the routing process is composed of multiple sub-processes. In each sub-process, a message is routed to an intermediate target node, which is the gateway node bridging two communities or the destination node if the latter belongs to the same community as the source node. At the beginning of each sub-process, the shortest path from the current community to the community of the destination node is computed according to the community graph. For instance, in Figure 5.1, when the source node s generates a message, it computes the shortest path from its community C_1 to the destination node's community C_x, that is, $C_1C_2C_x$. After that, node s looks up its gateway table to find the gateway node w_1, which connects its community with the community C_2. The node w_1 is then considered as the first intermediate target node to reach. At the same time, the minimum number of message copies needed to route the message to the intermediate target node is computed, based on the message TTL and a predefined corresponding delivery ratio. These message copies are sprayed inside the current community in a binary spraying manner in which a message carrier hands over half of the copies it holds to each node it meets. When the intermediate target node encounters a node in its connecting community (e.g., C_2 in the case of w_1), it forwards the message to it. The above process is repeated until the message reaches the destination node or expires.

4.3.2 Design of CAS

As explained above, the CAS routing process is divided into multiple sub-processes. Each sub-process consists of the following two phases: 1) optimization of the number of message copies to be distributed and 2) message routing.

Optimization of the Number of Message Copies

According to the analysis of the routing performance presented in [111], if L copies of a message are distributed in the network and the TTL of the message is T, the expected delivery probability of the message can then be calculated as $p_d = 1 - e^{-\lambda LT}$, where λ is the contact rate, that is, the inverse of the average inter-contact time between nodes [12]. Hence, if the expected delivery probability p_d is assigned, the relationship between the number of message copies and the TTL can be expressed as Equation 4.3. Note that the delivery probability is the delivery ratio from the viewpoint of all generated messages.

$$L \times T = -\frac{ln(1-p_d)}{\lambda} \tag{4.3}$$

Recall that an MDTN is modeled as a directed graph of communities $G(\mathbb{M},E)$ (Section 4.2.1). Let h a path in $G(\mathbb{M},E)$. Let n be the number of communities in path h. The weight (denoted as $wgt(h)$) of path h is the sum of the weights of the edges that form the path, which is expressed in Equation 4.4. It represents the sum of the average inter-contact time between the communities in path h. Note: we assume that in the community graph that represents a MDTN every community is reachable (i.e., a non-reachable community constitutes a distinct MDTN).

$$wgt(h) = \begin{cases} \sum_{i=1}^{n-1} \varpi^{w}_{i,i+1}, & \text{if } n > 1 \\ 0, & \text{if } n = 1 \end{cases} \tag{4.4}$$

Further, let us consider two communities C_1 and C_2. Let $h(C_1,C_2)$ be any path which originates within community C_1 and ends within community C_2. In a network, it is possible that more than one such path exists. Let $H(C_1,C_2)$ be the set of all possible paths $h(C_1,C_2)$. Let $h^*(C_1,C_2)$ be the shortest path from community C_1 to community C_2.

Assume a node u carries a message denoted by $\langle S,D,I,C,L,T\rangle$. Let h^* be the shortest path from the community of node u to the community of the destination node. Let n be the number of communities in the path h^*. Let L_i denote the number of message copies distributed in a community C_i. Let w be the gateway node connecting community C_i to community C_{i+1}. Let p_d be the expected delivery probability of the message. Let λ_i be the contact rate of nodes in community C_i, which is the inverse of the average inter-contact time ϖ_{C_i} of nodes in community C_i. Then the optimization of delivery cost can be expressed as in Equation 4.5-4.7.

$$min \sum_{i=1}^{n} L_i \tag{4.5}$$

$$s.t. \sum_{i=1}^{n-1} \varpi_{i,i+1}^{w} + \sum_{i=1}^{n} \frac{-ln(1-p_d)}{\lambda_i L_i} \leq T \tag{4.6}$$

$$L_i \leq |C_i|, \text{ where } 1 \leq i \leq n \tag{4.7}$$

The purpose of the objective function Equation 4.5 is to minimize the number of message copies utilized to route a message. Equation 4.6 expresses the fact that the sum of the time spent to route the message in communities and the sum of the average inter-contact time between communities should be less than the TTL of the message. The constraint Equation 4.7 expresses the fact that the number of message copies allowed in a community should be less than the number of nodes in the community.

All L_i are positive integers and finite; thus Equation 4.5-4.7 defines a classical integer optimization problem for which a number of heuristics can be applied. However, in practice, the number of communities in path h^*, n, and the acceptable values for L_i are small, so an exhaustive search, i.e. the enumeration all the possible values for L_i, is clearly tractable. For instance, there are about 8 communities in the MIT reality mining dataset [48], and most of communities contain less than 10 nodes. The pseudo-code of the solution is illustrated in Figure 4.3.

Without loss of generality, let's consider that the algorithm is invoked by a node u. Node u has a message m. The shortest path from the community of node u to the community of the destination node of message m is h^*. The algorithm initially gets the community (denoted by C^*) next to the community of nodes u on the path h^* (Line 1); and sets the number of the nodes in node u' community as the default value for L_u, which indicates the number of message copies of message m for node u (Line 2). The algorithm then enumerates all the possible values for the number of copies in each community on the path h^* (Lines 19 to 31). More specifically, the algorithm uses the elements in a vector π to express the number of copies of message m allocated for each community on the path h^*. The algorithm initially sets all the elements in π to be 1. It then starts with the last element in π (Line 19), and selects all possible values for the last element (Lines 21 to 23). The algorithm then uses the values of the elements in π to verify whether (1) the allocated number of message copies can satisfy Equation 4.6; (2) the sum of these values is smaller than the previous minimal sum, which is initially set as the number of nodes on the path h^* (Line 15). If so, the algorithm updates the minimal sum (Line 16) and L_u (Line 17). The algorithm then resets the value of the last element to be 1 (Line 25), and goes to the element before the last element (Line 26). The above process is repeated until all possible values are enumerated. Finally, the algorithm returns C^* and L_u.

There are two heuristic methods to reduce the computation complexity of algorithm OptimalCost: (1) using a smaller sample to test the values for $L_2, \ldots, L_n$. (2) terminate the computation of the algorithm when the first combination of values in π is found.

Message Routing

In CAS, we distinguish the routing within a community from the routing among communities. In the former case, the allocated message copies are sprayed inside the current community in a binary manner, in order to minimize the time to spray message copies. Specifically, a node hands over half of the number of a message copies to an encountered node, which is in the same community and does not have the message. When the node has just one copy of the message, the message can only be forwarded to the intermediate target node. As we stated in Section 4.2, the meeting frequency of nodes in the same community is much higher than that of nodes in different communities. Thus, messages can be delivered to the intermediate target node with a higher probability.

As for the routing among communities, a gateway node forwards a message to an encountered node if it belongs to the next community along the shortest path from the gateway node's community to the community of the destination node. After it gets the message, the receiving node re-computes the shortest path from its community to the community of the destination node, and sets the intermediate target node as the gateway node bridging the next community, or the destination node if it is in the same community as the destination node. After that, it calls the Algorithm *OptimizeCost* to compute the needed number of message copies utilized to route the message to the intermediate target node. The complete mechanism of the routing protocol in CAS is summarized in Figure 4.4.

4.3.3 CAS Generalizes Classes of Routing Protocols

Another relevant point about CAS is that it represents the generalization of many routing protocols, including Direct [115], Spray-and-Wait [111], Epidemic [117], and Clustering [25]. Depending on the number of communities in the network and the number of message copies, CAS can dynamically transform to these routing protocols. Let V be the set of all nodes in a network. Let $\mathbb{M}$ be the set of all

Algorithm: OptimizeCost

Participants: Node u.

Input: (1) D_m, the destination node of message m. (2) T, the TTL of message m. (3) p_d, a predefined delivery probability of message m. (4) h^*, the shortest path from node u's community to node D_m's community.

Output: (1) C_u^*, the community next to node u's community on the path h^*. (2) L_u, the number of copies of message m carried by node u.

Setup: (1) n, the number of communities in path h^*. (2) ϖ_{sum}, the weight of the shortest path h^*. (3) L_{min}, the minimum number of message copies utilized to route the message. It is initially set to the sum of the number of nodes in each communities in the path h^*. (4), π, $1 \times n$ vector, whose elements are 1 (5) λ_i, the contact rate of nodes in community C_i, where $1 \leq i \leq n$.

$C_u^* \leftarrow h^*[2]$ {The index of the community C_u^* is 2.}
$L_u \leftarrow |h^*[1]|$
$j \leftarrow n$ {The index for the vector π.}
while $j \geq 1$ **do**
  $T_{sum} \leftarrow 0$
  $L_{sum} \leftarrow 0$
  {Compute the needed number of message copies for a combination of values in π.}
  for $i \leftarrow 1$ **to** n **do**
    if $\pi[i] \leq |h^*[i]|$ **then**
      $L_i' \leftarrow \pi[i]$
    else
      $L_i' \leftarrow |h^*[i]|$
    end if
    $T_{sum} \leftarrow T_{sum} + \frac{-ln(1-p_d)}{\lambda_i L_i'}$
    $L_{sum} \leftarrow L_{sum} + L_i'$
  end for
  {Find the minimal number of message copies.}
  if $T_{sum} + \varpi_{sum} \leq T$ **and** $L_{sum} < L_{min}$ **then**
    $L_{min} \leftarrow L_{sum}$
    $L_u \leftarrow \pi[1]$
  end if
  {List all possible combination of values in π}
  $j \leftarrow n$
  while $j \geq 1$ **do**
    if $\pi[j] < |h^*[j]|$ **then**
      $\pi[j] \leftarrow \pi[j] + 1$
      break
    else
      $\pi[j] \leftarrow 1$
      $j \leftarrow j - 1$
    end if
  end while
end while
return $\langle C_u^*, L_u \rangle$

Figure 4.3: Algorithm:OptimizeCost

Protocol: RouteMessage

Participants: Two encountering nodes u and v.

Input: (1) m, a message carried by node u. (2) D_m, the destination node of message m. (3) I_m, the intermediate target node of message m. (4) L_u, the number of copies of message m for node u. (5) L_v. (6) T, the TTL of message m. (7) p_d, a predefined message delivery probability. (8) C_u^*, the community next to node u's community on the shortest path from node u's community to node D_m's community. (9) C_v^*.

Output: Message m is forwarded to node v, if one of the following conditions is met: (1) node v is the destination; (2) node v is the intermediate target node; (3) nodes u and v are in the same community, and node u has more than one copy of message m; (4) node u is the gateway node bridging community C_u^*, and node v is in community C_u^*, and node u has not yet forwarded message m to another node member of the community C_u^*. Furthermore, the number of copies of message m for node v, L_v, is computed and the intermediate node on the message route is updated.

Events and Associated Actions:

node u **initiates the protocol**

if $C^u = C^v$ **then**
    if $v = I_m$ **then**
        node u forwards message m to node v.
    else if $L_u > 1$ **then**
        node u forwards half number of copies of message m to node v, and keeps the remaining copies.
    end if
else
    if $v = D_m$ **then**
        node u forwards message m to node v
    else if u is the gateway node connecting C_u^* **and** $C_u^* = C^v$ **and** node u has not yet forwarded message m to another node in the community C_u^* **then**
        node u forwards message m to node v
    end if
end if

upon node v **receives a message** m **from node** u

if $C^u \neq C^v$ **and** $v \neq D_m$ **then**
    $\langle C_v^*, L_v \rangle \leftarrow OptimizeCost(D_m, T, p_d)$
    if $C^{D_m} = C^v$ **then**
        $I_m \leftarrow D_m$
    else
        node v looks up its gateway table to find the gateway node w to community C_v^*
        $I_m \leftarrow w$
    end if
end if

Figure 4.4: Protocol:RouteMessage

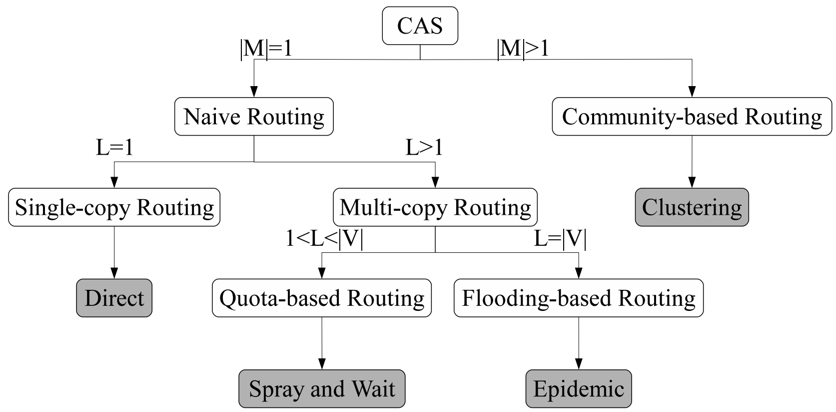


Figure 4.5: The transformation from CAS to several kinds of routing protocols, where L is the number of allocated copies for a message, V is the set of all nodes in a network, M is the set of all communities in a network.

communities in the network. Let L be the number of allocated copies for a message. The transformation from CAS to these routing protocols is then illustrated in Figure 4.5.

- Direct ($L = 1$, $|\mathbb{M}| = 1$): The source node of a message can only forward the message to the destination node. Direct is a well known single-copy routing protocol.

- Spray-and-Wait ($1 < L < |V|$, $|\mathbb{M}| = 1$): Each message is associated with some forwarding tokens, which indicates message copies. Each message has two phases: the spray phase and the wait phase. If there is more than one forwarding token left, the message is in the spray phase. During the spray phase, the forwarding tokens of a message can be sprayed to an encountered node without the message. If there is only one forwarding token left, it is in the wait phase. During the wait phase, a message can only be forwarded to the designation node. Spray-and-Wait is a representative routing protocol of quota-based routing.

- Epidemic ($L = |V|$, $|\mathbb{M}| = 1$): Each node stores its messages in its buffer. When two nodes meet, each of them complements the missing messages according to the messages in the other's buffer. It is therefore flooding-based in nature.

- Clustering ($|\mathbb{M}| > 1$): The encounter information of nodes is synchronized and processed to identify the community structure and the gateway nodes. A gateway node connecting its community to another community is the node that has the highest probability of meeting any node in the latter community. In this protocol, Direct routing is employed to route a message to a gateway node or the destination node if the message is in the destination node's community. Clustering routing is a community-based routing.

It is worth pointing out that Epidemic and Direct achieve the upper and lower bounds of routing performance in terms of delivery ratio and delivery cost [114, 115] respectively. CAS realizes some kind of trade-off by minimizing the delivery cost while maintaining a predefined delivery ratio.

4.4 Analysis

In this section, we first develop an analytical model for the cover time of the community graph. We then evaluate the accuracy of our analytical model by comparing the theoretical results with the simulation results by using a standard simulator.

4.4.1 Analytical Model

Since our protocol depends on the local community graph to allocate message copies, it is necessary to estimate the cover time of the community graph when the maintained information is changed at a node.

Cover Time of Maintained Information

In the following sections, we theoretically analyze the cover time of maintained information from the following two aspects: 1) intra-community message exchange and 2) inter-community message exchange.

A. Intra-community message exchange

When the role of a node changes within the network – for instance, when a node becomes the new gateway node – the information about this change is propagated to all the relevant nodes of the community. This process applies to all the information that has to be maintained to grant the correct operation of the network. The cover time of the maintained information inside a community is defined as the difference from the time of the change at a node to the time when all the other nodes in the community are aware of the change.

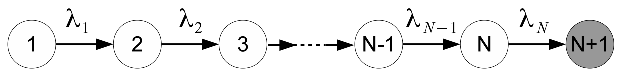


Figure 4.6: The continuous time Markov chain model for modeling the flooding process in a community. States (1) to (N) are N transition states and state $(N+1)$ is the absorbing state.

Let us consider a community C, which has $N+1$ nodes. According to the update mechanism, the modified information (see the case of a message hereafter) is disseminated inside the community through flooding. Let $n(t)$ represent the number of nodes with a given message at time t. We model the flooding process by a one-dimensional continuous time Markov chain (with state $n(t)$). States and transitions of this chain are illustrated in Figure 4.6. We can observe that this Markov chain starts with state (1) – when a message is generated by a node – and has N transient states and an absorbing state. When all nodes in the community C receive the message, the system enters the absorbing state (the state $(N+1)$). The corresponding infinitesimal generator matrix $\mathbf{Q}$, with dimension of $N+1$, is shown in Equation 4.8.

$$\mathbf{Q} = \begin{pmatrix} \mathbf{D} & \mathbf{R} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}. \tag{4.8}$$

where the sub-matrix $\mathbf{D}$ is a $N \times N$ matrix with element $D_{i,j}(1 \leq i, j \leq N)$ expresses the transition rate from transient state (i) to transient state (j). $\mathbf{R}$ is a $N \times 1$ matrix with element $R_{i,N+1}$ representing the transition rate from transient state (i) to the absorbing state $(N+1)$. The left $\mathbf{0}$ matrix is a $1 \times N$ vector with all elements 0 representing the transition rates from the absorbing state to transient states. The right $\mathbf{0}$ matrix includes only a single element, 0, representing the negative sum of the left $\mathbf{0}$ vector. Based on the above choice of the message dissemination process, we obtain the transition rate $q_{i,j}$ from state (i) to state (j) as follows.

In the flooding process, each of the nodes with the message replicates the message to an encountered node without the message. Assume that the system is currently in state $(n)_{n \leq N}$, that is, there are n nodes with the message and $N+1-n$ nodes without the message. When one of the nodes without the message encounters a node with the message, the system state turns to state $(n+1)$. The transition rate from state (n) to state $(n+1)$ is $\lambda_n = (N+1-n)n\lambda$, since there are $(N+1-n)$ nodes that can receive the message from n nodes and meetings take place at rate λ. When the last node receives the message, the system turns to the absorbing state (i.e., state $(N+1)$). Let $D\{(j)|(i)\}$ be the transition rate from state (i) to state (j) and let $R\{(N+1)|(i)\}$ be the transition rate from the state (i) to the absorbing state. The non-zero transition rates in the Markov chain can be expressed as in Equation 4.9.

$$\begin{cases} D\{(n+1)|(n)\} = (N+1-n)n\lambda, n \in [1, N-1] \\ R\{(N+1)|(n)\} = N\lambda, \text{ if } n = N \\ D\{(n)|(n)\} = -D\{(n+1)|(n)\} - R\{(N+1)|(n)\}, n \in [1, N] \end{cases} \tag{4.9}$$

B. Inter-community message exchange

The cover time in the inter-community message exchange is defined as the difference between the time when the information to be maintained is changed at a node in a community C_i and the time when all nodes in another given community (e.g., C_j) are aware of the change.

Let us consider two communities C_i and C_j, which have $M+1$ and $N+1$ nodes respectively. Unlike in the single community case, when flooding a message from one community to another one we need to record the number of nodes with the message in each of the two communities. Consequently, as in [67], we use a two-dimensional continuous time Markov chain to model the flooding process in two communities.

Moreover, as pointed in [46], the distributions of inter-contact time between nodes belonging to the same community and nodes belonging to different communities are different. Thus, we set λ^{intra} and λ^{inter} as the contact rate of the nodes belonging to the same community and different communities respectively.

