TÉCNICO LISBOA

UNIVERSIDADE DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

# Efficient and Secure Routing in Wireless Ad Hoc Networks

**Naercio David Pedro Magaia**

| | |
|---|---|
| **Supervisor:** | **Doctor Paulo Rogério Barreiros d'Almeida Pereira** |
| **Co-Supervisor:** | **Doctor Miguel Nuno Dias Alves Pupo Correia** |

**Thesis approved in public session to obtain the PhD Degree in**

**Electrical and Computer Engineering**

**Jury final classification: Pass with Distinction**

**2017**

UNIVERSIDADE DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

# Efficient and Secure Routing in Wireless Ad Hoc Networks

**Naercio David Pedro Magaia**

Supervisor: Doctor Paulo Rogério Barreiros d'Almeida Pereira
Co-Supervisor: Doctor Miguel Nuno Dias Alves Pupo Correia

Thesis approved in public session to obtain the PhD Degree in

**Electrical and Computer Engineering**

Jury final classification: **Pass with Distinction**

## Jury

Chairperson: Doctor Isabel Maria Martins Trancoso, Instituto Superior Técnico da Universidade de Lisboa

Members of the Committee:

Doctor Luís Filipe Lourenço Bernardo, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa
Doctor André Ventura da Cruz Marnoto Zúquete, Universidade de Aveiro
Doctor Paulo Rogério Barreiros d'Almeida Pereira, Instituto Superior Técnico da Universidade de Lisboa
Doctor António Manuel Raminhos Cordeiro Grilo, Instituto Superior Técnico da Universidade de Lisboa

Funding Institutions

**Fundação Calouste Gulbenkian**

2017

To my wife, parents and siblings.

# Acknowledgments

This thesis would not have been possible without the contribution, support and encouragement of many people.

In first place, I thank my supervisors, Professor Paulo Rogério Pereria and Professor Miguel Pupo Correia, for all they taught me through these years, for many great ideas that have made scientific research really exciting and for their constant support. This thesis would not exist without them.

In second place, I thank Professor Antonio Grilo and Professor Augusto Casaca in the Communication and Mobility Network group of INESC-ID for their encouragement and support.

In third place, I thank Professor Nuno Horta, Professor Rui Neves, Professor Alexandre Francisco and Professor Mario Figueiredo of the IST, for teaching me so much, for their attention to details, their encouragement and support.

My sincerest gratitude goes to Fundação Calouste Gulbenkian, specifically to Margarida Cunha and Dra. Margarida Abecassis for all their attention and support.

Many friends and colleages at INESC-ID were a constant support and encouragement through these years. I do not mention any because I would forget many more. Thanks!

Last but not least, I must thank all my family. This thesis is dedicated to my wife and my parents: it was intended for them since the beginning. My siblings were always a great support.

This work was partially supported by Fundação Calouste Gulbenkian and by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013.

# Resumo

Uma Rede Ad Hoc Sem Fios (Wireless Ad Hoc Network - WANET) pode ser definida como um conjunto de nós sem fios, de que são exemplos os computadores portáteis, telefones móveis, sensores, entre outros, que formam dinamicamente uma rede temporária sem a existência de qualquer infraestrutura de rede ou administração centralizada. Nas WANETs as decisões de encaminhamento são tomadas por cada nó individualmente, o que poderá levar ao consumo dos seus recursos, que por sua vez são limitados. Embora estas redes sejam auto-organizadas, necessitam para o seu correto funcionamento de que cada nó dê o seu próprio contributo. Assim sendo, surgem algumas preocupações: relativamente ao estabelecimento de confiança entre nós; ou em como estimular a sua cooperação; ou ainda devido à justiça das contribuições destes. Acrescem ainda questões de segurança como a confidencialidade, integridade, autenticação e privacidade.

Motivado por requisitos de Qualidade de Serviço (QoS) como atraso, variações de atraso, perda de pacotes, etc., e por desafios de segurança como a vulnerabilidade dos canais e nós, a ausência de infraestrutura, e pelo facto da topologia mudar dinamicamente, que são desafios críticos em WANETs, foram considerados dois estudos de caso. Em particular, um estudo de caso estático dada a natureza estática de muitas Redes de Sensores Multimedia Sem Fios (Wireless Multimedia Sensor Networks - WMSNs), e um estudo de caso dinâmico devido à natureza oportunista e dinâmica de muitas Redes Tolerantes a Atrasos (Delay-Tolerant Networks - DTNs).

É proposto um novo protocolo de encaminhamento eficiente de um ponto de vista energético para WMSNs, que utiliza uma abordagem cruzada, isto é, uma abordagem envolvendo as camadas de ligação e encaminhamento, para resolver o problema das falsas falhas de encaminhamento. São ainda utilizados pacotes de sonda para identificar as falsas falhas de encaminhamento, reduzindo assim a perda de pacotes e operações desnecessárias de manutenção de caminhos.

As aproximações clássicas por somente otimizarem um único parâmetro de QoS (como por exemplo, minimizar o atraso ou a taxa de perda de pacotes), não têm em conta a natureza conflituosa de vários parâmetros de QoS visto cada parâmetro corresponder a uma solução ótima diferente, o que leva a soluções não otimizadas de encaminhamento. Propõe-se na presente tese um novo protocolo de encaminhamento de múltiplos objectivos que considera vários requisitos de QoS das aplicações para WMSNs. Os resultados de simulação mostram que a abordagem proposta apresenta melhorias claras nas soluções de encaminhamento de QoS, mesmo quando comparada com a abordagem eficiente de um ponto de vista energético proposta acima.

A mobilidade dos nós ainda representa um desafio considerável para a maioria dos protocolos de encaminhamento das WMSNs por tornar os caminhos entre nós inúteis visto estes ficarem constantemente fora do alcance uns dos outros. Uma forma de tratar este problema é utilizando uma abordagem *store-carry-and-forward*, sendo esta mais comum em redes dinâmicas como as DTNs.

Contudo, os nós podem não dar o seu contributo na rede, uma vez que ao encaminhar mensagens de outros nós estes gastam os seus próprios recursos. Por forma a estudar este problema, é apresentada uma análise do impacto da existência de nós malcomportados nas DTNs. Os resultados das simulações realizadas mostram que, além da degradação do desempenho e da disponibilidade da rede, diferentes protocolos de encaminhamento são mais resistentes a diferentes tipos de nós malcomportados.

Os sistemas de reputação distribuídos podem ser usados para fomentar a cooperação entre nós em sistemas descentralizados e auto-organizados devido à inexistência de uma entidade central. Propõe-se um novo sistema de reputação para DTNs baseando-se numa abordagem bayesiana e que utiliza a distribuição Beta. Este sistema é robusto contra o envio de avaliações falsas do comportamento de nós, e eficiente na deteção de nós malcomportados, podendo ainda ser integrado em qualquer protocolo de encaminhamento para DTNs.

Nas DTNs, uma vez que os nós representam entidades e as ligações entre nós representam a relação entre duas entidades, algumas informações podem ser privadas, como por exemplo, as entidades proprietárias que gerem nós DTN, e suas relações. As aplicações DTN beneficiarão de mecanismos que reforcem a anonimidade da identidade das entidades e/ou das suas relações visto tratar-se de informação sensível ou confidencial. Propõe-se um novo protocolo de encaminhamento oportunista de preservação de privacidade para DTNs. Este garante a privacidade protegendo as informações confidenciais de cada nó, mesmo que ela tenha que ser enviada para ser processada noutro lugar. Nesta abordagem, os nós também comparam as suas métricas de encaminhamento duma forma privada por meio do esquema de criptografia homomórfica de Paillier.

**Palavras-chave:** Rede de Sensores Multimédia Sem Fios, Rede Tolerante a Atrasos, Encaminhamento eficiente, Reputação, Privacidade

# Abstract

A Wireless Ad hoc NETwork (WANET) can be seen as a collection of wireless nodes (e.g., laptops, smartphones, sensors, etc.) dynamically forming a temporary network without any existing network infrastructure or centralized administration. In WANETs, forwarding decisions are made individually by each node, which may lead to the consumption of nodes' limited resources. Despite being self-organized, these networks require for their correct operation that each node gives its own contribution. Therefore, some concerns arise in WANETs: in the establishment of trust between nodes; or to stimulate their cooperation; or even due to the fairness of their contributions. In addition, security issues like confidentiality, integrity, authentication and privacy also arise.

Motivated by Quality of Service (QoS) requirements such as delay, jitter, packet loss, etc., and security challenges, such as the vulnerability of channels and nodes, the absence of infrastructure, and the dynamically changing topology, that are critical challenges in WANETs, two case studies were considered, namely a static, due to the static nature of many Wireless Multimedia Sensor Networks (WMSNs), and a dynamic one, because of the opportunistic and dynamic nature of many Delay-Tolerant Networks (DTNs).

A novel energy efficient routing protocol for WMSNs, which uses a cross-layer approach to address the false routing failures problem involving the MAC and routing layers, is proposed. It also uses probe packets to identify routing failures, which reduces packet loss and unnecessary route maintenance operations.