The Markov chain consists in states which can be indicated by $(m(t), n(t))$, where $m(t)$ (or $n(t)$) represents the number of nodes with the message in community C_i (or C_j) by time t. The state transition is illustrated in Figure 4.7. We can observe that this Markov chain starts with state $(1, 0)$ when a message is generated by a node in community C_i, and has $F = (M+1)(N+1)$ transient states. When all the nodes in community C_j have received the message, the system enters into the absorbing state, denoted by $(x, (N+1))$, where $1 \leq x \leq M+1$ (in this case, we are considering as final state of the propagation to the second community, the state where all the nodes of the second community have received the message, irrespectively on the number of nodes of the first community who have been informed). Similar to what we did for the model of the flooding of a message inside a single community, we can obtain a generator matrix $\mathbf{Q}$ expressed by Equation 4.8.

Here, the sub-matrix $\mathbf{D}$ is a $F \times F$ matrix, $\mathbf{R}$ is a $F \times 1$ matrix, the left $\mathbf{0}$ matrix is a $1 \times F$ vector, and the right $\mathbf{0}$ matrix includes a single element 0, which have the same meanings as in the case of the flooding of a message inside a single community. Assume the current state is state $(m, n)_{m \leq M, n \leq N}$ (that is, the number of nodes

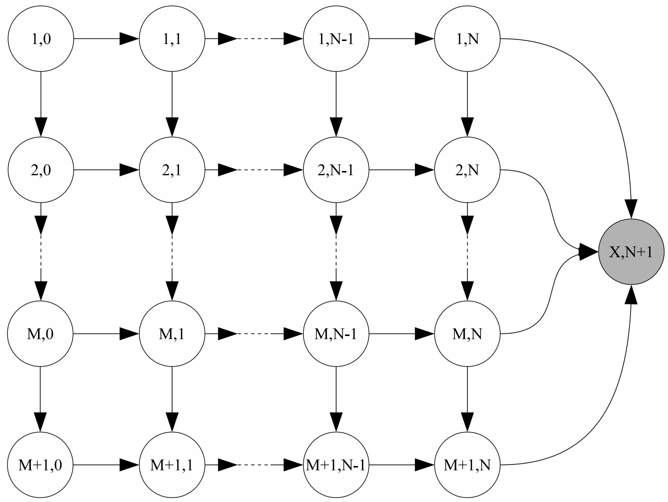


Figure 4.7: The two-dimensional continuous time Markov chain model for modeling the message dissemination within two communities. States $(1,0)$ to $(M+1,N)$ are $(M+1)(N+1)$ transition states and state $(x,N+1)$ is the absorbing state.

in community C_i and C_j with the message is m and n respectively). The transmission rate from state (m,n) to state $(m+1,n)$ is $(M+1-m)(m\lambda^{intra}+n\lambda^{inter})$, since $M+1-m$ nodes without the message can receive the message from m nodes inside community C_i with rate of λ^{intra} or from n nodes inside community C_j with rate of λ^{inter}. Similarly, the transmission rate from state (m,n) to state $(m,n+1)$ is $(N+1-n)(m\lambda^{inter}+n\lambda^{intra})$, since $N+1-n$ nodes without the message can receive the message from m nodes inside community C_i with rate of λ^{inter} or from n nodes inside community C_j with rate of λ^{intra}. When the last node in community C_j receives the message, the system turns to the absorbing state (i.e., state $(x,N+1)$), and the transition rate is $m\lambda^{inter}+N\lambda^{intra}$. Thus, the non-zero transition rates in the Markov chain can be expressed as in Equation 4.10.

$$
\begin{cases}
D\{(m+1,n)|(m,n)\} = (M+1-m)(m\lambda^{intra}+n\lambda^{inter}), m\in[1,M], n\in[0,N] \\
D\{(m,n+1)|(m,n)\} = (N+1-n)(m\lambda^{inter}+n\lambda^{intra}), m\in[1,M+1], n\in[0,N-1] \\
R\{(x,N+1)|(m,n)\} = (m\lambda^{inter}+N\lambda^{intra}),\ \text{if } m\in[1,M+1], n=N \\
D\{(m,n)|(m,n)\} = -D\{(m+1,n)|(m,n)\}-D\{(m,n+1)|(m,n)\} \\
\qquad\qquad -R\{(x,N+1)|(m,n)\}, m\in[1,M+1], n\in[0,N]
\end{cases}
\tag{4.10}
$$

Computation of the Cover Time

Let F be the dimension of the transition matrix $\mathbf{D}$. Based on the transition matrix $\mathbf{D}$, we can derive the cover time of the maintained information in the intra- and in the inter-community case, denoted by $D^d_{(intra)}$ and $D^d_{(inter)}$ respectively, as the following expression [54, 67]:

$$D^d_{(\cdot)} = \mathbf{e}\left(-\mathbf{D}^{-1}_{(\cdot)}\right)\mathbf{1} \tag{4.11}$$

where $\mathbf{e}$ is a $1 \times F$ vector denoting the initial state probability vector $\mathbf{e} = [1, 0, \ldots, 0]$, whereas $\mathbf{1}$ is a $F \times 1$ all-one vector, that is, $\mathbf{1} = [1, 1, \ldots, 1]^T$.

4.4.2 Model Validation

In this section, we evaluate the accuracy of our analytical model by comparing its predictions to the simulation results, which were obtained by simulating message dissemination through a standard simulator.

Evaluation Settings

The simulation scenario considers a rectangular area of 2000 m $\times$ 1000 m. This area is equally partitioned into 2 regions, that is, each region is an area of 1000 m $\times$ 1000 m. Initially, a given number of nodes are deployed in each region. Each node considers the region in which it has been deployed as its *local region*. According to the mobility model, which will be further described below, a node is more likely to visit its local region than other regions. This leads to the encounter frequency of nodes deployed in the same region being much higher than the encounter frequency of nodes deployed in different regions. Consequently, nodes associated to one region constitute a community.

Let r be the transmission range. Let v be the speed of the nodes. Following [68], the parameters are set as follows: $r = 20$ m, $v = 10$ m/s. The number of nodes is set as a variable parameter for the investigation. Let λ^{intra} be the contact rate of nodes in the same community. Let λ^{inter} be the contact rate of nodes belonging to different communities. According to the study in [112], the contact rates λ^{intra} and λ^{inter} are set to 1.703 h^{-1} and 0.672 h^{-1} respectively in the simulation model.

The simulations, in the simulator ONE (Opportunistic Network Environment simulator) [57], also include the settings of the message generation parameters. Every minute, a message is generated with a random node as the source and all other nodes in the community as the destinations. Therefore, there are 360 messages generated in each simulation. Moreover, in order to ensure that each message can eventually reach all destination nodes, the TTL is set to “never expire”.

Mobility Model

We adopted the community-based mobility model proposed in [112], which has been widely used for the evaluation of community-based routing protocols [42, 25]. In this mobility model, each community is associated with a geographical area. The movement of a node, which belongs to a community, consists of a sequence of *local* and *roaming* epochs. A local (or roaming) epoch of a node is a movement with a given speed towards a random destination inside the area associated with the community (or the entire network). When the node reaches the destination, it stays there for a certain period of time (i.e, a pause time). When the pause time expires, it starts a new epoch. If the node's previous epoch was a local one, the next epoch is a local one with probability p_l, or a roaming epoch with probability $1-p_l$. Similarly, if the node's previous epoch was a roaming one, the next epoch is a roaming one with probability p_r, or a local one with probability $1-p_r$.

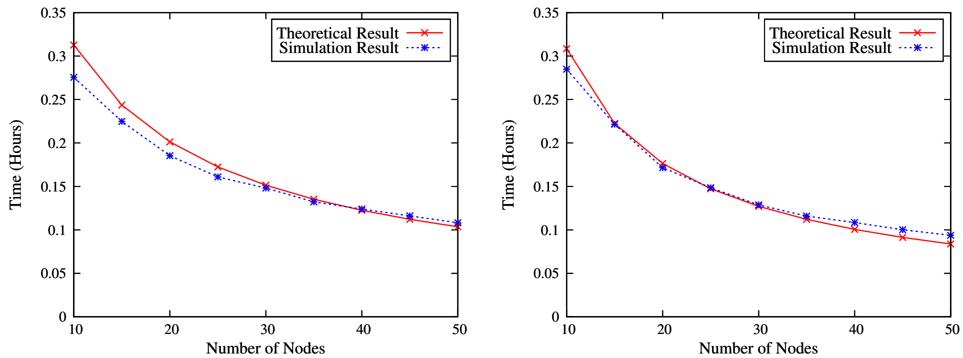
In our case, in order to avoid biasing the cover time by the pause time, we set the pause time in the simulations to zero [69, 81]. Moreover, we assign the same mobility characteristics to all the nodes in the network, since these characteristics do not influence the accuracy of the model. Specifically, the values of p_l and p_r are set to 0.8 and 0.2 respectively.

Evaluation Results

Figure 4.8a and Figure 4.8b show the results obtained for the cover time for the case of intra-community and inter-community. It can be seen that the average deviation of the theoretical results from the simulated results is small. Specifically, in the case of intra-community, the minimum, maximum, and average deviations are 1.13%, 13.50%, and 5.64% respectively; while in the case of inter-community, the minimum, maximum, and average deviations are 0.45%, 10.57%, and 4.83% respectively. This demonstrates the accuracy of our analytical model in the evaluation of the cover time.

4.5 Performance Evaluation

In this section, we present the performance evaluation of CAS. We first introduce our simulation settings and the mobility model in Sections 4.5.1 and 4.4.2 respectively. We then present the routing protocols against which we compare the performance of CAS in Section 4.5.2 followed by the description of the performance evaluation metrics in Section 5.5.4. Finally, we present the simulation results in Section 5.5.5.



(a) Intra-community (b) Inter-community

Figure 4.8: Theoretical and simulation result comparison in the case of: (a) intra-community and (b) inter-community message exchange. The contact rates between nodes in the same community and different communities are $\lambda^{intra} = 1.703\ \mathrm{h}^{-1}$ and $\lambda^{inter} = 0.672\ \mathrm{h}^{-1}$ respectively.

4.5.1 Simulation Settings

The simulations have been conducted by the ONE simulator [57]. The simulation scenario considers a rectangular area of 2000 m $\times$ 1500 m. This area is equally divided into 12 regions each measuring 500 m $\times$ 500 m. Initially, twenty nodes are deployed in each region. Each node considers the region in which it has been deployed as its *local region*. According to the mobility model (see Section 4.4.2), a node is more likely to visit its local region than other regions. This leads to the encounter frequency of nodes deployed in the same region being much higher than that of nodes deployed in different regions. Consequently, nodes associated to one region constitute a community.

In addition, we assign different mobility characteristics to the nodes in a region in order to produce a scenario closer to the reality, where there is heterogeneity in mobility among nodes. To achieve this, the nodes in a region are divided into two kinds: *low mobility* nodes and *high mobility* nodes. The low mobility nodes can only roam inside their local regions, while the high mobility nodes may roam among their local region as well as the entire space. Specifically, the values of p_l and p_r for the low mobility nodes are set to 1 and 0 respectively. Whereas, the values of p_l and p_r for the high mobility nodes are set to 0.5.

Moreover, each node is equipped with a radio interface for short-range communication. The transmission range and bandwidth are set as 10 m and 2 Mb/s respectively. This is consistent with contemporary protocols, such as Bluetooth [57].

Table 4.1: Parameter settings

Parameter Name	Value
Simulation area	2000 m $\times$ 1500 m
Transmission range	10 m
Simulation duration	13 hours + TTL
Warm-up period	1 hour
Message generation rate	1 message per minute
Number of communities	12
Number of nodes in a community	20
Node speed	1.34 m/s
p_l (low mobility nodes)	1
p_r (low mobility nodes)	0
p_l (high mobility nodes)	0.5
p_r (high mobility nodes)	0.5
γ	1 hour
p_d	0.8

Additionally, the speed of nodes is set to 1.34 m/s, which is an average human walking speed [57]. Since we find that the routing performance of CAS and the chosen routing protocols converges after about six hours, we set thirteen hours for each simulation in order to achieve statistical confidence. Since CAS depends on the knowledge of the network topology, we specify the first hours as a warm-up period to allow the nodes to setup the community graph. During the warm up period, no message is generated. After this period, every minute, a random node sends a message to a random destination node. The detailed settings are listed in Table 5.1.

4.5.2 Routing Protocols

Based on the above settings, we have compared the performance of CAS against the following protocols:

Epidemic: Each node forwards a copy of each unexpired message in its buffer to any encountered node that does not have a copy of the message.

Direct: The source node of a message can only forward the message to the destination node.

Binary Spray-and-Wait (BSW): Each message has L copies. A message carrier forwards half of the message copies to an encountered node, if its $L > 1$ and the latter does not have the message. The message, which has only one message copy left, can only be forwarded to the destination.

Bubble: It utilizes social information about nodes, such as their centrality and the community to which they belong. In this protocol, a message is forwarded based on the global rankings of two encountering nodes, until it reaches a node in the community of the destination node. After that, the message is forwarded based on the local rankings of two encountering nodes, until it either reaches the destination node or expires. In the experiment, the length of time window is set to 1 hour.

4.5.3 Performance Metrics

The goal of CAS is to minimize the number of message copies and while maintaining a predefined delivery ratio. Therefore, we have measured the following metrics for the simulations that we have conducted in this work:

Delivery ratio: The proportion of messages that have been delivered out of the total unique messages created.

Delivery cost: The total number of message transmissions in the simulation. To normalize this, we divide it by the total number of unique messages created.

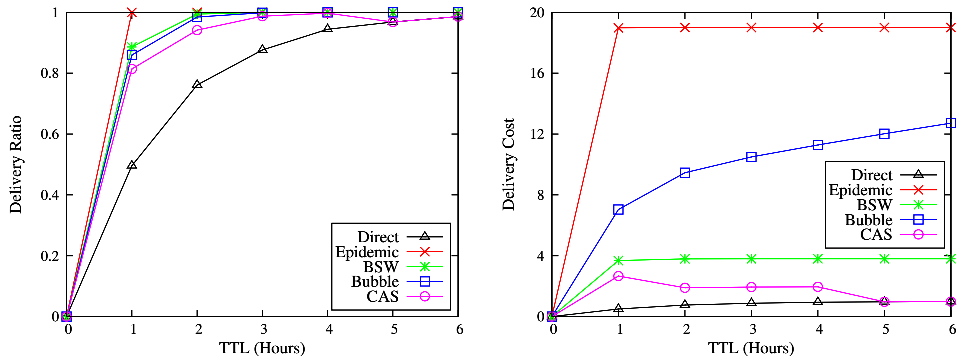
4.5.4 Simulation Results

As mentioned in Section 4.3.3, CAS includes some basic routing protocols as special cases according to the number of communities in the network and the allocated number of message copies. Therefore, in this section, we compare the routing performance of CAS with the chosen routing protocols in the following two cases: single-community case and multiple-community case.

Single-community Case

We arbitrarily select one of the twelve regions for this study. The nodes deployed in this region constitute a community. For the Binary Spray-and-Wait protocol, each message is initially associated with 4 copies. Thus, at most 20% nodes can participate in message routing in Binary Spray-and-Wait.

Figure 4.9(a) shows the delivery ratio of the compared routing protocols. We can observe that, as expected, Epidemic and Direct always achieve the best and worst delivery ratio respectively, for all values of TTL. We also observe that the delivery ratio of Bubble is very close to that of Binary Spray-and-Wait. Binary Spray-and-Wait achieves a better delivery ratio than CAS. The maximum and minimum difference between CAS and Binary Spray-and-Wait is 7.22% and 0.28%. The delivery ratio of CAS is always higher than the predefined delivery ratio (i.e., 0.8). It is worth noting that CAS achieves the same delivery ratio as Direct when



(a) Delivery Ratio

(b) Delivery Cost

Figure 4.9: Comparison of the routing performance of several algorithms in the single-community case.

the TTL is greater or equal to 5 hours. This is because CAS allocates the same number of copies for each message as Direct. This indicates that CAS can dynamically transform to Direct.

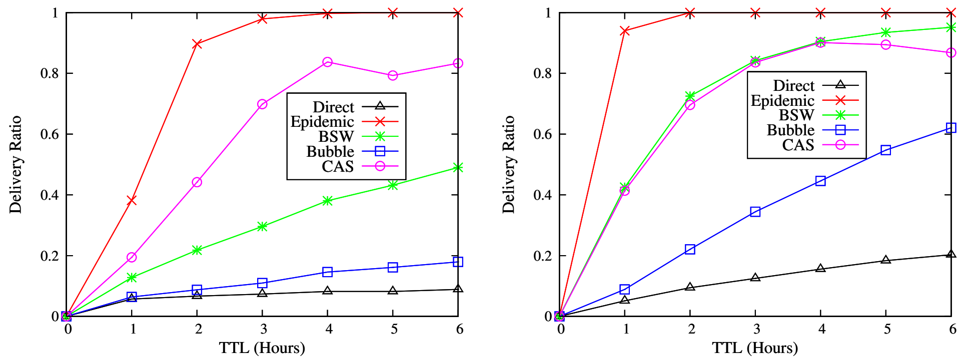
Figure 4.9(b) shows the delivery cost of the compared routing protocols as a function of the TTL of the generated messages. We observe that Epidemic and Direct have the highest and lowest delivery cost respectively, whatever the value of TTL. The delivery cost of Bubble keeps increasing as the TTL increases. The delivery cost of Binary Spray-and-Wait is lower than that of Bubble and remains stable as the TTL increases. However, it is very close to the allocated number (i.e., 4) of copies for each message in Binary Spray-and-Wait. It is worth noting that unlike other routing protocols, the delivery cost of CAS decreases as the TTL increases. In particular, CAS achieves the same delivery cost as Direct when the TTL is greater or equal to 5 hours. This is because CAS can dynamically allocate copies for a message according to its TTL.

Multiple-community Case

Concerning the multiple-community case, we conducted two experiments to investigate the impact of the density of high mobility nodes and of the mobility model settings on the routing performance of our protocol.

A. Impact of the Density of High Mobility Nodes

In this section, we investigate the impact of the density of high mobility nodes



(a) (b)

Figure 4.10: Comparison of delivery ratio of several algorithms with different densities of high mobility nodes. 5% and 50% nodes are chosen as the high mobility nodes in (a) and (b) respectively.

on the routing performance of CAS and the chosen routing protocols. The investigation is conducted by two experiments, in which the densities of high mobility nodes are set as 5% and 50% respectively. In order to avoid the impact of the settings for roaming nodes on the routing performance, the values of p_l and p_r for high mobility nodes are set to 0.5. Moreover, for Binary Spray-and-Wait, each message is initially associated with 24 copies. Since there are twenty nodes in each of the twelve regions, at most 10% nodes can participate in message routing in Binary Spray-and-Wait.

Figs. 4.10(a) and (b) show the delivery ratio of the compared routing protocols with different densities of high mobility nodes. As expected, we can observe that Epidemic and Direct always achieve the best and worst delivery ratio respectively. We also observe that CAS always achieves a much better delivery ratio than Bubble for all values of TTL. Indeed, the maximum difference between the delivery ratio of CAS and Bubble is about 69.17% and 49.17% in Figs. 4.10(a) and (b) respectively. This is because CAS takes advantage of gateway nodes to steer the forwarding of the messages whose source and destination nodes belong to different communities in the right direction. Figure 4.10(a) shows that CAS achieves a much better delivery ratio than Binary Spray-and-Wait; while Figure 4.10(b) shows that Binary Spray-and-Wait even achieves a little better delivery ratio than CAS. This results from that the increasing number of roaming nodes raises the opportunity of message exchange between different communities. This also reflects the shortcoming of Binary Spray-and-Wait, that is, assuming that nodes have the same capability of visiting the entire network. Moreover, It is worth noting that the de-

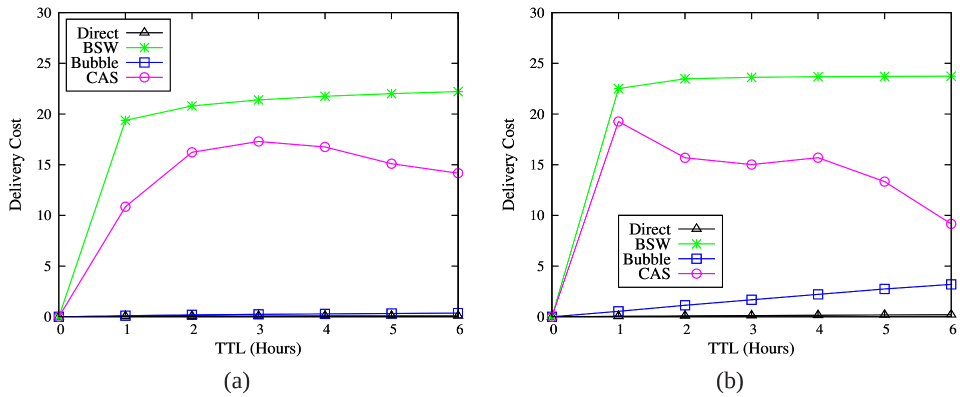


Figure 4.11: Comparison of delivery cost of several algorithms with different densities of roaming nodes. 5% and 50% nodes are chosen as the roaming nodes in (a) and (b), respectively.

livery ratio achieved by CAS is a little lower when TTL is 5 hours than that when TTL is 4 hours. This is because CAS allocates less copies for messages with longer TTLs.

The performance of the delivery cost of the compared routing protocols is illustrated in Figs. 4.11(a) and (b). Since the delivery cost of Epidemic increases much quickly than that of other protocols, we omit the delivery cost of Epidemic to illustrate the delivery cost of other protocols in detail. We can observe that Direct always has the lowest delivery cost. Bubble achieves a lower delivery cost than CAS in Figure 4.11(a) and (b). However, as shown in Figure 4.10(a), the delivery ratio of Bubble is unacceptable, while, in Figure 4.10(b), the delivery ratio of Bubble is much lower than CAS. If the predefined delivery ratio of CAS is set to the deliver ratio achieved by Bubble, the delivery cost of CAS is also low. Additionally, Binary Spray-and-Wait always consumes the assigned number of message copies. Figure 4.11(a) shows that the delivery cost of CAS raises as the TTL increases, when the TTL is less than 3 hours. Since the TTL increase, the gateway nodes can reach more communities, which invokes the allocation of message copies. When the TTL is greater than 3 hours, the delivery cost of CAS decreases as the TTL increases. This is because CAS allocates less copies for the messages with longer TTLs. Figure 4.11(b) shows that the delivery cost of CAS decreases as the TTL increases. The increasing number of high mobility nodes can enhance the successful delivery probability for the messages whose source and destination nodes belong to different communities, which can terminate the allocation of message.

B. Impact of the Settings of the Mobility Model

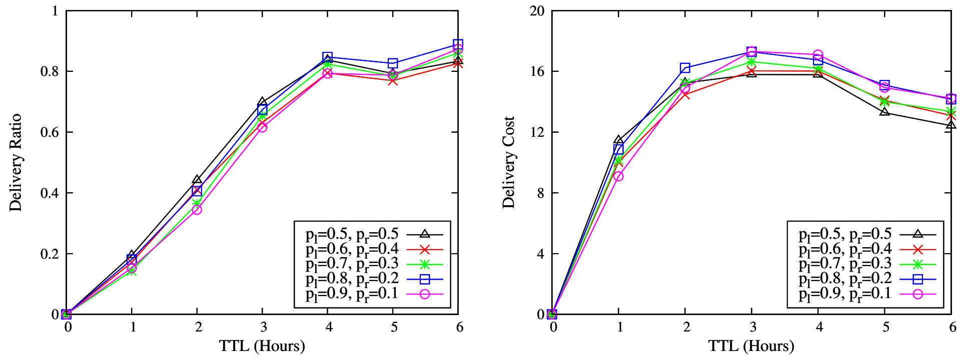
In this section, we investigate the impact of the settings of the chosen mobility model on the routing performance of CAS. As we stated in Section 4.5.1, a high mobility node is more likely to visit its local region than other regions. Hence, for high mobility nodes, we vary the value of p_l from 0.5 to 0.9 with step by 0.1 and set the value of p_r as $1-p_l$ in the simulations. Moreover, 5% nodes are chosen as high mobility nodes in order to avoid the impact of the density of high mobility nodes on the routing performance of CAS.

First we look at the delivery ratio. From the results illustrated in Figure 4.12(a), we can observe that CAS achieves similar results under different settings with respect to p_l and p_r. The settings with $p_l = 0.5, p_r = 0.5$ and $p_l = 0.9, p_r = 0.1$ always achieve the best and worst delivery ratio respectively, when the TTL is less than 4 hours. This is because a higher probability p_r can provide more opportunities to route messages whose source and destination nodes are in different communities. When the TTL is greater than 4 hours, the setting with $p_l = 0.8, p_r = 0.2$ achieves the best delivery ratio while the setting with $p_l = 0.6, p_r = 0.4$ achieves the worst delivery ratio. As the TTL increases, a higher probability p_l can indeed make the gateway nodes collect more messages whose source and destination nodes are in different communities.

Next we compare the delivery cost of CAS with different settings. As shown in Figure 4.12(b), we can observe that CAS achieves similar results under different settings with respect to p_l and p_r. When the TTL is greater than 3 hours, the setting with $p_l = 0.5, p_r = 0.5$ has the lowest delivery cost. Indeed, with a higher probability p_r, the gateway nodes collect fewer messages from their community. Additionally, the gateway nodes can forward the messages whose source and destination nodes belong to different communities earlier to the next communities, which in turn results in fewer message copies are allocated in the next communities.

4.6 Chapter Review

In this chapter, we presented CAS, a self-regulating protocol for efficient routing in mobile delay tolerant networks. CAS can dynamically control the message replication based on the urgency of messages. We demonstrated that CAS includes some basic routing protocols as special cases. An analytical model that estimates the cover time of the local knowledge maintained in CAS was developed and validated by simulations. Our simulations on a widely used community-based mobility model, demonstrate that CAS can improve the routing performance compared to



(a) Delivery Ratio (b) Delivery Cost

Figure 4.12: The impact of the settings of the mobility model on the routing performance of CAS

the quota-based Binary Spray-and-Wait protocol and the community-based Bubble protocol.

Chapter 5

An Efficient Privacy Preserving Prediction-based Routing Protocol for Mobile Delay Tolerant Networks

5.1 Introduction

As discussed in chapter 1.3.2, a number of routing protocols in MDTNs belong to the category of prediction-based routing. These algorithms compute the encounter probability of nodes (i.e., the probability that a given node encounters another node in the network) by analyzing their logs of encounters in the network. Nodes then take routing decisions building on the computed encounter probabilities. As previously discussed, these protocols perform better than other protocols when nodes exhibit well-known mobility patterns. However, prediction-based routing protocols implicitly assume that nodes accept to reveal their mobility patterns to other nodes. Unfortunately, such an assumption is not realistic, since this information can be used to infer private information about them, as demonstrated by Gambs et al. [33].

To the best of our knowledge, only we in [42] have addressed the privacy issue of prediction-based routing protocols. In [42], message routing is guided by the maximum probability that nodes in a community will encounter a destination node. In order to compute the value of such maximum probability, the protocol in [42] needs to run $2+\beta$ (where $\beta \geq 7$) times of another protocol (named private_sum), which computes the sum of the probability that nodes in a community will encounter a destination node. In each run of the private_sum protocol, kN messages are exchanged among the nodes in a community, where N is the number of nodes in the community, k is a constant and $2 \leq k < N$. We present in this chapter an Efficient Privacy Preserving Prediction-based Routing protocol in DTNs, named E3PR, which preserves the privacy of the node mobility patterns and is computationally efficient. Indeed, E3PR has a running time of $1/(\beta+2)$ of that of the protocol presented in [42], where $\beta \geq 7$. Similarly to the protocol presented in [42], E3PR is intended for environments in which nodes belong to communities. Recent studies of real mobility traces have shown that this is the case for most nodes in real settings [48, 10, 25, 9].

For routing a message, E3PR distinguishes the routing inside a community from the routing between communities. For disseminating a message inside a community, E3PR relies on the epidemic protocol [117], which by construction preserves the privacy of nodes and is efficient as communities are small. The main challenge addressed by E3PR is thus the routing of a message between communities in a privacy preserving manner. To do so, each node in the network calculates the probability that at least one of the nodes in its community will encounter the destination. When two nodes from different communities encounter, instead of comparing their respective probabilities to encounter the destination node, they compare the aforementioned probabilities to determine the message forwarding decision. The probability that at least one node in a community encounter a given node in the network is computed in a privacy preserving manner within the community using the MDTN-Private-Union protocol, also presented in this chapter.

We evaluate E3PR both theoretically by providing a security analysis and practically through extensive simulations. We have conducted our simulations based on a well established community-based mobility model [112, 25]. We compare the performance evaluation of E3PR against four state-of-the-art protocols, i.e., the protocol in [42], epidemic [117], Direct [115], PRoPHET [70], and Bubble [48]. Epidemic and Direct are traditionally considered to achieve the upper and lower bounds of routing performance. PRoPHET and Bubble are representatives in prediction-based and social-based routing protocols, respectively. Results show that E3PR has comparable performance to existing prediction-based protocols while preserving the nodes' privacy.

The remainder of this chapter is structured as follows. We first present our system model in Section 5.2. We then present the E3PR protocol in Section 5.3 followed by the MDTN-Private-Union protocol presented in Section 5.4. We further present our performance evaluation in Section 5.5 and a conclusion in Section 5.6.

5.2 System Model

We assume that nodes belong to communities and that each community has a unique identifier. A node l in each community is designated as the leader of the community. The leader node maintains the list of the nodes in the community. Let set of nodes in a community $C = \{a_1, a_2, \ldots, a_n\}$, where $n = |C|$. We consider a community to comprise of at least three nodes, that is, $n \geq 3$.

Let an event $e_{a,d}$ denotes that a node a encounters a node d. Let $P(e_{a,d})$ be the probability that the event $e_{a,d}$ will happen. For simplicity, we omit the symbol of event e, that is, $P(e_{a,d}) \equiv P_{a,d}$. It is worth noting that our work is focused on the preserving rather than the computing of such probability. Hence, the reader is referred to [70, 85] for further understanding of the computation of such probability. We consider the probability that node a will encounter node d, that is $P_{a,d}$, as private information. Routing protocols can utilize such encounter probability to guide message forwarding. However, nodes require that their private information is not revealed to any other node in the network, which includes fellow nodes in a community.

In this chapter, we consider the semi-honest adversarial model [34]. The nodes in this model always execute the protocol according to the specification. However, adversaries passively attempt to learn the private information of the nodes by using intermediate information gleaned during the execution of the protocol.

5.3 Privacy Preserving Prediction-based Routing

5.3.1 Protocol Description

In this section, we give an overview of E3PR, our Privacy Preserving Prediction-based Routing protocol. A routing example is depicted in Figure 5.1. This figure shows a number of nodes belonging to three communities C_1, C_2 and C_x. A source node s that belongs to the community C_1 wants to send a message to a node d that belongs to the community C_x.

In E3PR, we distinguish the routing inside a community (i.e., intra-community routing) from the routing between communities (i.e., inter-community routing). Specifically, when two nodes that belong to the same community encounter each other, they exchange all the messages they have. On the other hand, if two nodes a_{11} and a_{21} that belong to different communities C_1 and C_2 respectively encounter each other, node a_{11} forwards a message intended for a destination node d to node a_{21}, only if the probability that the nodes in community C_2 will encounter the destination node d is higher than the probability that the nodes in C_1 will encounter

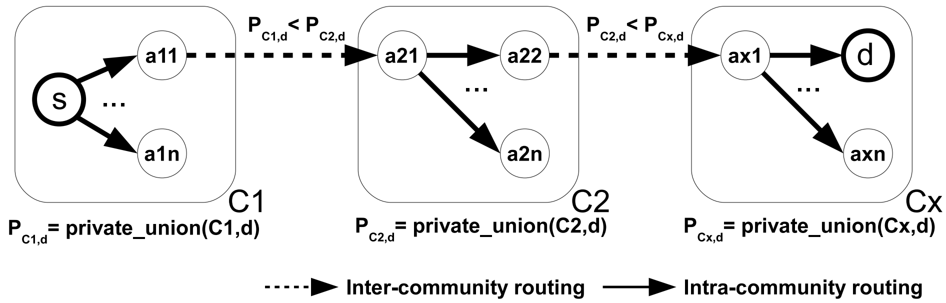


Figure 5.1: E3PR Protocol Overview

d. Let $P_{C_a,d} = union(C_a, d)$ be the probability that at least one node in community C_a will encounter the destination node d. In Figure 5.1, when node a_{11} encounters node a_{21}, node a_{11} forwards a copy of the message intended for d to node a_{21} because $P_{C_2,d} > P_{C_1,d}$.

Summarizing, to route a message from the source node s to the destination node d, the message is first disseminated in an epidemic manner inside the community of the source node. The message then moves from a community to another such that: (1) at each forwarding step, the probability that the message reaches the destination node is increased, (2) as soon as it reaches a community, the message is disseminated in an epidemic manner within the community.

A key characteristic of E3PR is that $P_{C_a,d} = union(C_a, d)$, the probability that at least one node in community C_a will encounter the destination node d, is computed in a privacy preserving manner, that is without revealing the individual probabilities of the nodes in the community. $union(C_a, d)$ is therefore denoted as $private\_union(C_a, d)$ in Figure 5.1.

The E3PR protocol for privacy preserving prediction-based routing in MDTNs is specified in Figure 5.2. The computation of $private\_union(C_a, d)$ is performed using a decentralized protocol for privately computing the union of a set of probabilities in a delay tolerant network without revealing the individual values, i.e., MDTN-Private-Union further described in Section 5.4.

The probability $private\_union(C_a, d)$ is computed periodically in the community independently from the routing protocol. Hence, the only impact it has on the performance of the routing protocol is related to the freshness of the computed probabilities. Therefore, the complexity of the MDTN-Private-Union protocol has no direct impact on the performance of the routing protocol.

Protocol: MDTN-E3PR
Participants: Node a and node b, where $a \notin C_b$ and $b \notin C_a$.
Input: (1) m, a message carried by node a. (2) d, the destination node of message m. (3) C_a, the set of all nodes in the community of node a. (4) C_b. (5) $P_{C_a,d} = union(C_a, d)$, the probability that at least one node in community C_a will encounter node d. (6) $P_{C_b,d}$.
Output: Message m is delivered to node b if $b = d$, or $b \in C_a$, or $P_{C_b,d} > P_{C_a,d}$.
Setup: Node a has a message m whose destination is node d. Node b does not have message m.
Events and Associated Actions:

node a encounters a node b

```
1:  if b = d then
2:      node a sends message m to node b
3:  else
4:      if b ∈ C_a then
5:          node a sends a copy of message m to node b
6:      else
7:          if P_{C_b,d} > P_{C_a,d} then
8:              node a sends a copy of message m to node b
9:          end if
10:     end if
11: end if
```

Figure 5.2: Protocol: MDTN-E3PR

5.3.2 Security Analysis: Correctness

In order to increase the message delivery probability, the conventional prediction-based routing strategy forwards message copies to the nodes which have a higher probability of encountering the destination node than the current message carrier does. We consider our protocol E3PR to be correct if it achieves the same effect as the conventional prediction-based routing strategy.

In E3PR, a node a in community C_a sends message m to an encountering node b in another community C_b if $P_{C_b,d} > P_{C_a,d}$, i.e., if the nodes in community C_b have a higher probability of encountering the destination node d than the nodes in community C_a do (lines 7 and 8). Upon receiving message m, node b disseminates message m inside its community C_a in a flooding manner (lines 4 and 5). According to the definition of community in Section 5.2, the nodes which frequently co-exist in a common location comprise a community. Therefore, a high probability exists of successful message delivery from any node in a community to any other node in the same community. With this in mind, message m reaches all nodes in community C_b with a high probability. Since the nodes in community C_b have a higher probability of encountering node d than the nodes in community C_a do, that is $P_{C_b,d} > P_{C_a,d}$, the protocol achieves a higher delivery probability by forwarding a copy of message m to node b.

5.3.3 Security Analysis: Privacy

In E3PR, a node a only reveals the probability that at least one node in its community C_a will encounter a given node to an outsider node. This probability is computed within the community in a privacy preserving manner using the MDTN-Private-Union protocol, thus individual probabilities of encountering the given node also remain confidential from the nodes inside the community.

One unavoidable side-effect of the protocol is that the adversary learns that node a's probability (i.e., $P_{a,d}$) of encountering the destination node d is no higher than $P_{C_a,d}$. Additionally, assume that node a achieves the maximum probability of encountering the destination node d in its community C. Therefore, the adversary also learns that the maximum probability of encountering node d is no higher than $P_{C_a,d}$. The reader may refer to Section 5.4 for the security analysis of the protocol MDTN-Private-Union.

5.4 Privacy Preserving Computation of Union

5.4.1 Protocol Description

Consider a community $C = \{a_1, a_2, \ldots, a_n\}$, where $n = |C|$. Let $P_{C,d}$ be the probability that at least one node in community C will encounter a given node d. In this section, we present a protocol for computing such probability $P_{C,d}$ in a privacy preserving manner. Let $e_{a,d}$ denote the event that a node a encounters node d, and $\overline{e_{a,d}}$ denote the opposite event of $e_{a,d}$. Let $P(e_{a,d})$ (denoted as $P_{a,d}$ in short) be the probability that event $e_{a,d}$ will happen. Therefore, the probability $P_{C,d}$ can be expressed as Equation (5.1).

$$
\begin{aligned}
P_{C,d} &= P(\textstyle\bigcup_{i=1}^{n} e_{a_i,d}) \\
&= 1 - P(\overline{\textstyle\bigcup_{i=1}^{n} e_{a_i,d}}) \\
&= 1 - P(\textstyle\bigcap_{i=1}^{n} \overline{e_{a_i,d}}) \\
&= 1 - \textstyle\prod_{i=1}^{n} P(\overline{e_{a_i,d}}) \\
&= 1 - \textstyle\prod_{i=1}^{n} (1 - P(e_{a_i,d})) \\
&= 1 - \textstyle\prod_{i=1}^{n} (1 - P_{i,d})
\end{aligned}
\tag{5.1}
$$

Each node in community C submits its individual probability of encountering node d, which is considered as a private information, to the protocol. After the computation of the protocol, each node learns the probability $P_{C,d}$ without disclosing its private information to other nodes. The protocol is specified in Figure 5.3.