Classical approximations optimize a single QoS parameter, not taking into account the conflicting nature of various QoS parameters which leads to sub-optimal routing solutions. A new multi-objective routing protocol that takes into account multiple QoS requirements of WMSNs applications is proposed. Simulation results show that the proposed approach presents clear improvements in the QoS routing solutions even when compared with existing energy efficient approaches.

The mobility of nodes still represents a considerable challenge to most WMSN routing protocols as nodes get more often unreachable rendering paths useless. One way to address this issue is to use a store-carry-and-forward approach that is more common in dynamic networks such as DTNs. However, nodes may not give their contribution to the network as forwarding other nodes' messages consumes their network resources. Therefore, an analysis of the impact of nodes' misbehavior in DTNs is presented. Besides degrading the performance and availability of the network, simulation results show that different routing protocols are more resilient to different types of nodes' misbehavior.

Distributed reputation systems can be used to foster nodes cooperation in decentralized and self-managed systems due to the nonexistence of a central entity. A new reputation system for DTNs, based on a Bayesian approach that uses the Beta distribution, is proposed. It is both robust against false ratings and efficient at detecting nodes' misbehavior, and can be integrated with any DTN routing protocol.

In DTNs, since nodes represent entities and links the relationship between two entities, some information may be private, such as the entities owning and managing DTN nodes and their relationships. A new privacy-preserving opportunistic routing protocol for DTNs that ensures privacy by protecting each node's sensitive information even if it has to be processed elsewhere, is proposed. In this approach, nodes also compare their routing metrics in a private manner using the Paillier homomorphic encryption scheme.

**Keywords:** Wireless Multimedia Sensor Network, Delay-Tolerant Network, Efficient routing, Reputa-

tion, Privacy

# Contents

Acknowledgments . . . . . . . . . . vii

Resumo . . . . . . . . . . ix

Abstract . . . . . . . . . . xi

List of Tables . . . . . . . . . . xviii

List of Figures . . . . . . . . . . xx

Glossary . . . . . . . . . . xxiv

**1 Introduction** **1**

1.1 Background and motivation . . . . . . . . . . 1

1.1.1 Wireless Multimedia Sensor Networks . . . . . . . . . . 2

1.1.2 Delay-Tolerant Networks . . . . . . . . . . 3

1.1.3 Research questions . . . . . . . . . . 4

1.2 Objectives . . . . . . . . . . 5

1.3 Contributions . . . . . . . . . . 5

1.4 Structure . . . . . . . . . . 6

**2 Background and related work** **9**

2.1 Notations . . . . . . . . . . 9

2.1.1 Static networks . . . . . . . . . . 9

2.1.2 Dynamic networks . . . . . . . . . . 10

2.2 Routing protocols . . . . . . . . . . 11

2.2.1 WMSNs . . . . . . . . . . 11

2.2.2 Multi-objective routing approaches . . . . . . . . . . 14

2.2.3 DTNs . . . . . . . . . . 15

2.3 The betweenness centrality metric . . . . . . . . . . 18

2.3.1 Concepts . . . . . . . . . . 18

2.3.2 Variants . . . . . . . . . . 21

2.3.3 Algorithms . . . . . . . . . . 24

2.3.4 DTN routing protocols . . . . . . . . . . 30

2.3.5 Discussion . . . . . . . . . . 42

2.4 Security mechanisms in WANETs . . . . . . . . . . 44

2.4.1 Security threats . . . . . 44
2.4.2 Security requirements . . . . . 45
2.4.3 Privacy in DTNs . . . . . 47
2.4.4 Homomorphic encryption . . . . . 48
2.4.5 Trust . . . . . 49
2.4.6 Incentive schemes . . . . . 50
2.4.7 Incentive schemes applied to DTNs . . . . . 55

**3 High throughput low coupling multipath routing for WMSNs 63**

3.1 Short retry limit in multipath routing protocols . . . . . 63
3.2 The ETX metric . . . . . 64
3.3 The correlation factor . . . . . 65
3.4 HTLC-MeDSR design . . . . . 66
3.4.1 Obtaining neighborhood information . . . . . 67
3.4.2 Obtaining ETX information . . . . . 68
3.4.3 Building route sets . . . . . 68
3.4.4 Data transmission . . . . . 69
3.4.5 Cross-layer approach . . . . . 69
3.4.6 Scalability . . . . . 69
3.5 Simulation model . . . . . 71
3.6 Simulation results . . . . . 72
3.6.1 Throughput . . . . . 72
3.6.2 Packet loss ratio . . . . . 74
3.6.3 Average end-to-end delay . . . . . 76
3.6.4 Control overhead . . . . . 78
3.6.5 Energy efficiency . . . . . 79
3.6.6 Path length . . . . . 81
3.7 Summary . . . . . 82