The protocol is initiated by the leader node of the community C. The leader node floods an *init* message (Figure 5.3: protocol initiation: line 3) to all nodes in community C. Hereafter, we only concern about the nodes in community C. After receiving the *init* message, a node a can send the *init* message to any encountering node which has not received it yet (INIT: lines 7 and 8). After that, node a exchanges random numbers with each of the first K distinct encountered nodes (INIT: lines 10 and 11). K is a constant and its value is known to all nodes. Node a then mixes its σ_a (initially $\sigma_a = 1 - P_{a,d}$) with the sent and received random numbers (INIT: line 12). After encountering the first K distinct encountered nodes, node a sends the mixed private value to the leader node (INIT: line 14), when it encounters the leader node. The leader node maintains a product of the received mixed private values (PARTIAL: line 2). When the leader node receives all the mixed private values from the nodes in community C, the leader node computes the final result and floods it in its community (PARTIAL: line 4 and 5). The final result is the probability that at least one node in community C will encounter node d.

5.4.2 Protocol Setting

An interesting question is the relationship between the number of nodes in a community and the constant K, that is, what the value of the constant K should be. Recall the stated requirement with regard to the community size in Section 5.2: we consider a community C to comprise of at least three nodes. i.e., $n = |C| \geq 3$. Moreover, according to the mechanism of our protocol, a node at the most can exchange random numbers with all other nodes in its community. Therefore, the domain of the constant K should be $[2, n)$, i.e., $2 \leq K < n$.

In addition, when $K = 2$, whatever the value n is, these n nodes can always make a pair. Therefore, K can always be set as 2. When $2 < K < n$, according to the mechanism of our protocol, each node should exchange random numbers with K distinct nodes in its community. Hence, there are nK random numbers generated in each execution of our protocol. These nK random numbers should be divisible by $K+1$. That is $n(K+1-1) = n(K+1) - n$ is divisible by $K+1$. Therefore, the value of the constant K should meet the following requirement: $n\%(K+1) = 0$. An easy understanding example is that every $K+1$ nodes construct a clique.

Summarizing, the value of the constant K should meet the following two requirements: 1) $2 \leq K < n$ and 2) $K = 2$ or $n\%(K+1) = 0$.

Protocol: MDTN-Private-Union

Participants: Nodes in a community denoted by the set C. One node in C is the leader node denoted by l.

Input: Each node a_i has a private input $P_{i,d}$, that is the probability that node a_i will encounter node d.

Output: The nodes in C learn $\sigma_C = 1 - \prod_{a_i \in C} P_{i,d}$.

Setup: (l, g) uniquely identifies an instance of the protocol, where g is an integer. K is a constant such that $2 \leq K < n$ and $n\%(K+1) = 0$, where $n = |C|$. Nodes are not ordered, that is, a_i denotes any given node in C. ε is a sufficiently small number (i.e., 10^{-5}), which does not affect the accuracy of the computation.

Events and Associated Actions:

leader node l initiates the protocol

$R \leftarrow \phi$
$\sigma_C \leftarrow 1$
l floods $\langle \text{INIT}, l, g \rangle$ to all nodes in C

node $a_i \in C$ receives $\langle$INIT$, l, g \rangle$

$\sigma_i^0 \leftarrow 1 - P_{i,d}$
if $\sigma_i^0 = 0$ **then**
    $\sigma_i^0 \leftarrow \varepsilon$
end if
for $j \leftarrow 1$ **to** K **do**
    a_i encounters node $a_j \in C$
    if a_j has not received $\langle \text{INIT}, l, g \rangle$ **then**
        a_i sends $\langle \text{INIT}, l, g \rangle$ to a_j
    end if
    a_i sends a random positive number r_{ij} to a_j
    a_i receives a random positive number r_{ji} from a_j
    $\sigma_i^j \leftarrow \sigma_i^{j-1} \times \frac{r_{ij}}{r_{ji}}$
end for
a_i sends $\langle \text{PARTIAL}, l, g, \sigma_i^K \rangle$ to l

leader node l receives $\langle$PARTIAL$, l, g, \sigma_i^K \rangle$ from a_i

$R \leftarrow R \cup \{a_i\}$
$\sigma_C \leftarrow \sigma_C \times \sigma_i^K$
if $R = C$ **then**
    $\sigma_C \leftarrow 1 - \sigma_C$
    l floods $\langle \text{FINAL}, l, g, \sigma_C \rangle$ to all nodes in C
end if

Figure 5.3: Protocol: MDTN-Private-Union

5.4.3 Security Analysis: Correctness

The first challenge for the protocol is that the nodes a node will encounter are not known beforehand in MDTNs. To address this challenge, the protocol allows a node $a_i \in C$ to encounter any other K nodes in C (INIT: lines 5 and 6). The encountered nodes by node a_i are given as a_j, where $j \in \{1, 2, \ldots, K\}$.

Each node $a_i \in C$ exchanges random numbers (i.e., the sending random number r_{ij} and the receiving random number r_{ji}) with each of the first K encountered node a_j (INIT: lines 10 and 11). Node a_i multiplies its σ_i with the ratio of random numbers r_{ij} and r_{ji}, whereas node a_j multiplies its σ_j with the ratio of random numbers r_{ji} and r_{ij} (INIT: line 12). When the leader node computes $\sigma_C = \prod_{i=1}^{n} \sigma_i^K$, where $n = |C|$ (PARTIAL: line 2), the product σ_C is the required value $\prod_{i=1}^{n} \sigma_i^0$ because the product of $\prod_{i=1}^{n} \prod_{j=1}^{K} \frac{r_{ij}}{r_{ji}} \times \frac{r_{ji}}{r_{ij}}$ is 1 (PARTIAL: line 2).

Moreover, consider a special case that the σ_i^0 of node a_i is 0. In such a case, the σ_i^K is 0, whereas the exchanged random numbers are positive. Hence, when node a_i sends its σ_i^K to the leader node, the leader node can be aware that the σ_i^0 of node a_i is 0. In other words, the private information of node a_i is disclosed. In order to protect the private information for such kind of nodes. we modify the original σ_i^0 to be a small positive constant ε, if the original σ_i^0 is 0 (INIT: lines 2 – 3).

The result of σ_C (PARTIAL: line 2) will be influenced due to such modification. In order to investigate the impact of the modification on the original result, consider there are s, where $0 \leq s \leq n$, nodes whose original σ^0 are 0. Let σ_C is the result without the modification. Let $\sigma_C^{'}$ be the result with the modification. The value of $\sigma_C^{'}$ is $\varepsilon^s \prod_{j=1}^{n-s} \sigma_j^0$. If $s = 0$, $\sigma_C^{'} = \sigma_C$; otherwise, the value of σ_C is then 0, and $\sigma_C^{'} - \sigma_C = \sigma_C^{'}$. However, the value of ε is so small that can be neglected. Hence, the value of $\sigma_C^{'} - \sigma_C = \sigma_C^{'}$ can be neglected.

The second set of related challenges of mobile delay tolerant network environments are as follows: connectivity is intermittent, messages may arrive after long and variable delays, and message transmission is asynchronous. Moreover, the MDTN-Private-Union protocol is based on community, while the community structure may change in the computation process of the protocol. For instance, some nodes in a community may leave the community, after the leader node initiates the computation of the protocol and before the computation is finished. Therefore, according to whether the community structure is changed or not during the computation of the protocol, we analyze the elements of the protocol that address this set of challenges in the following two cases.

In the case that the community structure does not change in the computation process, the following two elements of the protocol address the above set of challenges: (1) The *init* message reaches all nodes in community C with high probability and thus they all participate in the protocol. This is because that the nodes which frequently co-exist in a common location comprise a community. Therefore, a high probability exists of successful message delivery from any node in a community to any other node in the same community. (2) If a node $a_i \in C$ that has received the *init* message encounters a node $a_j \in C$ that has not yet received the *init* message then a_i sends a copy of the message to a_j to initiate it to the protocol (INIT: lines 7 and 8). Nodes consider an encounter successful only if they exchange all messages (i.e., a sending number and a receiving number) according to the specification during their period of contact. Otherwise, they ignore any partial messages sent and received.

In the case that the community structure changes in the computation process, the above set of challenges are addressed by the following element of the protocol: the protocol is invoked periodically in a community independently from the routing protocol. Thus, even if the computation of MDTN-Private-Union cannot be finished due to the change of the community structure, the nodes in that community can still use the previous results of the protocol to guide the message forwarding police.

5.4.4 Security Analysis: Privacy

Without loss of generality, let's consider a node $a_i \in C$. In an ideal protocol, the node would submit its private value $P_{i,d}$ to a TTP. The TTP is considered trustworthy, therefore it would not disclose the private value $P_{i,d}$ of node a_i to any other party. It would only reveal the output of the protocol, which is the union of the private values received from all the nodes in community C.

In the MDTN-Private-Union protocol, node a_i discloses the following information: (1) a random positive number to each of the K nodes that it encounters after receiving the *init* message (INIT: line 10); (2) the value σ_i^K to the leader node l (INIT: line 14).

For the random positive numbers r_{ij}, where $1 \leq j \leq K$, since these numbers are independent of $P_{i,d}$, the encountered nodes do not learn any information about $P_{i,d}$.

With regard to $\sigma_i^K = \sigma_i^0 \times \gamma_i$, where $\gamma_i = \prod_{j=1}^{K} r_{ij}/r_{ji}$, let's assume that the interval of the random numbers is large compared to the interval of $P_{i,d}$ and that the random positive numbers are distributed uniformly. This implies that the interval of γ_i is also large and that it is distributed uniformly. Thus, the adversary can learn no information about $P_{i,d}$ from σ_i^K.

The adversary can learn $P_{i,d}$ if and only if it learns γ_i in addition to σ_i^K. To learn γ_i, the adversary must learn all values r_{ij} and r_{ji}. This is possible only if all K nodes a_j that encountered node a_i are dishonest and collude to reveal all of their individual r_{ij} and r_{ji} values and consequently the value of γ_i^K. The σ_i^K can be learned by the adversary, if the leader node is dishonest and colludes with the adversary (i.e., the K dishonest nodes).

After understanding the context where the private value of node a_i can be disclosed due to the collusion of dishonest nodes, an interesting question is the probability that such event happens. Let P_D denote the probability that the private value of a node a_i is disclosed by the collusion of dishonest nodes. According to the above analysis, we can see that $P_D = Prob\{\text{leader node } l \text{ is dishonest}\} \times Prob\{K \text{ encounters are dishonest}\}$. Hence, P_D depends on the number of nodes in community C, the value of K, and the number of dishonest nodes in community C. In order to identify P_D, we assume that the number of dishonest nodes excluding node a_i is known and denoted as m, where $0 \leq m \leq n-1$.

Let's assume that each node excluding node a_i in C has the same chance to be dishonest. Hence, $Prob\{\text{leader node } l \text{ is dishonest}\}$ can be expressed as Equation (5.2).

$$Prob\{\text{leader node } l \text{ is dishonest}\} = \frac{m}{n-1} \tag{5.2}$$

Moreover, due to the random mobility model, we can probably assume that the encounters are random and cannot be scripted by the adversary. Hence,

$$Prob\{\text{K encounters are dishonest}\} = \begin{cases} 0, & \text{if } 0 \leq m < K \\ \frac{C_m^K}{C_{n-1}^K}, & \text{if } K \leq m \leq n-1 \end{cases} \tag{5.3}$$

Combining (5.2) and (5.3), the probability P_D can then be expressed as Equation (5.4).

$$P_D = \begin{cases} 0, & \text{if } 0 \leq m < K \\ \frac{m}{n-1} \times \frac{C_m^K}{C_{n-1}^K}, & \text{if } K \leq m \leq n-1 \end{cases} \tag{5.4}$$

The impact of m and K on the probability that the privacy of nodes is disclosed is illustrated in Figure 5.4. It can bee seen that (1) P_D decreases as K increases; (2) P_D increases as m increases. Moreover, we observe that P_D is high when m is chosen a big number.

In addition, one unavoidable side-effect of the protocol is that the adversary learns that node a_i's probability (i.e., $P_{i,d}$) of encountering the destination node d is not higher than $P_{C,d}$, since $P_{C,d} = P(\bigcup_{z=1}^{n} e_{z,d}) \geq P_{i,d}$, where $n = |C|$, $1 \leq i \leq n$. Furthermore, assume that node a_i achieves the maximum probability of encountering the destination node d in its community C. Therefore, the adversary also learns that the maximum probability of encountering node d is not higher than

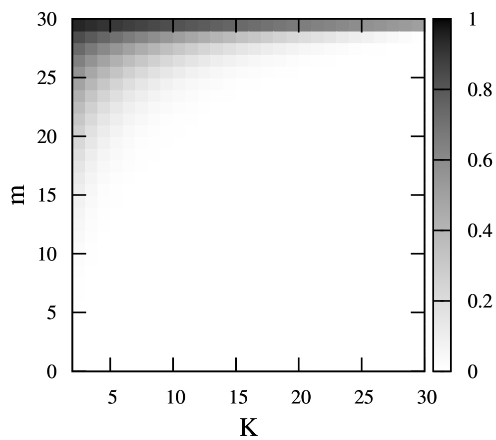


Figure 5.4: The impact of m and k on the privacy, where $n = 31$

$P_{C_a,d}$. However, the adversary can learn whether node a_i is the one who has the maximum probability of encountering node d, no better than a random guess with probability $1/(n-m)$. Moreover, the adversary can learn the exact value of the maximum probability of encountering node d, no better than a random guess with probability $\frac{1}{n-m}P_D$.

As in the ideal protocol, the output of the protocol is the union of the private values of all nodes in C. The MDTN-Private-Union protocol thus does not reveal any more information about the private value $P_{i,d}$ of node a_i than the ideal protocol if the following assumptions hold true: (1) the interval of the random numbers r_{ij} and r_{ji} is large compared to the interval of $P_{i,d}$ and the random numbers are distributed uniformly, and (2) at least one of the K nodes that encountered node a_i and the leader node is honest.

5.4.5 Performance Analysis: Running Time

In this section, we first develop an analytical model of the running time of E3PR. We then simulate the analytical model by numerical results.

Analytical Model

The running time of our protocol is defined as the time since the leader node initiates the protocol until all nodes receive the result of our protocol. Consider a community $C = \{a_i | 1 \leq i \leq M+1\}$. According to the mechanism of E3PR, the running time is spent in four sub-processes: (1) the leader node floods *init* message to all other nodes; (2) each node sprays K random values, each of which can be considered a message, to the first K distinct nodes; (3) each node directly sends the mixed value to the leader node; and (4) the leader node floods the final result to all other nodes. The time spent in each sub-process is characterized by the message dissemination manner. Let D_f denote the delivery delay in the flooding manner. Let D_s^i be the delivery delay of node a_i by spraying K messages. Let D_d be the delivery delay of direct delivery. In addition, since sub-processes (1) and (4) utilize the same manner to disseminate messages, the time consumed in these two sub-processes are the same.

Let $t_1 \leq t_2 \leq \ldots \leq t_{M+1}$ be the time when each node receives the *init* message. t_1 is the time when the leader node l initiates the protocol. Let τ_i, where $1 \leq i \leq M+1$, be the time duration that each node a_i spends to spray K random values to K distinct nodes. The earlier a node receives a *init* message, the more nodes that it can spray its K random values. In other words, it spends less time to spray all of its K random values, that is $\tau_i \leq \tau_j$, where $i \leq j$.

Let $t_i^l = t_i + \tau_i + \tau_{EM}$, where $2 \leq i \leq M+1$, be the time when a community member a_i sends its mixed private value to the leader node. Hence, $t_2^l \leq t_3^l \leq \cdots \leq t_{M+1}^l$. In other words, the time when the leader node l starts to flood the finial result inside community C depends on the last node (denoted by a_M) receiving the *init* message. Consequently, the running time of our protocol is estimated by considering the node lastly receiving the *init* message. The time consumed by our protocol can be expressed as Equation 5.5.

$$D_{3pr} = 2D_f + D_s^{M+1} + D_d \tag{5.5}$$

Since the time spent in sub-process (3) is equal to the inter-meeting time of a node pair, we focus on investigating the time spent in sub-processes (1), (2) and (4). We model the message dissemination in sub-process (1), (2) and (4) by a one-dimensional Continuous Time Markov Chain (CTMC) with state $(n(t))$, where $n(t)$ represents the number of community members with a given message in sub-processes (1) and (4), or community members have been encountered by a given node in sub-process (2), by time t. The state transition is illustrated in Figure 5.5. There are N transient states and 1 absorbing state. The transition from the current state to another different state is characterized by the *transition rate* between these two states, which measures how quickly the state transition happens. According to

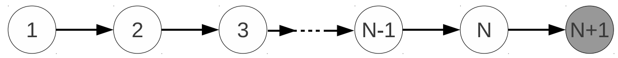


Figure 5.5: The continuous time Markov model for modeling the general message dissemination in E3PR . States (1) to (N) are N transition states and state $(N+1)$ is the absorbing state.

Figure 5.5, we can obtain the generator matrix $\mathbf{Q}$ with dimension $N+1$ as Equation 5.6.

$$\mathbf{Q} = \begin{pmatrix} \mathbf{T} & \mathbf{R} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}. \tag{5.6}$$

where sub-matrix $\mathbf{T}$ is an $N \times N$ matrix with element $T_{i,j}, (1 \leq i, j \leq N)$ meaning the transition rate from transient state (i) to transient state (j). $\mathbf{R}$ is a $N \times 1$ matrix with element $R_{i,N+1}$ means the transition rate from transient state (i) to the absorbing state $(N+1)$. The left $\mathbf{0}$ matrix is a $1 \times N$ vector with all element 0 meaning zero transition rates from the absorbing state to transient states. The right $\mathbf{0}$ matrix includes only a single element 0 representing the negative sum of the left $\mathbf{0}$ vector. According to the different manners of disseminating messages, we obtain the transition rate $q_{i,j}$ from state (i) to state (j) as follows.

i) Flooding Messages. In such a case, $N = M$. In the flooding process, each of the nodes with a given message replicates the message to an encountering node without the message. Consider the current state is in state $(n)_{n \leq M}$, that is, there are n nodes including the leader node with the message and $M+1-n$ nodes without the message. When one of the nodes without the message encounters a node with the message, the system state turns to state $(n+1)$. The transition rate from state (n) to state $(n+1)$ is $(M+1-n)n\lambda$, since there are $M+1-n$ nodes that can receive the message from n nodes in the rate of λ. When the last node receives the message, the system turns to state $(N+1)$, (i.e., the absorbing state), and the transition rate is $M\lambda$. Let $\mathbf{T}_f$ and $\mathbf{R}_f$ denote the matrix $\mathbf{T}$ and $\mathbf{R}$ in $\mathbf{Q}$, respectively. Let $T_f\{(j)|(i)\}$ and $R_f\{(N+1)|(i)\}$ be the tradition rate from state (i) to state (j) and from state (i) to the absorbing state, respectively. Therefore, the non-zero transition rates in the Markov chain can be expressed as Equation 5.7.

$$\begin{cases} T_f\{(n+1)|(n)\} = (M+1-n)n\lambda, n \in [1, M-1] \\ R_f\{(N+1)|(n)\} = M\lambda, \text{ if } n = M \\ T_f\{(n)|(n)\} = -T_f\{(n+1)|(n)\} - R_f\{(N+1)|(n)\}, \\ \qquad n \in [1, M] \end{cases} \tag{5.7}$$

ii) Spraying Messages. In such a case, $N = K-1$. From the viewpoint of the node a_{M+1} lastly receiving the *init* message, only node a_{M+1} can forward its random values to the remaining K available nodes. Consider the current state is in state $(n)_{(n \leq K-1)}$, that is, there are n distinct nodes in the remaining K nodes which have received random values from node u, and $K-n$ nodes have not received random values from node a_{M+1}. When node a_{M+1} has not sent all K random values and encounters one of the remaining nodes without random values from node a_{M+1}, the system state turns to state $(n+1)$. The transition rate from state (n) to state $(n+1)$ is $(K-n)\lambda$, since there are $K-n$ community members only which can receive the remaining random values from node a_{M+1} in the rate of λ. When the K-th random value is sent by node a_{M+1}, the system turns to state $(N+1)$, (i.e., the absorbing state), and the transition rate is λ. Let $\mathbf{T}_s$ and $\mathbf{R}_s$ denote the matrix $\mathbf{T}$ and $\mathbf{R}$ in $\mathbf{Q}$, respectively. Let $T_s\{(j)|(i)\}$ and $R_s\{(K)|(i)\}$ be the tradition rate from state (i) to state (j) and from state (i) to the absorbing state, respectively. Therefore, the non-zero transition rates in the Markov chain can be expressed as Equation 5.8.

$$
\begin{cases}
T_s\{(n+1)|(n)\} = (K-n)\lambda, n \in [1, K-2] \\
R_s\{(K)|(n)\} = \lambda, \text{ if } n = K-1 \\
T_s\{(n)|(n)\} = -T_s\{(n+1)|(n)\} - R_s\{(K)|(n)\}, \\
\qquad\qquad n \in [1, K-1]
\end{cases}
\tag{5.8}
$$

According to the transition matrix $\mathbf{T}_{f(s)}$, we can derive the message delivery delay, dented by $\mathrm{D}_{f(s)}$, as the following expression [54, 67]:

$$
D_{f(s)} = \mathbf{e} \cdot (-\mathbf{T}_{f(s)}^{-1}) \cdot \mathbf{I} \tag{5.9}
$$

where $\mathbf{e}$ is a $1 \times N$ vector denoting the initial state probability vector $\mathbf{e} = [1, 0, \ldots, 0]$, and I is a $N \times 1$ all-one vector, that is, $\mathbf{I} = [1, 1, \ldots, 1]^T$.

Numerical Results

In this section, we give numerical results of the running time based on the analytical model. The simulations are conducted under a widely utilized mobility model in MDTNs called Random WayPoint (RWP) [52]. From the perspective of the whole network space, the mobility characteristics of RWP have been shown to be very different from human movement in real-life scenarios by recent studies [48, 45, 10]. However, by virtue of its simplicity and mathematical tractability, which in turn enable to theoretically analyze the performance bounds, RWP is widely utilized by the existing community-based mobility models [45, 112] to imitate the node mobility inside a community.

In RWP, each node is initially specified a random destination within a given area, and it then moves towards the destination with a given speed. When it reaches

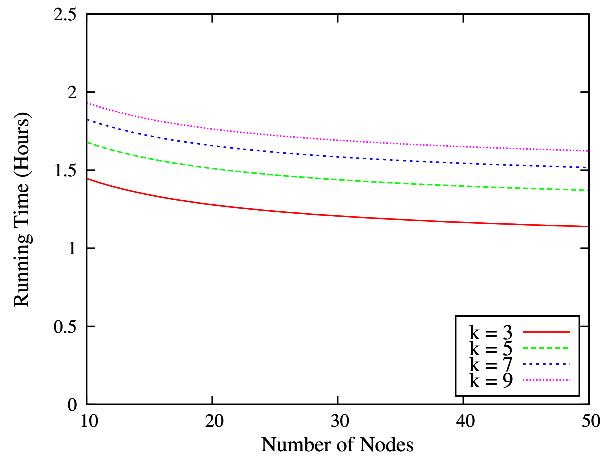


Figure 5.6: Simulation Results of the running time of E3PR with a given contact rate $\lambda = 2.51$ h^{-1}.

the destination, it stays there for a certain period of time (i.e., a pause time). When the pause time expires, it randomly chooses a new destination, and repeats the above process. In order to avoid the impact of the setting of pause time on the routing performance, there is no pause time in the simulations as done by [69, 68].

We utilize the similar simulation settings as in [54, 69, 68], in which a standard RWP [52] mobility scenario is considered. The parameters in our numerical evaluation include the contact rate λ, the number of nodes $M+1$ in a community, and the value of K. The contact rate λ is obtained according to the study of Groenevelt et al. in [37]. Specifically, the contact rate λ of nodes under RWP mobility is $(8wrv)/\pi L^2$, where $w = 1.3683$ is a constant, r is the transmission range, v is the speed of nodes, L is the size of the network. In our numerical simulation, the parameters are set as follows: $\lambda = 2.51$ h^{-1} with $r = 20m$, $v = 10$ m/s, and $L = 1000$ m. The number of nodes and the value of K are set as variable parameters for the investigation.

The numerical results are illustrated in Figure 5.6. From the results, we can see that running time of the protocol decreases, as the number of nodes in a community increases. The greater number of nodes provide more opportunities of relaying flooded message and participate in sharing random values. For a given number of nodes, the greater value of K, the longer the running time is. It results from that the node last receiving the *init* message spends more time to spray the K random values to the remaining K distinct nodes.

5.5 Experimental Evaluation

We present in this section the performance evaluation of E3PR. We start by presenting the simulation settings and the mobility model we have used in sections 5.5.1 and 5.5.2, respectively. We then introduce the routing protocols against which we compare the performance of E3PR and the performance metrics we used in sections 5.5.3 and 5.5.4, respectively. Finally, we present the results of our experiments in Section 5.5.5. As none of the non-naive algorithms against which we compare E3PR are privacy preserving, the objective of this performance evaluation is to assess the cost of introducing privacy preservation mechanisms in the routing process.

5.5.1 Simulation Settings

We have implemented E3PR as a module of the Opportunistic Network Environment simulator (ONE) [57]. We summarized the simulation parameters that we used.

We have used a simulation area of 2000×1500 m^2. This area is equally divided into 12 regions as shown in Figure 5.7. In each region we initially deploy a varying number of nodes (from ten to fifty). Each node considers the region in which it has been deployed as its *local region*. According to the mobility model we used, further described below, a node is more likely to visit its local region than other places. Nodes associated to a region constitute a community. This simulation scenario is very similar to the one used in PRoPHET [70].

The communication between nodes is performed using the Bluetooth protocol since modern mobile devices are commonly equipped with this technology. Bluetooth has been often used in the evaluation of DTN protocols. For instance, the reality mining mobility traces [27], which have been used for the evaluation of many protocols, e.g., Habit [78], have been collected with mobile phones using Bluetooth. According to the specification of Bluetooth version 2.0 [57], the transmission range and bandwidth are set as 10 m and 2 Mb/s, respectively. Furthermore, the speed of nodes is set to 1.34 m/s, since this is an average human walking speed [59]. Each experiment we run approximately lasts for thirteen hours (simulation time) among which one hour is a warm up period during which no message is generated. After this period, every thirty seconds, a random node sends a message to random destination node. We have considered only messages for which the source and the destination belong to different communities.

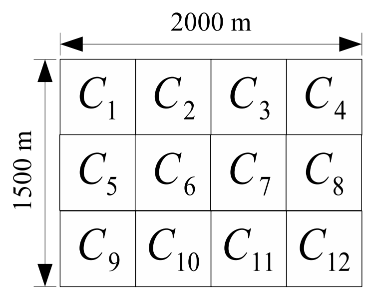


Figure 5.7: Community Model

Table 5.1: Parameter settings

Parameter Name	**Value**
Simulation area	$2000 \times 1500\ m^2$
Transmission range	10 m
Simulation duration	13 hours + TTL
Warm-up period	1 hour
Message generation rate	1 message per 30 seconds
Number of communities	12
Number of nodes in a community	from 10 to 50
Node speed	1.34 m/s
p_l	0.8
p_r	0.2

5.5.2 Mobility Model

In our evaluation, we adopt the community-based mobility model proposed in [112], which has been widely utilized for the evaluation of community-based routing protocols [114, 25]. In this mobility model, each community is associated with a geographical area. The movement of node i, which belongs to the community C_i consists of a sequence of *local* and *roaming* epochs. A local epoch is a random direction movement restricted inside the area associated with the community C_i. A roaming epoch is a random direction movement inside the entire network. If the previous epoch of a node i was a local one, the next epoch is a local one with probability p_l, or a roaming epoch with probability $1-p_l$. Similarly, if the previous epoch of node i was a roaming one, the next epoch is a roaming one with probability p_r, or a local one with probability $1-p_r$. In our simulations, we adopt the same values for p_l and p_r as in [70], i.e., p_l=0.8 and p_r=0.2.

5.5.3 Routing Protocols

We have compared the performance of E3PR against the following protocols:

Epidemic: in this protocol, a node forwards a copy of each unexpired message it holds to every node it encounters, which does not already have a copy of the message. Epidemic routing achieves the upper bounds of delivery ratio and delivery cost, and achieves the lower bound of delivery latency.

Direct: in this protocol, the source node only forwards the message to the destination node. Contrary to Epidemic, Direct routing achieve the lower bounds of delivery ratio and delivery cost, and achieves the upper bound of delivery latency.

PRoPHET: in this protocol, a node forwards a copy of a message it holds to a node it encounters, only if the latter has a higher probability of encountering the destination node of the message. The parameters of the protocol are set as described in [70]. PRoPHET is a well known prediction-based routing protocol.

Bubble: this protocol utilizes social information about nodes, such as their centrality and the community to which they belong. There are two kinds of centrality in this protocol: local centrality and global centrality. The local (global) centrality value of a node indicates the number of its community members (nodes) that it encountered in time windows. In this protocol, a message is forwarded based on the values of the values of the global centrality of two encountering nodes, until it reaches a node in the same community as the destination node. After that, the message is forwarded based on the values of the local centrality of two encountering nodes, until it either reaches the destination node or expires. In our simulations, considering the TTLs of messages, the size of a time window is set to 1 hour. The centrality value of a node is accumulated in all time windows. Moreover, Bubble is a well known community-based routing protocol.

3PR: in this protocol, message forwarding decision is made by comparing information about communities of nodes instead of individual nodes. Specifically, it compares the maximum probability that a node in the community of a potential intermediate node will encounter the destination node. The parameters of the protocol are set as described in [42].

We have compared the performance of E3PR against this set of algorithms for the following reasons. First, Epidemic and Direct are often used to show the upper and the lower bound in terms of performance, that can be reached in a given environment. Then, as E3PR is a prediction-, community-based algorithm, we used PRoPHET and Bubble as the representative algorithms for the categories of prediction-based and community-based algorithms, respectively.

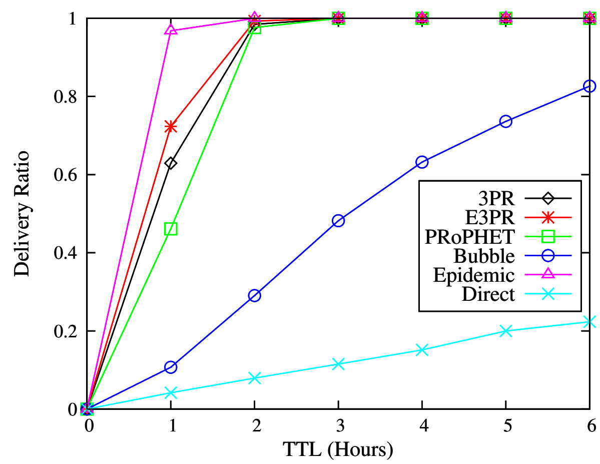


Figure 5.8: Delivery ratio wrt the increasing TTL of messages.

5.5.4 Performance Metrics

To evaluate E3PR we used three well known metrics: the delivery ratio, the delivery cost and the delivery latency defined as follows.

Delivery ratio: is the proportion of messages that have been delivered out of the total unique messages created.

Delivery cost: is the total number of messages transmitted in the simulation. To normalize this, we divide it by the total number of unique messages created.

Delivery latency: is the average time needed to finish transmitting messages to their destinations.

5.5.5 Performance Results

We performed two experiments. First, we compared the performance of E3PR against the protocols introduced above, with respect to the above three performance metrics. We then analyze the impact of the community size on the performance of E3PR.

Performance Comparison of Routing Protocols

Figure 5.8 shows the delivery ratio of the compared protocols as a function of the Time-To-Live (TTL) of the generated messages. As expected, Epidemic and Direct achieve the best and worse delivery ratio, respectively, for all values of TTL. We also observe that E3PR achieves a better delivery ratio than PRoPHET and 3PR when the TTL is less then 2 hours, and achieves a similar delivery ratio to that of

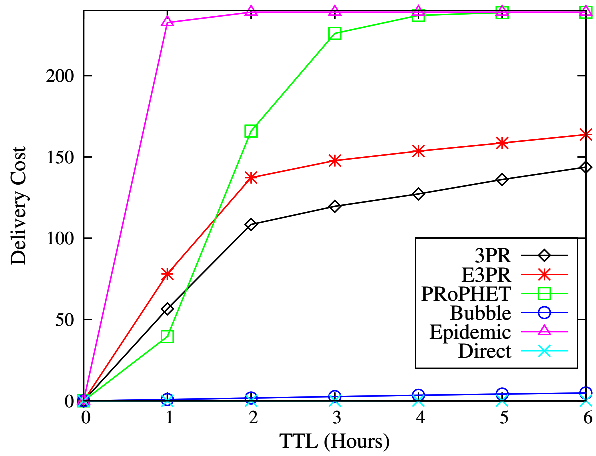


Figure 5.9: Delivery cost wrt the increasing TTL of messages.

PRoPHET and 3PR when the TTL is greater than 2 hours. Finally, E3PR has a much higher delivery ratio than Bubble. The difference between the performance of the two protocols gets up to 70.29% for a TTL of 2 hours. This is because E3PR floods a message inside the communities which are on the path from the community of its source node to the community of its destination node.

Figure 5.9, shows the delivery cost of the compared routing protocols. As expected, Epidemic and Direct have the highest and lowest delivery cost, respectively, whatever the value of TTL. Compared to the others, Bubble has a low delivery cost, which remains stable when the TTL increases. The delivery cost of E3PR is higher than that of Bubble and 3PR, but much lower than the one of PRoPHET.

Figure 5.10 shows the delivery latency of the compared routing protocols. Epidemic has the lowest delivery latency, whatever the TTL. Further, E3PR follows the same trend as Epidemic with higher latencies (around 0.29 hour). 3PR and PRoPHET achieve a little higher delivery latency than E3PR. The performance of Bubble and Direct increases linearly with the increase of the TTL.

Influence of the Number of Nodes in a Community

In order to investigate the impact of the number of nodes in each community on the routing performance of our protocol, we run an experiment in which we vary the number of nodes in each community from 10 to 50.

Figure 5.11a, 5.11b and 5.11c show the impact of the increasing community size on the delivery ratio, the delivery cost and the delivery latency, respectively. Results show that the larger the communities, the higher the delivery ratio and

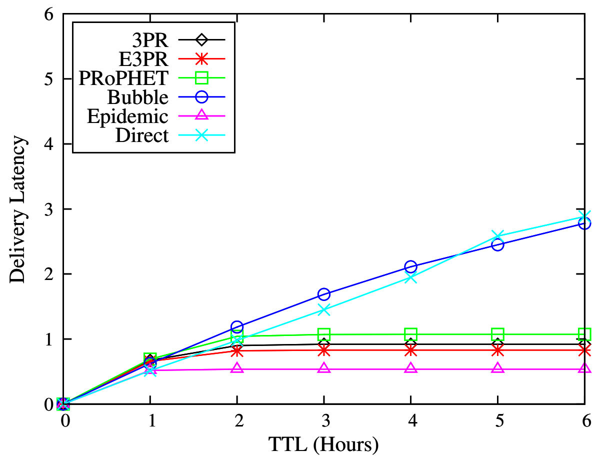


Figure 5.10: Delivery latency wrt the increasing TTL of messages.

cost and the lower the delivery latency. Since E3PR floods a message inside the community of the message carriers, the delivery cost increase as the communities become larger. However, more message copies increase the delivery probability and reduce the delivery latency.

Impact of the Settings of the Mobility Model

In this section, we investigate the impact of the settings of the adopted mobility model on the routing performance of E3PR. We run an experiment in which we vary the value of p_l from 0.5 to 0.9 with step by 0.1 and set the value of p_r as $1 - p_l$.

First, we look at the impact of the settings of the adopted mobility model on the delivery ratio. As shown in Figure 5.12, we can observe that E3PR achieves similar results with different settings of p_l and p_r. The performance of delivery ratio increases as the increment of the value of p_l when the TTL is not greater than 3 hours. The performance of delivery ratio with different settings is the same, when the TTL is greater than 3 hours. Since E3PR floods messages inside a community, under the pre-condition that messages can be transferred among communities, the higher probability that a node stays inside its community, the higher probability that the node gets a message flooded inside its community.

Next, we compare the delivery cost of E3PR with different settings of the adopted mobility model. From the results illustrated in Figure 5.13, we can observe that the performance of delivery cost increases as the value of p_l increases when the TTL is not greater than 3 hours. When the TTL is greater than 3 hours, the

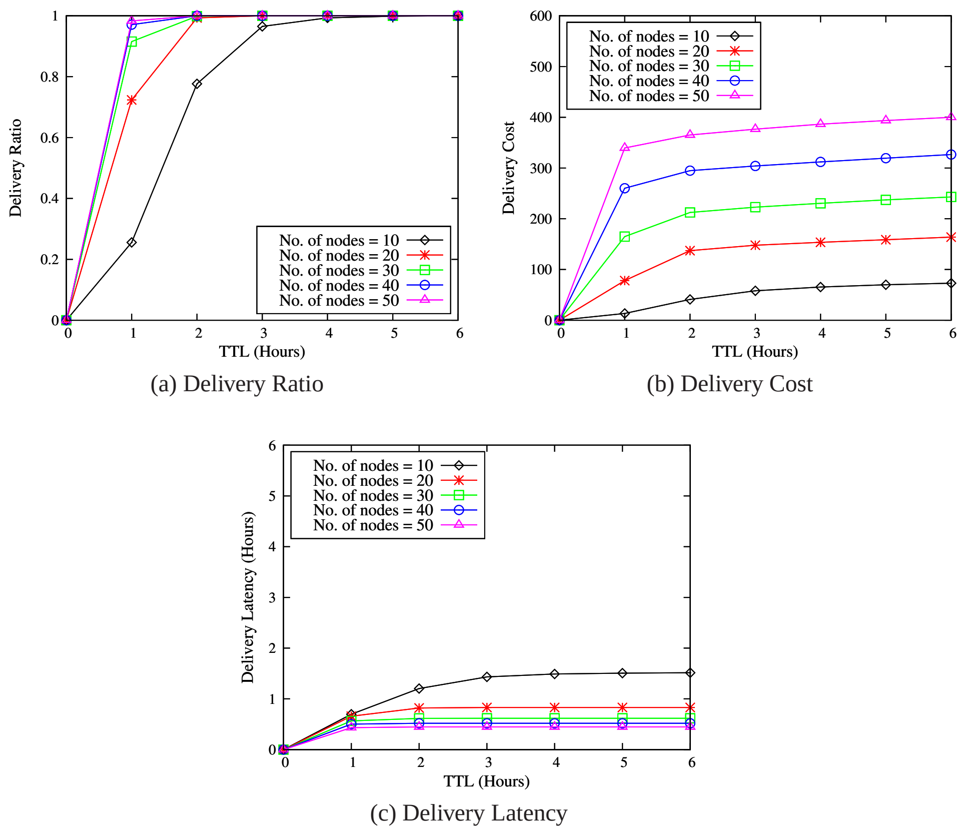


Figure 5.11: (a) delivery ratio, (b) delivery cost, and (c) delivery latency wrt the increasing size of communities.