**4 A multi-objective routing algorithm for WMSNs 85**

4.1 Description of the problem . . . . . 85
4.2 Multi-objective optimization . . . . . 86
4.2.1 Basic definitions . . . . . 86
4.2.2 MOP formulation . . . . . 86
4.2.3 Strength pareto evolutionary algorithm . . . . . 88
4.2.4 SPEA implementation . . . . . 89
4.3 Simulation model . . . . . 90
4.3.1 Network simulator setup . . . . . 91
4.3.2 SPEA setup . . . . . 92
4.4 Simulation results . . . . . 92

4.4.1 Non-dominated sets . . . 92
4.4.2 Pareto-optimal sets . . . 93
4.4.3 QoS routing . . . 94
4.4.4 Discussion . . . 95
4.5 Summary . . . 96

**5 Nodes' misbehavior in DTNs 97**
5.1 Description of the problem . . . 97
5.2 Simulation model . . . 98
5.3 Simulation results . . . 99
5.3.1 Contact Characteristics . . . 99
5.3.2 Misbehavior of Types I and II for S1 . . . 100
5.3.3 Misbehavior of Type I and II for S1 and S2 . . . 103
5.3.4 Misbehavior of Type III . . . 104
5.4 Summary . . . 107

**6 A Robust and distributed reputation system for Delay-Tolerant Networks 110**
6.1 Assumptions and notations . . . 110
6.2 Bayesian decision theory . . . 111
6.3 The robust and distributed reputation system . . . 112
6.3.1 The Bayesian approach . . . 112
6.3.2 Information gathering . . . 113
6.3.3 Reputation rating . . . 114
6.3.4 Trust rating . . . 116
6.3.5 Bayesian classification . . . 116
6.4 Simulation model . . . 118
6.5 Simulation results . . . 119
6.5.1 The selection of the parameter $\chi$ . . . 120
6.5.2 Detection time of misbehaving nodes . . . 121
6.5.3 Robustness . . . 123
6.5.4 Reputation system overhead . . . 124
6.6 Summary . . . 125

**7 PRIVO: A PRIvacy-preserVing Opportunistic routing protocol for Delay-Tolerant Networks 127**
7.1 The PRIVO protocol . . . 127
7.1.1 Construction of the neighboring graph . . . 128
7.1.2 Anonymization of the neighboring graph . . . 129
7.1.3 Determination of routing metrics . . . 130
7.1.4 Routing algorithm . . . 131
7.2 Simulation model . . . 132

7.2.1 Synthetic mobility models . . . . . . . . . . 132
7.2.2 Real mobility traces . . . . . . . . . . 133
7.3 Simulation results . . . . . . . . . . 133
7.3.1 The selection of the parameters: number of timeslots and weight threshold . . . . . . . . 133
7.3.2 Routing performance . . . . . . . . . . 135
7.3.3 Cryptographic costs . . . . . . . . . . 135
7.3.4 Information loss . . . . . . . . . . 137
7.4 Summary . . . . . . . . . . 138

**8 Conclusions 141**
8.1 Achievements . . . . . . . . . . 141
8.2 Future work . . . . . . . . . . 142

**Bibliography 164**

# List of Tables

2.1 WMSN Routing Protocols . . . . . 14
2.2 Summary of multi-objective routing approaches . . . . . 15
2.3 DTN Routing Protocols . . . . . 17
2.4 A summary and comparison of the betweenness centrality metric . . . . . 22
2.5 A summary and comparison of the variants of betweenness centrality . . . . . 25
2.6 A summary and comparison of the algorithms used to compute betweenness centrality . . . . . 30
2.7 A summary and comparison of DTN Routing protocols that use betweenness centrality . . . . . 41
2.8 A summary list of potential attacks to WANETs . . . . . 46
2.9 A summary and comparison of incentive mechanisms applied to DTNs . . . . . 62

3.1 Simulation parameters . . . . . 71
3.2 MeDSR2 throughput gains for normal SRL (%) . . . . . 74
3.3 MeDSR2 throughput gains for SRL = 15 (%) . . . . . 74
3.4 MeDSR2 packet drop ratio gains for normal SRL (%) . . . . . 75
3.5 MeDSR2 packet drop ratio gains for SRL = 15 (%) . . . . . 76
3.6 MeDSR2 average end-to-end delay gains for normal SRL (%) . . . . . 77
3.7 MeDSR2 average end-to-end delay gains for SRL = 15 (%) . . . . . 78
3.8 MeDSR2 control overhead gains for normal SRL (%) . . . . . 78
3.9 MeDSR2 control overhead gains for SRL = 15 (%) . . . . . 79
3.10 MeDSR2 energy efficiency gains for normal SRL (%) . . . . . 80
3.11 MeDSR2 energy efficiency gains for SRL = 15 (%) . . . . . 81
3.12 MeDSR2 path length Gains for normal SRL (%) . . . . . 81
3.13 MeDSR2 path length gains for SRL = 15 (%) . . . . . 82

4.1 Simulation parameters . . . . . 91
4.2 SPEA parameters . . . . . 92
4.3 Maximum transmission times . . . . . 93
4.4 Optimal solutions for the proposed SPEA . . . . . 93
4.5 Average objective function values with 95% confidence interval . . . . . 94
4.6 Proposed SPEA improvement ratio for both scenarios . . . . . 95

6.1 The most used notations in this chapter . . . . . 111

7.1 The most used notations in this chapter . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 128

7.2 Average Paillier execution times (ms) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 136

7.3 Average delivery ratio losses and gains using the Paillier cryptosystem (%) . . . . . . . . . . . . . 137

7.4 Average delivery ratio losses and gains using the Paillier cryptosystem with 1024Bits key (%) . . 137

# List of Figures

3.1 MeDSR throughput analysis in various static scenarios . . . . . . . . . . . . . . . . . . . . . . . 64
3.2 The ETX concept . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65
3.3 The correlation factor concept for a 12 nodes scenario . . . . . . . . . . . . . . . . . . . . . . . 66
3.4 The correlation factor concept for a randomly distributed scenario with 100 nodes . . . . . . . . . 67
3.5 Neighborhood information structure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 69
3.6 10x10 Grid network. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70
3.7 Throughput as function of source rate on S1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 73
3.8 Throughput as function of source rate on S2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 73
3.9 Packets loss ratio gains as function of source rate on S1 . . . . . . . . . . . . . . . . . . . . . . 75
3.10 Packets loss ratio gains as function of source rate on S2 . . . . . . . . . . . . . . . . . . . . . . 75
3.11 Average end-to-end delay as function of source rate on S1 . . . . . . . . . . . . . . . . . . . . . 76
3.12 Average end-to-end delay as function of source rate on S2 . . . . . . . . . . . . . . . . . . . . . 77
3.13 Control overhead as function of source rate on S1 . . . . . . . . . . . . . . . . . . . . . . . . . 78
3.14 Control overhead as function of source rate on S2 . . . . . . . . . . . . . . . . . . . . . . . . . 79
3.15 Energy Efficiency as function of source rate on S1 . . . . . . . . . . . . . . . . . . . . . . . . . 80
3.16 Energy Efficiency as function of source rate on S2 . . . . . . . . . . . . . . . . . . . . . . . . . 80
3.17 Path Length as function of source rate on S1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
3.18 Path Length as function of source rate on S2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 82

4.1 Solution representation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89
4.2 A candidate solution for the scenario of Figure 4.4(a) . . . . . . . . . . . . . . . . . . . . . . . 89
4.3 During the crossover operation, *offspring*$_1$ and *offspring*$_2$ are created by concatenating parts from *parent*$_1$ and $parent_2$. For example, *offspring*$_1$ results from the concatenation of nodes starting from *SN* to common node B from *parent*$_1$ and with the nodes following B to *DN* in *parent*$_2$ . . . . 90
4.4 The simulation scenarios . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 91
4.5 Non-dominated Sets . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 92
4.6 Pareto Optimal Sets . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93

5.1 The simulation scenarios . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 98
5.2 Contact characteristics as function of the scenarios . . . . . . . . . . . . . . . . . . . . . . . . 99

5.3 Average delivery probability as function of the percentage of Types I and II misbehaving nodes for S1 . . . 100
5.4 Average latency as function of the percentage of Types I and II misbehaving nodes for S1 . . . . . 101
5.5 Average overhead ratio as function of the percentage of Types I and II misbehaving nodes for S1 . 102
5.6 Average buffer time as function of the percentage of Types I and II misbehaving nodes for S1 . . . 102
5.7 Average hop count as function of the percentage of Types I and II misbehaving nodes for S1 . . . 103
5.8 Average delivery probability as function of the percentage of Types I and II misbehaving nodes in both scenarios . . . 104
5.9 Average delivery probability as function of the percentage of Type III misbehaving nodes in both scenarios . . . 105
5.10 Average latency as function of the percentage of Type III misbehaving nodes in both scenarios . . 106
5.11 Average overhead as function of the percentage of Type III misbehaving nodes in both scenarios . 107
5.12 Average buffer time as function of the percentage of Type III misbehaving nodes in both scenarios 108