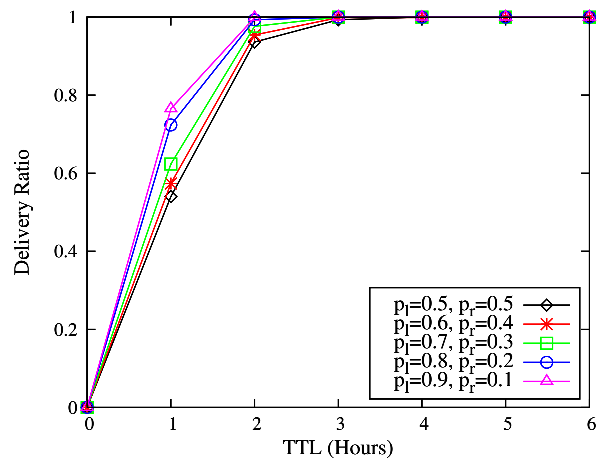


Figure 5.12: The impact of the settings of the mobility model on the delivery ratio of E3PR

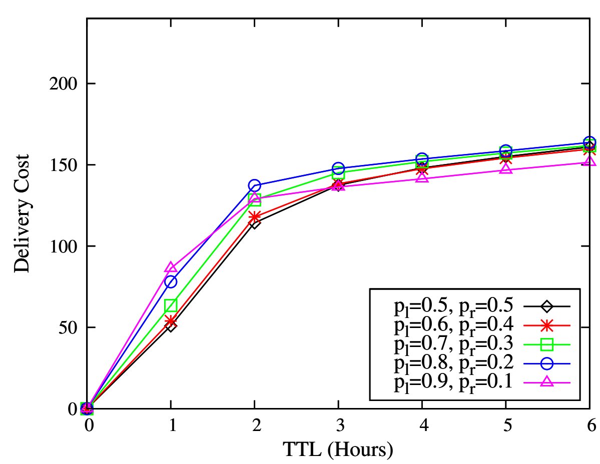


Figure 5.13: The impact of the settings of the mobility model on the delivery cost of E3PR

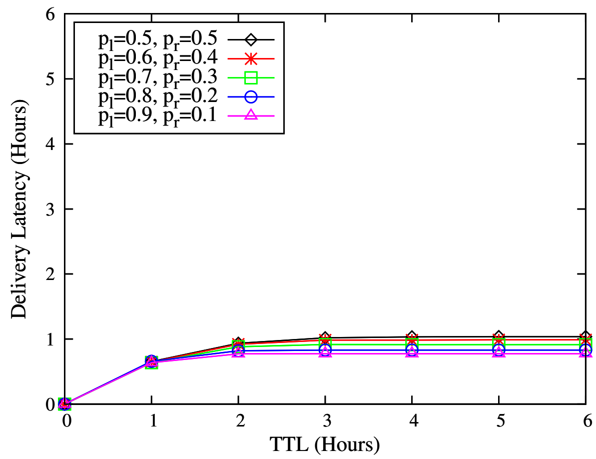


Figure 5.14: The impact of the settings of the mobility model on the delivery latency of E3PR

performance of delivery cost decreases as the increment of the value of p_l. This is because that the higher probability that a node stays inside its community, the higher probability that the node gets a message flooded inside its community. In our case, for a given message, most of nodes on the routing path from the community of its source node to the community of its destination node can get a copy of the message within 3 hours. Therefore, when the TTL is greater than 3 hours, the delivery cost increases slowly for the simulations with high values of p_l. This is consistent with the results of the delivery ratio.

At last, we investigate the results of delivery latency of (E3PR) with different settings of the adopted mobility model. As shown in Figure 5.14, we can see that the delivery latency decreases as the increment of p_l. For each setting, the delivery latency increases as the TTL increases, when the TTL is less than 3 hours; the delivery latency keeps the same as the TTL increase, when the TTL is greater than 3 hours. For the case that the TTL is less than 3 hours, the messages that need more time can be delivered as the TTL increases. As for the case that the TTL is greater than 3 hours, the latency keeps the same, since the messages are delivered within 3 hours. Note that this is consistent with the results of the delivery ratio.

5.6 Chapter Review

In this chapter, we presented E3PR, the first privacy-preserving prediction-based routing protocol for mobile delay tolerant networks. E3PR takes advantage of the mobility patterns of nodes to route messages, yet preserves the privacy of nodes by hiding their individual mobility patterns. The protocol requires that the nodes in a community compute the probability that at least one of the nodes in the community will encounter a destination node. We presented a protocol that computes this union in mobile delay tolerant networks in such a manner that the individual private values are not revealed even to the nodes inside the community. We evaluated E3PR both theoretically, with correctness and privacy analyses, and practically, through extensive simulations. Our simulations on a well established community-based mobility model, demonstrate that E3PR has comparable performance to existing prediction-based protocols, while preserving the privacy of nodes.

Chapter 6

Perspective–Selfishness of Nodes in MDTN Routing

6.1 Introduction

In the literature, considerable efforts have been done to deal with routing in MDTNs. Most of the existing routing protocols in MDTNs explicitly or implicitly assume that the nodes in a network are willing to relay messages for others. Unfortunately, reality is different. Indeed, as it has been previously demonstrated in the literature [69], collaborative systems are subject to rational behavior (also called selfish behavior). MDTNs are particularly suited for exacerbating such behavior due to the resource constraints of mobile devices (e.g., battery, memory and bandwidth) [101]. The main contribution of the ongoing work in this chapter is to survey the existing related research works that overcome the issue of nodes' selfishness.

The remainder of this chapter is organized as follows. We first classify selfish behaviors, and summarize the impact of selfish behaviors on routing performance in Section 6.2. We then investigate different strategies for preventing selfish behavior in Section 6.3. This is followed by an experiment to compare the performance of different strategies in Section 6.4. Finally, we conclude this chapter in Section 6.5.

6.2 Selfishness

In this section, we first develop a unified view of the classification of selfish behaviors. We then discuss the methodologies utilized for investigating the influence of selfish behaviors on the performance of routing protocols. Finally, we highlight the performance degradation caused by selfish behaviors.

6.2.1 Classification of Selfish Behavior

Recent years have seen considerable research works addressing the issue of selfish behavior in DTNs [68, 127, 73]. Traditionally, most works consider selfish behavior as the unwillingness of a single node to relay the messages of all other nodes in order to conserve its limited resources. Nevertheless, people in real life (i.e., the carriers of mobile devices) generally do not act alone, but tend to belong to communities [49]. In an alternative type of selfishness, a node that belongs to a community is willing to relay messages for the nodes within the same community but refuses to relay messages for the nodes outside its community. For this reason, selfish behavior can be classified into two categories: *individual selfishness* and *social selfishness* [65].

Moreover, in the literature investigating the impact of selfish behavior on routing performance [95, 54], authors generally consider the following two types of selfish actions: *non-forwarding of messages* and *dropping of messages*. Non-forwarding of messages means that a node refuses to relay messages for the nodes towards which it is selfish. Dropping of messages means that a node agrees to relay messages for the nodes towards which it is selfish, but it drops the messages after receiving them.

From the above description, we can see that there are two classifications of selfish behavior from different aspects. In this chapter, we develop a unified view of the classification of selfish behavior. We term the two aspects of the classification as *collusion* and *non-cooperation*. From the viewpoint of collusion, selfish behavior can be classified into two categories: individual selfishness and social selfishness. From the viewpoint of non-cooperation, selfish behavior can be classified into two categories as well: non-forwarding of messages and dropping of messages. The reader is requested to refer to Figure 6.1 for an illustration of the unified view of the classification. To the best of our knowledge, this is the first work to develop this unified view of the classification of selfish behavior.

6.2.2 The Methodologies of Investigating the Impact of Selfish Behavior

Since Panagakis et al. [95] first presented their study on the performance degradation caused by selfish behavior in DTNs, researchers have shown significant interest in this field. To evaluate the impact of selfish behavior on the performance of

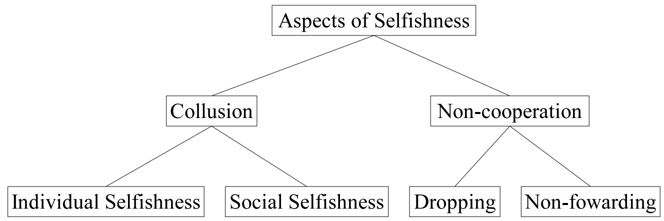


Figure 6.1: Classification of selfish behavior in DTNs

existing routing protocols, some works utilize theoretical analysis models, such as Continuous Time Markov Chains (CTMC), whereas others utilize simulations.

To the best of our knowledge, CTMC is first exploited by Karaliopoulos et al. [54] to demonstrate the impact of selfish behavior in DTNs. Later studies [67, 66, 69, 68] explored CTMC to show the influence of selfish nodes on routing performance in the contexts of social selfishness, constrained energy and multicast routing. CTMC provides a theoretical approach for analyzing selfish behavior in DTNs.

However, CTMC can only be utilized to model the routing process of simple routing protocols, such as Epidemic [117] or Spray and Wait [111]. These routing protocols are generally considered to be inefficient in practice [80]. In addition, studies based on CTMC do not evaluate the performance of the routing protocols in terms of delivery ratio, which is traditionally considered to be the most important performance metric in DTNs. Therefore, authors in [58, 69, 22] utilize simulation methods to investigate the influence of selfish behavior on the routing performance.

6.2.3 The Impact of Selfish Behavior

Existing research works [68, 69, 58, 22] based on theoretical analysis and experimental simulations reveal the following two characteristics of the impact of selfish behavior on the routing performance. Firstly, the routing performance (i.e., delivery ratio, delivery cost and delivery latency) is seriously degraded, if a major portion of the nodes in the network is selfish. For instance, the delivery ratio in the presence of selfish nodes can be as low as 20% compared to what can be achieved under full cooperation [108]. Secondly, the impact on the routing performance is related to the non-cooperative action of selfish behavior (i.e., non-forwarding of messages and dropping of messages). Specifically, the behavior of non-forwarding messages reduces the delivery cost, while the behavior of dropping of messages increases the delivery cost. However, both of them decrease the delivery ratio, and prolong the delivery latency, even if messages are eventually delivered.

6.3 Strategies for Preventing Selfish Behavior

In order to reduce the impact of selfish behavior on routing performance, a number of studies focus on stimulating selfish nodes to be cooperative. The existing incentive strategies are traditionally classified into three categories [16, 21]: barter-based [16, 15, 121], credit-based [21, 93, 127, 73, 126] and reputation-based [7, 118, 26, 120, 73, 64]. In the following subsections, we will introduce the representative strategies in each category and summarize their common problems.

6.3.1 Barter-based Strategies

The simplest strategies are barter-based or pair-wise Tit-For-Tat (TFT) strategies [16, 15, 121]. The mechanism is that two encountering nodes exchange the same amount of messages. In [16, 15], the authors divide the messages into two categories: primary messages and secondary messages. For a given node, the messages in which it is interested (e.g., the messages destined for it) are primary messages. Other messages are secondary messages. When two nodes encounter each other, they first exchange the description about the messages stored in their buffers. Based on the analysis of the description, each node determines an initial list of the desired messages from the other node, and sorts the messages in order of preference (i.e, the priority of primary messages is higher than that of secondary messages). For the sake of simplicity, let us assume that the size of messages is the same. Finally, each node refines the list by keeping the top K messages in its initial list, where K is the minimum size of two initial lists.

From the above depiction of message selection under this strategy, we can see that it is entirely up to the nodes to determine the desired messages. Thus, a node may adopt selfish behavior towards the secondary messages, in order to conserve its limited resources. However, exchanging the secondary messages is also beneficial, since they can be used to exchange the primary messages in the future. In other words, each message has a potential value, which is employed to prevent selfish behavior. Moreover, the authors in [16, 121] consider the message selection process as a two-person game, and utilize a Nash Equilibrium [87] to increase the message delivery ratio.

After the message selection process, two encountered nodes exchange the messages in the lists one by one (i.e., if a node has sent a message to the other node, it would not send another message, until it receives a message from the latter). In such a manner, even if the connection is disrupted during the exchange process, the maximum difference of the number of exchanged messages between two nodes is one. Consequently, the fairness of message exchange can be ensured by exchanging approximately the same amount of messages between two encountering nodes.

However, the requirement for exchanging the same amount of messages is a two-edged sword. It can degrade the routing performance dramatically in the case that one of the two encountering nodes has fewer messages. For instance, let's consider that there are two encountering nodes, called node A and B. Node A contains a message whose destination is node B. However, there is no message in the buffer of node B at the moment. In such a case, the message cannot be delivered to node B. Furthermore, if node A is the source of the message, the performance in terms of delivery ratio is even worse than that achieved by utilizing Direct Delivery [115] which is generally considered to achieve the lower bound for the delivery ratio in DTNs.

6.3.2 Credit-based Strategies

Credit-based strategies are proposed to avoid the disadvantages of barter-based strategies. This kind of strategy stimulates nodes to be cooperative by utilizing the concept of virtual credit, which is utilized to pay for message forwarding. The mechanism is that if a node cooperates to forward a message for others, it receives a certain amount of credit as a reward that it can later utilize for its own benefit.

Based on which node is charged with the message forwarding, the credit-based strategies can be further sub-divided into two models [17]: 1) *Message Purse Model* and 2) *Message Trade Model*. In message purse model [127, 73, 21], the source node of a message pays credits to the intermediate nodes which participate in delivering the message to the destination. In the message trade model [93], messages are considered as valuable goods. The receiver pays credits to the sender of a message in each hop-by-hop transmission until the message reaches the destination, which finally pays for the message forwarding. Since the source nodes do not pay for the message forwarding, the message trade model is inherently vulnerable to the source nodes flooding the network. For this reason, most of the credit-based works utilize the message purse model.

In the strategies that belong to the message purse model, the common assumption is the existence of a Virtual Bank (VB), or Credit Clearance Service (CCS). The VB covers the space that the mobile nodes can reach, and can be connected by any nodes in the network. The responsibility of the VB is to charge the source node of a message and reward the intermediate nodes which participate in delivering the message to the destination.

The strategies [73, 21, 127] belonging to the message purse model are suitable for different routing protocols. In [73], the proposed strategy is designed for the single-copy routing protocols (e.g., Direct Delivery and First Contact [50]) under which only one message copy exists in the routing process. Although single-copy routing protocols consume the least resources, the routing performances in terms of delivery ratio and delivery latency are generally too low to be applicable in practice [114]. Therefore, more routing protocols (e.g., Epidemic and Spray and Wait) are multi-copy based. In [21, 127], the proposed strategies are targeted to multi-copy based routing protocols in DTNs. In [127], Zhu et al. include the solution of cheating actions (i.e., credit forgery attack, nodular tontine attack and submission refusal attack) adopted by the selfish nodes to maximize their benefits. Detailed information about these cheating actions is given in [127].

From the above discussion, we can see that the process of charging and rewarding is invoked at the side of the VB, when (1) a message is successfully delivered to the destination and (2) there are intermediary nodes participating in the routing process. Let's consider a scenario where a major portion of the nodes is selfish and each node has enough credits to request the message forwarding service from an encountering node in a contact. In such a case, a message can only be delivered when the source node directly encounter the destination node. In addition, before the message reaches the destination node, the credits of the source node are reusable to request the message forwarding service. Therefore, a selfish node cannot be aware of the necessity of cooperation with other nodes. Due to the above two reasons, the credit-based strategies cannot efficiently stimulate the selfish nodes to be cooperative, when a major portion of the nodes is selfish.

6.3.3 Reputation-based Strategies

We first explain the concept of reputation before discussing the reputation-based strategies: "Reputation of an agent is a perception regarding its behavior norms, which is held by other agents, based on experiences and observation of its past actions" [71]. In the scope of investigating selfish behavior, the reputation value of a node indicates other nodes' perception about the cooperation of the node. For instance, if the reputation value of a node is low, it means that the node is considered to be selfish by other nodes. If the reputation value of a node is high, it means that the node is considered to be cooperative by other nodes.

The mechanism of this kind of strategy is that a message generated by a given node is forwarded only if the node has forwarded messages originating from others, i.e., has a good reputation. Therefore, the observation about the behavior of other nodes plays a significant role in this kind of strategy. Based on the feasibility of observation by other nodes, we further divide the existing strategies into two models: 1) *detection-based model* and 2) *non-detection model*.

In the detection-based model, each node monitors the behavior of the intermediary nodes. In [120, 64, 7], the authors propose different methods to detect selfish behavior in DTNs. In [120], each intermediate node receives a receipt after forwarding a message to another node. The receipt is a proof about the cooperation of the intermediate node. The cooperation of an encountering node is assessed by Beta distribution, which is parameterized by the number of cooperative and selfish actions taken by the node. However, the strategy cannot prevent collusion cheating, which means that some nodes together cheat other nodes in order to increase their reputation. Detailed information about this cheating action is given in [127]. Similar to [120], the behavior of intermediary nodes is proved by the return of a receipt. The difference is that a receiver floods the receipt instead of sending the receipt to the sender. In [7], selfish behavior is detected in a different way: the sender (including the source and intermediate nodes) of a message keeps the records of the encountered nodes and the forwarding records which contain the identifier of the message, the destination of the message and the forwarding time. When two nodes encounter each other, they check the forwarding records and received messages since last encounter time, in order to detect the cooperative nodes and selfish nodes.

However, due to the unique features of DTNs (e.g., the lack of an end-to-end continuous path and the high variation in network conditions), the detection of selfish behavior is considered to be difficult by some authors. The alternatives belonging to reputation-based strategies are not based on the detection of selfish nodes [26, 73]. In [26], Dini et al. decrease the reputation of all nodes periodically, and only increase the reputation of the intermediate nodes who participate in the successful message delivery. Similar to [26], the proposed strategy in [73] decreases the reputation of all nodes periodically. The differences between them are twofold. First, it involves credit-based incentive strategy to reward the intermediate nodes which participate in the successful message delivery. Second, no matter whether the message delivery succeeds or not, all cooperative nodes can get good reputation values by sending the proofs of collaboration to a Trusted Authority (TA), which is responsible for credit and reputation clearance.

From the above description, we can see that reputation-based strategies can work even if a major portion of the nodes takes the selfish behavior of dropping messages. However, this kind of strategy mistakenly considers the collaboration of intermediate nodes as selfish behavior, if the reason causing the failure of message delivery is the message expiration other than the selfish behavior of intermediate nodes. This is unfair to the cooperative nodes. Furthermore, it results in the decrement of the delivery probability of the messages generated by this kind of cooperative node, since they are mistakenly considered as selfish nodes by other nodes. Moreover, since the reputation-based strategies only check whether an intermediate node forwards the message to other nodes or not, it cannot tackle the selfish behavior of non-forwarding messages.

6.4 Experimental Analysis of Existing Strategies for Preventing Selfish Behavior

In this section, we first introduce representative strategies in the categories discussed above. We then present the experiment settings. The routing algorithm and performance metrics are subsequently depicted. Finally, we compare the performance of the different strategies for preventing selfish behavior.

6.4.1 Compared Strategies for Preventing Selfish Behavior

In the following experiments, we compare the performance of preventing selfish behavior of the following strategies against a basic routing protocol (i.e., Binary Spray and Wait), called *Non-strategy*, which does not cope with the selfish behavior of nodes. The detailed settings of the selected strategies are depicted in Table 6.1.

Barter: When two nodes encounter each other, they exchange the same amount of messages [15].

MobiCent: Due to the selected routing algorithm, which will be presented later, is multi-copy based, we choose the MobiCent as the representative strategy in the category of credit-based. In [21], the charging and rewarding processes are performed at the side of Virtual Bank (VB), when a message is firstly delivered to the destination. A constant credit is charged from the account of the source node in VB. The charged credit is equally divided, and distributed to the intermediate nodes in the message delivery path as a reward.

IRONMAN: Compared to barter-based and credit-based strategies, IRONMAN [7] includes the detection of selfish behavior. Therefore, it is selected as the representative strategy in reputation-based strategy. As depicted in Chapter 6.3.3, when two nodes encounter each other, they firstly check the forwarding records and the received messages, in order to detect the selfish nodes. The two encountering nodes then update the opinion about others' behavior with each other.

6.4.2 Simulation Settings

In order to evaluate the performance of different strategies for preventing selfish behavior, we utilize a widely utilized mobility model in MDTNs called Random WayPoint (RWP) [52] in the Opportunistic Network Environment (ONE) simulator [56] to conduct the experiment. In RWP, each node is initially specified a random destination within a given area, and it then moves towards the destination with a given speed. When it reaches the destination, it stays there for a certain period of time (i.e., a pause time). When the pause time expires, it randomly chooses a new destination, and repeats the above process. In order to avoid the impact of the setting of pause time on the routing performance, there is no pause time in the simulations as [60]. In addition, we specify a warm-up period (0.5 hour) as in [18] to uniformly distribute the initial position of nodes.

In this experiment, we consider 50 nodes in each simulation. To simulate the social relationships, we equally divide the nodes into two groups. Two nodes that belong to the same group are considered to have a social relationship; otherwise, the nodes are considered not to have a social relationship. During the simulation, a message with a random source and destination is generated every 5 seconds. Since the message generation process lasts for 12 hours, there are 8640 messages generated in each simulation. The detailed settings of the simulation are listed in Table 6.2

Table 6.1: Simulation Parameters for Strategies

Strategy Name	**Parameter Name**	**Value**
MobiCent	Initial Credit for Each Node	1
	Payment for Each Message	1
IRONMAN	Initial Trust for Each Node	0.5
	Trust Increment	0.5
	Trust Decrement	0.5
	Threshold	0.49

Table 6.2: Simulation parameters

Parameter Name	**Value**
Simulation Area	500 m x 500 m
Simulation Length	13.5 hours
Mobility Model	Random WayPoint (RWP)
Number of Mobile Nodes	50
Number of Groups	2
Number of Nodes in Each Group	25
Transmission Range	10 m
Node Speed	1 m/s
Warm-up Period	0.5 hour
Duration of Message Generation	12 hours
Message Generation Rate	1 message per 5 seconds
Time-To-Live (TTL)	1 hour

6.4.3 Routing Algorithms

Based on the above settings, we conducted our experiment with an efficient multi-copy routing algorithm in MDTNs, called Binary Spray and Wait [111]. The Binary Spray and Wait routing algorithm provides a platform for the selected strategies. The routing process is elaborated below.

Binary Spray and Wait: In [111], each message is associated with an attribute L, which indicates the maximum copies of the message that a message carrier can make. For each message, there are two phases: *spray* phase and *wait* phase. In the spray phase (i.e., $L > 1$), a message carrier hands over half of its message copies to an encountering node without the message. In the wait phase, the message can only be forwarded to the destination node. In the experiment, L is set to 5.

6.4.4 Performance Metrics

We use the following metrics to assess the impact of selfish behavior in DTNs:

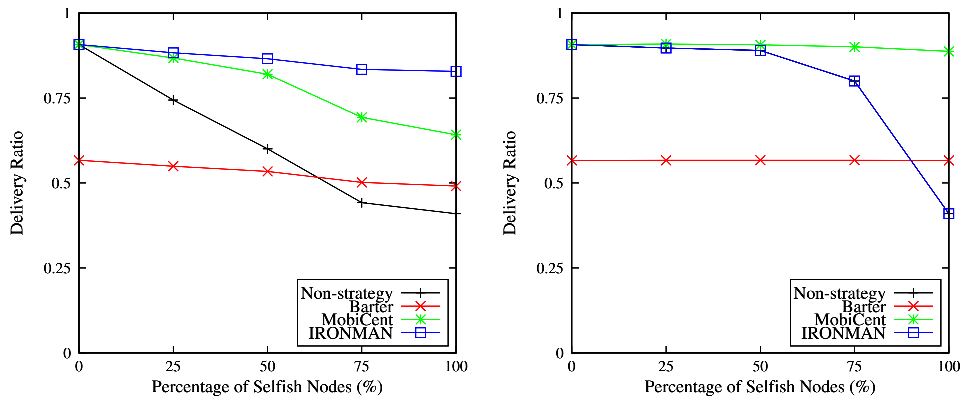
Delivery Ratio: The proportion of messages that have been delivered out of the total unique messages created.

Delivery Cost: The total number of messages (including duplicates) transmitted in the simulation. To normalize this, we divide it by the total number of unique messages created.

6.4.5 Simulation Results

In Figure 6.2(a), the performance of the strategies for preventing dropping messages is shown. When there is no selfish node, the performance of Non-strategy, MobiCent and IRONMAN is the same, since the cooperative nodes always cooperate with other nodes. However, the performance of Barter is lower than those of other strategies, due to the requirement of exchanging the same amount of messages. As the percentage of selfish nodes increases, the performance of all strategies is degraded. The performance of IRONMAN and MobiCent is always better than that of Non-strategy. The performance of Barter exceeds that of Non-strategy, when the percentage of selfish nodes is about 60%. The performance of IRONMAN is much better than those of other strategies even if all nodes are selfish, since it can detect the dropping of messages of a selfish node.

Figure 6.2(b) illustrates the performance of the strategies for preventing non-forwarding messages. The performance of all strategies decreases, as the percentage of selfish nodes increases. The performance of MobiCent is always better than other strategies. MobiCent can stimulate selfish nodes to be cooperative, as the number of the messages generated by selfish nodes increases. IRONMAN always



(a) Dropping

(b) Non-forwarding

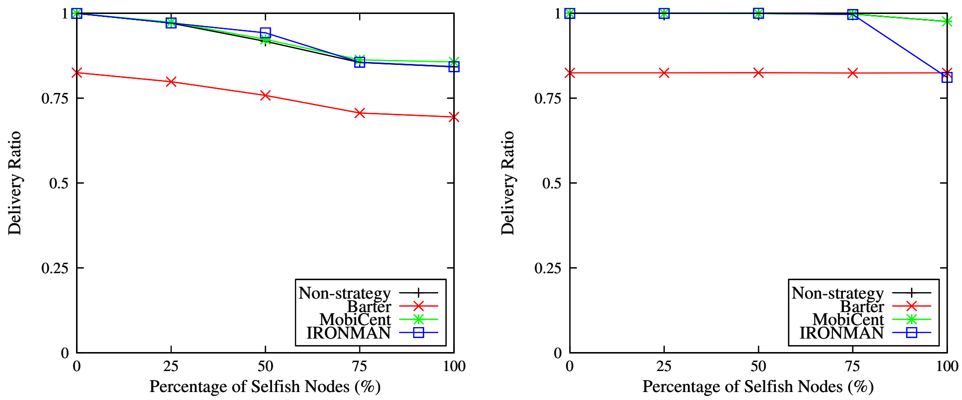
Figure 6.2: The routing performance in terms of delivery ratio under individual selfishness. The selfish actions of dropping and non-forwarding messages are illustrated in (a) and (b) respectively.

achieves the same performance as Non-strategy, since it cannot detect the selfish behavior of non-forwarding of messages. The performance of Barter only exceeds than that of Non-strategy, when the percentage of selfish nodes is about 82%.

In Figure 6.3, the delivery ratio of the four strategies under the social selfishness is investigated. From the figures, we can see that all the strategies cannot work well under social selfishness. Specially, the performance of Barter is even worse than that of Non-strategy, due to the requirement of exchanging the same amount of messages. For the selfish behavior of non-forwarding of messages, the performance of MobiCent is much better than those of other strategies, when the selfish nodes are 75% percentage.

From the above analysis of the simulation results, we can see that, for individual selfishness, the reputation-based strategies cannot prevent the selfish behavior of non-forwarding of messages. For social selfishness, there is no strategy that can efficiently prevent the selfish behavior of dropping of messages, and credit-based strategies can prevent the selfish behavior of non-forwarding of messages. The performance of barter-based strategies is always worse than that of credit-based and reputation-based strategies.

In this ongoing investigation, we do not study the delivery cost of these incentive stratigies, since the underling Spray-and-Wait routing protocol restricts the number of message copies to a constant.



(a) Dropping (b) Non-forwarding

Figure 6.3: The routing performance in terms of delivery ratio under social selfishness. The selfish actions of dropping and non-forwarding messages are illustrated in (a) and (b) respectively.

6.4.6 Comparison of Strategies

According to the above simulation results, we utilize three types of circles to indicate the performance of the selected strategies compared with that of Non-strategy: (1) ● indicates that the performance of a given strategy is always better than that of Non-strategy; (2) ◑ indicates that the performance of a given strategy is better than that of Non-strategy, only when the percentage of selfish nodes is high; and (3) ○ indicates that the performance of a given strategy always cannot exceed that of Non-strategy. The performance of the representative strategy in each category is listed in Table 6.3.

Table 6.3: Performance comparison of the selected strategies

Strategy	**Individual selfishness**		**Social selfishness**	
	Dropping	**Non-forwarding**	**Dropping**	**Non-forwarding**
Barter	◑	◑	○	○
MobiCent	●	●	○	◑
IRONMAN	●	○	○	○

6.5 Chapter Review

In this chapter, we first classified the selfish behavior in DTNs. The existing strategies for preventing selfish behavior are traditionally classified into three categories: Barter-based, credit-based and reputation-based. We subsequently analyzed the mechanisms explored in the existing strategies. Further, we pointed out the problems in each category. At last, we conducted an experiment to investigate the performance of the representative strategies in each category. The results of our experiment illustrate that (1) the performance of Barter-based strategies is impaired by the requirement of exchanging the same amount of messages; (2) reputation-based strategies cannot deal with non-forwarding selfishness in an environment in which the nodes have a high probability of being selfish to other nodes; (3) almost of the existing strategies cannot efficiently prevent social selfishness.

Chapter 7

Conclusions and Future Work

This chapter first concludes this thesis by summarizing the issues addressed and the contributions made in this thesis. We then discuss the future research directions.

7.1 Conclusions

The integration of wireless, short-range communication capabilities in personal mobile devices forms novel networks, in which nodes can directly communicate when they come within the radio range of each other without any pre-installed network infrastructure. Due to the mobility of nodes and the limited resources of mobile devices, the connectivity in these networks may be intermittently disrupted for a relative long period of time. Therefore, such novel networks are called mobile delay tolerant networks.

Routing is one of the fundamental challenges in mobile delay tolerant networks, since a complete routing path between two nodes that wish to communicate cannot be guaranteed. In the literature, a large body of research work has been done to deal with routing in these networks. However, these works have introduced some new open issues. In this thesis, we identified two routing issues and a privacy issue which are caused by existing protocols in mobile delay tolerant networks, and proposed a solution for each of them. Our research contributions in this thesis are summarized in the following sections.

7.1.1 An Adaptive Routing Protocol

In the literature, a large number of protocols have been proposed for routing in mobile delay tolerant networks. Epidemic can achieve the highest delivery ratio at the cost of a huge resource consumption. This is inconvenient as the resources in mobile devices are generally limited. In order to better choose intermediary nodes and thus reduce the routing overhead, the research community focused on using human mobility patterns to design efficient routing protocols. In order to better understand human mobility, various studies that have collected and analyzed real mobility traces have been conducted [35, 83, 10]. These studies have shown that: (1) human mobility is influenced by social relationships; (2) such social relationships are relatively stable over time. According to these observations, a number of social properties characterizing the mobility patterns of nodes have thus been defined.

Building on these social properties, routing protocols [47, 119, 78, 46, 48] have been proposed in the literature. However, these protocols assume that all messages can be correctly steered to their destinations by leveraging a single social property. This assumption is easily be violated due to the variety and dynamics of human social properties in real-life environment.

To resolve the above issue, we proposed an adaptive routing protocol that can dynamically adapt to the nodes' social properties at the very specific time and location. To this end, the proposed protocol dynamically learns the properties of the nodes based on their historical contacts. The forwarding decisions are then made by comparing utility values, which normalize and aggregate social properties. Thus, each time when an intermediate node is encountered, the most appropriate property is exploited to make forwarding decisions. The evaluation results showed that our protocol can achieve a better delivery ratio than the existing state-of-the-art routing protocols that rely on a single mobility property.

7.1.2 A Delay and Cost Balancing Routing Protocol

In the literature, a number of routing protocols have been proposed in MDTNs. Nevertheless, most of them are inefficient to achieve a good balance between the delivery delay and the delivery cost. As an example, flooding-based routing protocols [117], which refers to those protocols that rely on an unlimited number of message copies to route a message, cause a large number of redundant message copies, which obfuscates the (limited) resources of mobile devices. Quota-based routing protocols [111, 89, 113], allocate the same amount of message copies for messages with different Time-To-Lives (TTLs). The rigidity of the latter approaches makes them often inefficient, as a fixed number of message copies cannot suit all the routing situations. As a result, the dynamic allocation of message copies according to the urgency of messages is still an open issue in MDTNs.

To resolve the above issue, we presented a novel routing protocol in MDTNs, called Community-based Adaptive Spray (CAS) routing protocol. It contains a sub-protocol, which is utilized to identify the community structure of nodes in a network and the gateway nodes connecting two communities. In our protocol, to route a message towards a given destination node, the source of a message uses the collected community topology to precompute the multi-hop path traversing the minimal number of communities through their gateway nodes and that has the highest delivery probability. Furthermore, once the routing process is engaged, our routing protocol allocates the minimum number of message copies at each hop while achieving a given delivery probability according to the remaining TTL of the message. The evaluation results demonstrated that the protocol can achieve a good balance between message delivery delay and delivery cost, compared with most of the existing routing protocols in the literature.

7.1.3 A Privacy-Preserving Routing Protocol

The contacts between nodes in MDTNs have been revealed to be not random but follow patterns which are repetitive to a certain extent [78, 70]. Therefore, the encounter probability of nodes, which can be inferred from the contact history of nodes, is widely utilized in the design of efficient routing protocols. Such routing protocols are known as prediction-based routing protocols. These protocols were shown to perform better than other protocols when nodes exhibit well known mobility patterns [70]. However, prediction-based routing protocols implicitly assume that nodes accept to reveal their mobility patterns to other nodes. Unfortunately, such an assumption is not realistic, since this information can be used to infer private information about them [33].

To resolve the above issue, we presented a novel routing protocol in MDTNs, called Efficient Privacy Preserving Prediction-based Routing (E3PR) protocol. The proposed protocol can preserve the mobility pattern of the nodes from being disclosed by exploiting the mobility patterns of the communities rather than individual patterns. The evaluation results demonstrate that the protocol can obtain comparable routing performance to existing prediction-based protocols while preserving the mobility pattern of nodes.

7.2 Future Work

In order to realize the wide-scale deployment of MDTNs, more work needs to be accomplished. Our future research plan will focus on the following issues:

Firstly, the issue of social selfishness is still an open issue. However, a MDTN inherently depends on the cooperation of nodes to enable communication between two nodes that wish to communicate with each other but are out of the radio range of each other. However, nodes tend to behave selfishly due to the limited resources of mobile devices. Selfishness of nodes thus breaks the basic conditions of a MDTN, and in turn seriously influences the performance of existing routing protocols. According to the investigation in Chapter 6, we can see that there is no efficient incentive strategy to prevent the social selfishness. Thus, we plan to develop an incentive strategy to prevent social selfishness.

Secondly, several social properties, such as centrality and community, have been defined and widely utilized in the design of efficient routing protocols. The performance of these routing protocols has been shown to be improved by using these social properties. Most of existing protocols assume that these properties do not vary frequently. For instance, a node that belongs to a community does not frequently change its community. However, this needs to demonstrated. To the best of our knowledge, no research efforts have been done in this direction. Thus, we plan to investigate the evolution of these properties based on real mobility traces.

Bibliography

[1] Wi-Fi Alliance. http://www.wi-fi.org/.

[2] K. W. Axhausen. Social networks, mobility biographies, and travel: survey challenges. *Environment and Planning B: Planning and design*, 35(6):981–996, 2008.

[3] A. Balasubramanian, B. N. Levine, and A. Venkataramani. Replication routing in dtns: A resource allocation approach. *IEEE/ACM Transactions on Networking*, 18(2):596–609, april 2010.

[4] S. Ben Mokhtar. *Semantic Middleware for Service-Oriented Pervasive Computing*. PhD thesis, University Pierre et Marie Curie (Paris 6), 2007.

[5] C. Bettstetter. Mobility modeling in wireless networks: categorization, smooth movement, and border effects. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(3):55–66, 2001.

[6] C. Bettstetter, H. J. Vogel, and J. Eberspächer. Gsm phase 2+ general packet radio service gprs: Architecture, protocols, and air interface. *IEEE Communications Surveys & Tutorials*, 2(3):2–14, 1999.

[7] G. Bigwood and T. Henderson. Ironman: Using social networks to add incentives and reputation to opportunistic networks. In *Proceedings of IEEE International Confernece on Social Computing (SocialCom)*, pages 65–72, 2011.

[8] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[9] M. Boc, A. Fladenmuller, M. D. de Amorim, L. Galluccio, and S. Palazzo. Price: Hybrid geographic and contact-based forwarding in delay-tolerant networks. *Computer Networks*, 55(9):2352–2360, 2011.

[10] C. Boldrini and A. Passarella. Hcmm: Modelling spatial and temporal properties of human mobility driven by users' social relationships. *Computer Communications*, 33(9):1056–1074, 2010.

[11] E. Bulut and B. K. Szymanski. Exploiting friendship relations for efficient routing in mobile social networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(12):2254–2265, dec. 2012.

[12] E. Bulut, Z. Wang, and B. K. Szymanski. Cost-effective multiperiod spraying for routing in delay-tolerant networks. *IEEE/ACM Transactions on Networking*, 18:1530–1543, 2010.

[13] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, pages 1 –11. IEEE, 2006.

[14] 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*, 41(6):128–136, 2003.

[15] L. Buttyán, L. Dora, M. Félegyházi, and I. Vajda. Barter-based cooperation in delay-tolerant personal wireless networks. In *Proceedings of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6, 2007.