6.1 The variation of NBFPR and NBFNR as function of $\chi$ for the Epidemic, MaxProp and Prophet routing protocols . . . 120
6.2 The time necessary to correctly classify misbehaving nodes as DO_NOT_FORWARD for Epidemic, MaxProp and Prophet routing protocols and for 20, 40, 60 and 80% of liars . . . . . . . . 121
6.3 The time necessary to correctly classify misbehaving nodes as DO_NOT_FORWARD for Epidemic, MaxProp and Prophet routing protocols and for 20, 40, 60 and 80% of black-hole nodes . . 122
6.4 Reputation, Trust, Node's behavior and Trustworthiness false positives and negatives ratios for Epidemic, MaxProp and Prophet routing protocols and for 20, 40, 60 and 80% of liars . . . . . . . 124
6.5 Reputation, Trust, Node's behavior and Trustworthiness false positives and negatives ratios for Epidemic, MaxProp and Prophet routing protocols and for 20, 40, 60 and 80% of black-hole nodes 125
6.6 The reputation system overhead ratio (RSOR) for Epidemic, MaxProp and Prophet routing protocols and for 20, 40, 60 and 80% of misbehaving nodes . . . 126

7.1 Delivery ratio of PrivoSDBC for WDM and INFO5 scenarios with $\eta$ varying from 24 to 288 and $\varepsilon$ varying from $1 \times 10^{-1}$ to $1 \times 10^{-10}$ . . . 134
7.2 Delivery ratio for PrivoASP and PrivoMTTE in SPMBM and TR scenarios with $\eta$ varying from 24 to 288 . . . 135
7.3 Delivery and overhead ratios for all the routing protocols considered in different scenarios for different message generation rates . . . 136
7.4 Average correlation coefficient (CC) and delivery ratio (DR) for PrivoSDBC in SPMBM and TR scenarios for different percentages of total (A) and binary anonymizations . . . 138

# Glossary

| | |
|---|---|
| **ACK** | Acknowledgment |
| **ACO** | Ant Colony Optimization |
| **AIS** | Automatic Identification System |
| **AODV** | Ad hoc On-demand Distance Vector |
| **AOMDV** | Ad hoc On-demand Multipath Distance Vector |
| **BFS** | Breadth First Search |
| **BS** | Base Station |
| **C-Window** | Cumulative window |
| **CAR** | Context Aware Routing |
| **CBR** | Constant Bit Rate |
| **CDF** | Cumulative Distribution Function |
| **CF** | Correlation Factor |
| **DARPA** | U.S. Defense Advanced Research Project Agency |
| **DCF** | Distributed Coordination Function |
| **DDF** | Delegation Destination Frequency |
| **DDLC** | Delegation Destination Last Contact |
| **DD** | Direct Delivery |
| **DNI** | Dynamic Network Information |
| **DSR** | Dynamic Source Routing |
| **DTN** | Delay-Tolerant Network |
| **EA** | Evolutionary Algorithms |
| **ETX** | Expected Transmission Count |
| **FC** | First Contact |
| **FRF** | False Routing Failures |
| **G2G** | Give2Get |
| **GrAnt** | Greedy Ant |
| **HTLC-MeDSR** | High Throughput Low Coupling Multipath extension to the Dynamic Source Routing |
| **IAMDV** | Interference-Aware Multipath Routing for Video Delivery |

| | |
|---|---|
| **IBE** | Identity Based Encryption |
| **IEEE** | Institute of Electrical and Electronics Engineers |
| **LQ** | Link Quality |
| **MAC** | Medium Access Control |
| **MANET** | Mobile Ad Hoc Network |
| **MDART** | Multipath Dynamic Address Routing protocol |
| **MIT** | Massachusetts Institute of Technology |
| **MODE** | Multi-Objective Differential Evolution |
| **MOEA** | Multi-Objective Evolutionary Algorithm |
| **MOP** | Multi-objective Optimization Problems |
| **MO** | Multi-objective Optimization |
| **MULE** | Mobile Ubiquitous LAN Extension |
| **MeDSR** | Multipath Extension to Dynamic Source Routing protocol |
| **NE** | Nash Equilibrium |
| **NLQ** | Neighbor Link Quality |
| **NSGA** | Non-dominated Sorting based Genetic Algorithm |
| **NS** | Network Simulator |
| **OSM** | Offline Security Manager |
| **OSPF** | Open Shortest Path First |
| **P2P** | Peer-to-Peer |
| **PKG** | Private Key Generator |
| **POM** | Proof Of Misbehavior |
| **POR** | Proof Of Relay |
| **POS** | Pareto-optimal set |
| **PP** | Public Parameters |
| **PQBCF** | P2P Query algorithm based on Betweenness Centrality Forwarding |
| **PRIVO** | PRIvacy-preserVing Opportunistic routing protocol |
| **PRNet** | Packet Radio Network |
| **PRoPHET** | Probabilistic Routing Protocol using History of Encounters and Transitivity |
| **PSN** | Pocket Switched Networks |
| **QoS** | Quality of Service |
| **RANK** | greedy RANKing algorithm |
| **RAPID** | Resource Allocation Protocol for Intentional DTN |
| **RCAR** | Reputation-based extension to the CAR |
| **RERR** | Route Error |