[16] L. Buttyán, L. Dora, M. Félegyházi, and I. Vajda. Barter trade improves message delivery in opportunistic networks. *Ad Hoc Networks*, 8(1):1–14, 2010.

[17] L. Buttyán and J. P. Hubaux. Enforcing service availability in mobile ad-hoc wans. In *Proceedings of ACM international symposium on Mobile ad hoc networking & computing (MobiHoc)*, pages 87–96. IEEE, 2000.

[18] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2(5):483–502, 2002.

[19] Y. Cao and Z. Sun. Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. *IEEE Communications surveys & tutorials*, pages 1–24, 2012.

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

[21] B. B. Chen and M. C. Chan. Mobicent: a credit-based incentive system for disruption tolerant network. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, pages 1–9. IEEE, 2010.

[22] M. Chuah and P. Yang. Impact of selective dropping attacks on network coding performance in dtns and a potential mitigation scheme. In *Proceedings of Internatonal Conference on Computer Communications and Networks (ICCCN)*, pages 1–6. IEEE, 2009.

[23] T. Clausen and P. Jacquet. Optimized link state routing protocol (olsr). 2003.

[24] E. M. Daly and M. Haahr. Social network analysis for information flow in disconnected delay-tolerant manets. *IEEE Transactions on Mobile Computing*, 8(5):606–621, 2009.

[25] H. Dang and H. Wu. Clustering and cluster-based routing protocol for delay-tolerant mobile networks. *IEEE Transactions on Wireless Communications*, 9(6):1874–1881, 2010.

[26] G. Dini and A. Lo Duca. A reputation-based approach to tolerate misbehaving carriers in delay tolerant networks. In *Proceedings of IEEE Symposium on Computers and Communications (ISCC)*, pages 772–777. IEEE, 2010.

[27] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.

[28] A. Elwhishi, P. Ho, K. Naik, and B. Shihada. Self adaptive contention aware routing protocol for intermittently connected mobile networks. *IEEE Transactions on Parallel and Distributed Systems*, PP(99):1–15, 2012.

[29] F. Fabbri and R. Verdone. A sociability-based routing scheme for delay-tolerant networks. *EURASIP Journal on Wireless Communications and Networking*, 2011(1):1–13, 2011.

[30] K. Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of ACM International Conference on Applications, technologies, architectures, and protocols for computer communications*, pages 27–34. ACM, 2003.

[31] Y. Feng, H. Gong, M. Fan, M. Liu, and X. Wang. A distance-aware replica adaptive data gathering protocol for delay tolerant mobile sensor networks. *Sensors*, 11(4):4104–4117, 2011.

[32] L. C. Freeman. A set of measures of centrality. *Sociometry*, 40(1):35–41, 1977.

[33] S. Gambs, M. O. Killijian, and M. N. del Prado Cortez. Show me how you move and i will tell you who you are. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, pages 34–41, 2010.

[34] O. Goldreich. *The Foundations of Cryptography - Volume 2*. Cambridge University Press, 2004.

[35] M. C. Gonzalez, C. A. Hidalgo, and A. L. Barabási. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.

[36] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of ACM International Conference on Computer and Communications Security (CCS)*, pages 89–98. ACM, 2006.

[37] R. Groenevelt, P. Nain, and G. Koole. The message delay in mobile ad hoc networks. *Performance Evaluation*, 62(1–4):210–228, 2005.

[38] M. Grossglauser and D. N. C. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Transactions on Networking*, 10(4):477–486, 2002.

[39] R. A. Guerin. Channel occupancy time distribution in a cellular radio system. *IEEE Transactions on Vehicular Technology*, 36(3):89–99, 1987.

[40] Z. J. Haas. A new routing protocol for the reconfigurable wireless networks. In *Proceedings of IEEE International Conference on Universal Personal Communications (ICUPC)*, volume 2, pages 562–566. IEEE, 1997.

[41] Z. J. Haas, J. Y. Halpern, and L. Li. Gossip-based ad hoc routing. *IEEE/ACM Transactions on Networking*, 14(3):479–491, 2006.

[42] O. Hasan, J. Miao, S. Ben Mokhtar, and L. Brunie. A Privacy Preserving Prediction-based Routing Protocol for Mobile Delay Tolerant Networks. In *Proceedings of IEEE International Conference on Advanced Information Networking and Applications (AINA)*, pages 1–8. IEEE, 2012.

[43] D. Hong and S. S. Rappaport. Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures. *IEEE Transactions on Vehicular Technology*, 35(3):77–92, 1986.

[44] X. Hong, K. Xu, and M. Gerla. Scalable routing protocols for mobile ad hoc networks. *IEEE Network*, 16(4):11–21, 2002.

[45] W. J. Hsu, T. Spyropoulos, K. Psounis, and A. Helmy. Modeling spatial and temporal dependencies of user mobility in wireless mobile networks. *IEEE/ACM Transactions on Networking*, 17(5):1564–1577, 2009.

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

[47] P. Hui and J. Crowcroft. Predictability of human mobility and its impact on forwarding. In *Proceedings of IEEE International Conference on Communications and Networking in China (ChinaCom)*, pages 543–547. IEEE, 2008.

[48] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: Social-based forwarding in delay-tolerant networks. *IEEE Transactions on Mobile Computing*, 10(11):1576–1589, 2011.

[49] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft. Distributed community detection in delay tolerant networks. In *Proceedings of ACM/IEEE International workshop on Mobility in the evolving internet architecture (MobiArch)*, pages 7:1–7:8. ACM, 2007.

[50] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. *SIGCOMM Computer Communication Review*, 34(4):145–158, August 2004.

[51] R. Jansen and R. Beverly. Toward anonymity in dtns: Threshold pivot scheme. In *Proceedings of Military Communications Conference (MILCOM)*, pages 587–592, 2010.

[52] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In Tomasz Imielinski and Henry F. Korth, editors, *Mobile Computing*, volume 353 of *The Kluwer International Series in Engineering and Computer Science*, pages 153–181. Springer US, 1996.

[53] T. Karagiannis, J. Y. Le Boudec, and M. Vojnović. Power law and exponential decay of intercontact times between mobile devices. *IEEE Transactions on Mobile Computing*, 9(10):1377–1390, 2010.

[54] M. Karaliopoulos. Assessing the vulnerability of dtn data relaying schemes to node selfishness. *IEEE Communications Letters*, 13(12):923–925, 2009.

[55] A. Kate, G. M. Zaverucha, and U. Hengartner. Anonymity and security in delay tolerant networks. In *Proceedings of IEEE International Conference on Security and Privacy in Communications Networks and the Workshops (SecureComm)*, pages 504–513. IEEE, 2007.

[56] Keränen, A., J. Ott, and T. Kärkkäinen. The one simulator for dtn protocol evaluation. In *Proceedings of International Conference on Simulation Tools and Techniques (SIMUTools)*, pages 55:1–55:10. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

[57] A. Keränen, T. Kärkkäinen, and J. Ott. Simulating mobility and dtns with the one. *Journal of Communications*, 5(2):92–105, 2010.

[58] A. Keranen, M. Pitkanen, M. Vuori, and J. Ott. Effect of non-cooperative nodes in mobile dtns. In *Proceedings of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–7. IEEE, 2011.

[59] M. Kim, D. Kotz, and S. Kim. Extracting a mobility model from real user traces. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, pages 1–13, 2006.

[60] Y. B. Ko and N. H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Wireless Networks*, 6(4):307–321, 2000.

[61] T. Kosch, C.J. Adler, S. Eichler, C. Schroth, and M. Strassberger. The scalability problem of vehicular ad hoc networks and how to solve it. *IEEE Wireless Communications*, 13(5):22–28, 2006.

[62] Z. Le, G. Vakde, and M. Wright. Peon: privacy-enhanced opportunistic networks with applications in assistive environments. In *Proceedings of ACM International Conference on PErvasive Technologies Related to Assistive Environments (PETRA)*, pages 76:1–76:8. ACM, 2009.

[63] J. Y. Le Boudec and M. Vojnovic. Perfect simulation and stationarity of a class of mobility models. In *Proceedings of IEEE International Conference on Computer and Communications (INFOCOM)*, volume 4, pages 2743–2754. IEEE, 2005.

[64] N. Li and S. K. Das. Radon: reputation-assisted data forwarding in opportunistic networks. In *Proceedings of ACM International Workshop on Mobile Opportunistic Networking (MobiOpp)*, pages 8–14. ACM, 2010.

[65] Q. Li, W. Gao, S Zhu, and G. Cao. A routing protocol for socially selfish delay tolerant networks. *Ad Hoc Networks*, 10(8):1619–1632, 2012.

[66] Y. Li, Y. Cao, S. Li, D. Jin, and L. Zeng. Integrating forwarding and replication in dtn routing: A social network perspective. In *Proceedings of IEEE Vehicular Technology Conference (VTC)*, pages 1–5. IEEE, 2011.

[67] Y. Li, P. Hui, D. Jin, L. Su, and L. Zeng. Evaluating the impact of social selfishness on the epidemic routing in delay tolerant networks. *IEEE Communications Letters*, 14(11):1026–1028, 2010.

[68] Y. Li, G. Su, and Z. Wang. Evaluating the effects of node cooperation on dtn routing. *AEU - International Journal of Electronics and Communications*, 66(1):62–67, 2012.

[69] Y. Li, G. Su, D.O. Wu, D. Jin, L. Su, and L. Zeng. The impact of node selfishness on multicasting in delay tolerant networks. *IEEE Transactions on Vehicular Technology*, 60(5):2224–2238, 2011.

[70] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 7(3):19–20, 2003.

[71] J. Liu and V. Issarny. Enhanced reputation mechanism for mobile ad hoc networks. In Christian Jensen, Stefan Poslad, and Theo Dimitrakos, editors, *Trust Management*, volume 2995 of *Lecture Notes in Computer Science*, pages 48–62. Springer Berlin / Heidelberg, 2004.

[72] M. Liu, Y. Yang, and Z. Qin. A survey of routing protocols and simulations in delay-tolerant networks. In Yu Cheng, Do Eun, Zhiguang Qin, Min Song, and Kai Xing, editors, *Wireless Algorithms, Systems, and Applications*, volume 6843 of *Lecture Notes in Computer Science*, pages 243–253. Springer Berlin / Heidelberg, 2011.

[73] R. Lu, X. Lin, H. Zhu, X. Shen, and B. Preiss. Pi: A practical incentive protocol for delay tolerant networks. *IEEE Transactions on Wireless Communications*, 9(4):1483–1493, 2010.

[74] R. Lu, Lin X., and Shen X. Spring: A social-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, pages 1–9. IEEE, 2010.

[75] X. Lu, P. Hui, D. Towsley, J. Pu, and Z. Xiong. Anti-localization anonymous routing for delay tolerant network. *Computer Networks*, 54(11):1899–1910, 2010.

[76] R. Malladi and D. P. Agrawal. Current and future applications of mobile and wireless networks. *Communications of the ACM*, 45(10):144–146, 2002.

[77] D. B. J. D. A. Maltz and J. Broch. Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks. *Computer Science Department Carnegie Mellon University Pittsburgh, PA*, pages 15213–3891, 2001.

[78] A. J. Mashhadi, S. Ben Mokhtar, and L. Capra. Habit: Leveraging human mobility and social network for efficient content dissemination in delay tolerant networks. In *Proceedings of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks & Workshops (WoWMoM)*, pages 1–6. IEEE, 2009.

[79] T. Matsuda and T. Takine. (p, q)-epidemic routing for sparsely populated mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 26(5):783–793, 2008.

[80] J. Miao, O. Hasan, S. Ben Mokhtar, and L. Brunie. A self-regulating protocol for efficient routing in mobile delay tolerant networks. In *Proceedings of IEEE International Conference on Digital Ecosystems Technologies (DEST)*, pages 1–6. IEEE, 2012.

[81] J. Miao, O. Hasan, S. Ben Mokhtar, L. Brunie, and K. Yim. An investigation on the unwillingness of nodes to participate in mobile delay tolerant network

routing. *International Journal of Information Management*, 33(2):252–262, 2013.

[82] M. Mouly, M.B. Pautet, and T. Foreword By-Haug. *The GSM system for mobile communications*. Telecom Publishing, 1992.

[83] M. Musolesi and C. Mascolo. A community based mobility model for ad hoc network research. In *Proceedings of ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc)*, pages 31–38. ACM, 2006.

[84] M. Musolesi and C. Mascolo. Designing mobility models based on social network theory. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(3):59–70, 2007.

[85] M. Musolesi and C. Mascolo. Car: context-aware adaptive routing for delay-tolerant mobile networks. *IEEE Transactions on Mobile Computing*, 8(2):246–260, 2009.

[86] M. Musolesi and C. Mascolo. Mobility models for systems evaluation. a survey. *Middleware for Network Eccentric and Mobile Applications (MINEMA)*, pages 43–62, 2009.

[87] J. Nash. Two-person cooperative games. *Econometrica: Journal of the Econometric Society*, pages 128–140, 1953.

[88] W. Navidi and T. Camp. Stationary distributions for the random waypoint mobility model. *IEEE Transactions on Mobile Computing*, 3(1):99–108, 2004.

[89] S. C. Nelson, M. Bakht, and R. Kravets. Encounter-based routing in dtns. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, pages 846–854. IEEE, 2009.

[90] J. Nieminen. On centrality in a graph. *Scandinavian Journal of Psychology*, 15:322–336, 1974.

[91] Institute of Electrical and Electronics Engineers. http://www.ieee.org/.

[92] M. R. Ogiela and U. Ogiela. Dna-like linguistic secret sharing for strategic information systems. *International Journal of Information Management*, 32(2):175 – 181, 2012.

[93] M. Onen, A. Shikfa, and R. Molva. Optimistic fair exchange for secure forwarding. In *Proceedings of International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous)*, pages 1–5, 2007.

[94] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[95] A. Panagakis, A. Vaios, and I. Stavrakakis. On the effects of cooperation in dtns. In *Proceedings of IEEE International Conference on Communication Systems Software and Middleware (COMSWARE)*, pages 1–6. IEEE, 2007.

[96] I. Parris and T. Henderson. Privacy-enhanced social-network routing. *Computer Communications*, 35(1):62–74, 2012.

[97] US Patent and Trademark Office. http://www.uspto.gov/.

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

[99] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, pages 90–100. IEEE, 1999.

[100] A. T. Prodhan, R. Das, H. Kabir, and G. C. Shoja. Ttl based routing in opportunistic networks. *Journal of Network and Computer Applications*, 34(5):1660–1670, 2011.

[101] J. M. Pujol, A. L. Toledo, and P. Rodriguez. Fair routing in delay tolerant networks. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, pages 837–845. IEEE, 2009.

[102] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, 16(4):482–494, 1998.

[103] E. M. Royer, P. M. Melliar-Smith, and L. E. Moser. An analysis of the optimum node density for ad hoc mobile networks. In *Proceedings of IEEE International Conference on Communications (ICC)*, volume 3, pages 857–861. IEEE, 2001.

[104] E. M. Royer and C. K. Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 6(2):46–55, 1999.

[105] G. Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, 1966.

[106] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. CRAWDAD trace cambridge/haggle/imote/content (v. 2006-09-15). Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle/imote/content, September 2006.

[107] A. Shamir. Identity-based cryptosystems and signature schemes. In George Blakley and David Chaum, editors, *Advances in Cryptology*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer Berlin / Heidelberg, 1985.

[108] U. Shevade, Han Hee Song, Lili Qiu, and Yin Zhang. Incentive-aware routing in dtns. In *Proceedings of IEEE International Conference on Network Protocols (ICNP)*, pages 238–247, 2008.

[109] C. Shi, X. Luo, P. Traynor, M. H. Ammar, and E. W. Zegura. Arden: Anonymous networking in delay tolerant networks. *Ad Hoc Networks*, 10(6):918–930, 2012.

[110] P. Smulders. The threat of information theft by reception of electromagnetic radiation from rs-232 cables. *Computers & Security*, 9(1):53–58, 1990.

[111] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of ACM SIGCOMM workshop on Delay-tolerant networking (WDTN)*, pages 252–259. ACM, 2005.

[112] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Performance analysis of mobility-assisted routing. In *Proceedings of ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*, pages 49–60. ACM, 2006.

[113] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. In *Proceedings of IEEE International Conference Pervasive Computing and Communications (PerCom) Workshops*, pages 79–85, 2007.

[114] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Efficient routing in intermittently connected mobile networks: the multiple-copy case. *IEEE/ACM Transactions on Networking*, 16:77–90, 2008.

[115] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Efficient routing in intermittently connected mobile networks: the single-copy case. *IEEE/ACM Transactions on Networking*, 16:63–76, 2008.

[116] T. Spyropoulos, T. Turletti, and K. Obraczka. Routing in delay-tolerant networks comprising heterogeneous node populations. *IEEE Transactions on Mobile Computing*, 8(8):1132–1147, 2009.

[117] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical report, Citeseer, 2000.

[118] M. Voss, A. Heinemann, and M. Muhlhauser. A privacy preserving reputation system for mobile information dissemination networks. In *Proceedings of International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm)*, pages 171–181, 2005.

[119] L. Vu, Q. Do, and K. Nahrstedt. 3r: fine-grained encounter-based routing in delay tolerant networks. In *Proceedings of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6. IEEE, 2011.

[120] L. Wei, H. Zhu, Z. Cao, and X. Shen. Mobiid: A user-centric and social-aware reputation based incentive scheme for delay/disruption tolerant networks. In Hannes Frey, Xu Li, and Stefan Ruehrup, editors, *Ad-hoc, Mobile, and Wireless Networks*, volume 6811 of *Lecture Notes in Computer Science*, pages 177–190. Springer Berlin / Heidelberg, 2011.

[121] X. Xie, H. Chen, and H. Wu. Bargain-based stimulation mechanism for selfish mobile nodes in participatory sensing network. In *Proceedings of IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 1–9. IEEE, 2009.

[122] Z. Yan, H. Zhou, and I. You. N-nemo: A comprehensive network mobility solution in proxy mobile ipv6 network. *Journal of Wireless Mobile Networks, Ubiquitous Computing and Dependable Applications*, Vol.1(2/3):52–70, 2010.

[123] Q. Yuan, I. Cardei, and J. Wu. An efficient prediction-based routing in disruption-tolerant networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(1):19–31, 2012.

[124] S. Zakhary and M. Radenkovic. Utilizing social links for location privacy in opportunistic delay-tolerant network. In *Proceedings of IEEE International Conference on Communications (ICC)*, pages 1059–1063, 2012.

[125] Z. Zhang. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges. *IEEE Communications Surveys and Tutorials*, 8(1):24–37, 2006.

[126] S. Zhong, J. Chen, and Y. R. Yang. Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *Proceedings of IEEE International Conference on Computer and Communications (INFOCOM)*, volume 3, pages 1987–1997. IEEE, 2003.

[127] H. Zhu, X. Lin, R. Lu, Y. Fan, and X. Shen. Smart: A secure multilayer credit-based incentive scheme for delay-tolerant networks. *IEEE Transactions on Vehicular Technology*, 58(8):4628–4639, 2009.

[128] Y. Zhu, B. Xu, X. Shi, and Y. Wang. A survey of social-based routing in delay tolerant networks: Positive and negative social effects. *IEEE Communications Surveys & Tutorials*, PP(99):1–15, 2012.