| | |
|---|---|
| **RMPSR** | Robust Multipath Source Routing |
| **RREP** | Route REPly |
| **RREQ** | Route REquest |
| **RRS** | Robust Reputation System |
| **RTS/CTS** | Request to Send / Clear to Send |
| **RVGA** | Real-Valued Genetic Algorithm |
| **S-Window** | Single window |
| **SMART** | Secure Multilayer credit-based incentive |
| **SNI** | Static Network Information |
| **SON** | Social Opportunistic Network |
| **SPEA** | Strength Pareto Evolutionary Algorithm |
| **SRL** | Short Retry Limit |
| **SSSP** | Single-Source Shortest-Paths |
| **SdCF** | Second degree Correlation Factor |
| **SnW** | Spray and Wait |
| **TFT** | Tit-for-Tat |
| **TTL** | Time-To-Live |
| **TTP** | Trusted Third Party |
| **VB** | Virtual Bank |
| **VC** | Vapnik-Chernovenkis |
| **VD** | Vertex Diameter |
| **WANET** | Wireless Ad hoc NETwork |
| **WLAN** | Wireless Local Area Network |
| **WMN** | Wireless Mesh Networks |
| **WMSN** | Wireless Multimedia Sensor Networks |
| **WNA** | Weighted Network Analysis |
| **WSLS** | Win-Stay-Lose-Shift |
| **WSN** | Wireless Sensor Network |

# Chapter 1

# Introduction

## 1.1 Background and motivation

The notion of *ad hoc wireless networking*, introduced by the U.S. Defense Advanced Research Project Agency (DARPA) Packet Radio (PRNet) Project [FL01] in the 70's, enables wireless networking in environments where either there is no wired or cellular infrastructure, or the existing infrastructure is not adequate or cost effective. According to the Oxford Dictionary of English [Ste10], the term "ad hoc" means "created or done for a particular purpose as necessary". Therefore, a Wireless Ad hoc NETwork (WANET) can be seen as a collection of wireless nodes (e.g., laptops, smartphones, sensors, etc.) dynamically forming a temporary network without any existing network infrastructure or centralized administration.

WANETs are characterized by: (1) mobility, since nodes can rapidly be repositioned and/or move; (2) multi-hopping, as the path from source to the destination node may traverse several intermediate nodes; (3) self-organization, since the network must autonomously determine its own configuration parameters (e.g., addressing, routing, etc.); (4) energy conservation, as most nodes may have limited power supplies or cannot generate their own power; (5) scalability, since the network can grow to thousands or even millions of nodes; (6) security, due to the open wireless medium; and optionally, (7) connectivity to the Internet, by extending infrastructure-based wireless networks opportunistically with ad hoc nodes [Moh05].

WANET applications can set up communications in a specialized, customized and improvised manner in areas, where there is no infrastructure (e.g., remote locations) or it has failed due to, for example, natural disasters, or even if it is not adequate or cost effective, as it is the case of Wireless Sensor Networks (WSNs) [ASSC02] that can be considered a static application of WANETs [Moh05]. WANETs are also used to *opportunistically* extend wireless infrastructure networks to areas not easily reached by the latter by means of Delay/Disruption-Tolerant Networking [KAF12, FSB$^{+}$07].

In WANETs, forwarding decisions are made individually by each node, which may incur in the consumption of his limited resources, e.g., battery power, bandwidth, processing, and memory, all over the network. Despite being self-organized (or self-managed), these networks require for their correct operation that each node gives its own contribution[1]. Therefore, some concerns arise in WANETs: (1) in the establishment of trust between

[1] It is assumed that nodes have equivalent privileges and responsibilities

nodes, or (2) to stimulate their cooperation, or even (3) due to the fairness of their contributions. In addition, security issues like confidentiality, integrity, authentication and privacy also arise. To deal with confidentiality, integrity, authentication and privacy, nodes should use cryptographic mechanisms, since forwarding decisions are made based on the packet's content and/or context. Due to nodes' resource scarcity and to the fact of them being controlled by rational entities, nodes might misbehave. The misbehavior of nodes (malicious or selfish) can significantly impact network performance [HXL$^+$09, MPC13a, MPC13b].

Motivated by Quality of Service (QoS) requirements (such as delay, jitter, packet loss, etc.) and security challenges (due to the vulnerability of channels and nodes, the dynamically changing topology, the natural unreliable communication and no human supervision) that are critical challenges in WANETs, two case studies were considered, namely: a static and a dynamic one. The former was considered due to the static nature of many WSNs and the latter was considered because of the opportunistic and dynamic nature of many Delay-Tolerant Networks (DTNs) [KAF12]. Furthermore, efficient and secure routing in such environments is another challenge.

### 1.1.1 Wireless Multimedia Sensor Networks

WSNs are ad hoc networks in which nodes (sensors or actuators) collaborate to forward sensor data hop-by-hop from source nodes to sink nodes and vice versa. Sensor nodes consist of sensing, data processing, communication, storage and energy components. In order to minimize the cost, these sensor nodes are tiny devices with significant resource constrains.

Traditionally, WSNs have been used for monitoring applications based on low-rate data collection with short periods of operation. Advances in image sensor technology have enabled the use of audio and video sensors, which have the ability to retrieve, store, process, fuse and correlate multimedia data, originated from heterogeneous sources (multimedia sensor nodes, scalar sensor nodes, etc), fostering the appearance of Wireless Multimedia Sensor Networks (WMSNs) [AMC07]. WMSNs can be designed for real-time applications, which demand strict data delivery deadline, low delay, high throughput and reliability, and non-real time applications, which usually require medium to high bandwidth, low packet loss, etc. The transmissions in real-time and non-real time of multimedia and scalar data may have different QoS requirements, such as bounded latency or delay, throughput, jitter, energy efficiency, and reliability, depending on the application. Some examples of real-time mission critical and monitoring applications are search and rescue, security surveillance and patient monitoring [EH12]. To comply with such stringent QoS requirements of WMSNs, multiple paths in which each single path is a high throughput one may be used between a source and a destination node.

Multipath-based routing [AKK04] (hereinafter called *multipath routing*) allows building and using multiple paths for routing between a source and a destination node, by exploiting the resource redundancy and diversity in the underlying network to provide benefits as well as improvements in QoS metrics [TM06]. To complement, the Expected Transmission Count (ETX) [CABM05] metric can be used. ETX allows finding high throughput paths on a multi-hop wireless network, also incorporating the effects of link loss ratios, asymmetry in the loss ratios between the two directions of each link, and the interference among the successive links of a path. Despite the obvious benefits of the use of multipath routing, some care must be taken due to the route coupling issue [TM06], that is, due to the radio interference or contention among routes, which may have a serious impact on the routing performance even if the routes are node disjoint and have been carefully selected. In addition, by minimizing the hop-count, the distance traveled in each hop is maximized, which may contradict the performance requirements of WMSN applications, since longer distances reduce the signal strength, resulting in additional transmission errors. Furthermore, according to [NHJK05], packet loss due to collisions is often misinterpreted as link failure, also known as False Routing Failure (FRF). Consequently, the routing protocol attempts to find an alternative path even though the current path is still valid.

Many routing protocols [EH12, AY05] have been proposed to address WMSN QoS requirements. In most of these protocols, only one of the desired objectives (or QoS requirements) is optimized, while others are assumed as problems' constraints [MRV$^+$10]. In certain applications, a meta-heuristic approach [DF10, KLKH14] using a Multi-objective Optimization (MO) algorithm that can provide several optimal solutions may be preferred, since single design objective algorithms ignore other relevant objectives. By considering all objectives simultaneously, a set of optimal solutions can be generated, also known as the Pareto solutions [Deb08] of the multi-objective optimization problem. It is also known that finding optimal routes that satisfy multiple objectives in networks (multi-constrained QoS routing problem) is a NP-Complete problem [KP07]. Hence, efficient heuristic search algorithms based on reduced-complexity Evolutionary Algorithms (EAs) [KFV11] are necessary.

### 1.1.2 Delay-Tolerant Networks

For WMSNs, only static networks were considered. The mobility of nodes still presents a considerable challenge for most WMSN routing protocols, since nodes get more often unreachable, rendering paths useless. A way of addressing the nodes' mobility issue is to use a *store-carry-and-forward* approach more common in dynamic networks such as DTNs.

DTNs are ad hoc networks in which end-to-end paths might not exist at all the time (or even when necessary) between a source-destination pair. This contrasts with traditional networks (e.g., Mobile Ad Hoc Networks - MANETs) where a continuous end-to-end path is assumed to exist before messages are exchanged. However, end-to-end connectivity allowing messages to be forwarded between any pair of nodes may never exist in real MANETs due to node heterogeneity (different radios, resources), volatile links (as a consequence of node mobility, devices being turned off or running out of battery), or even energy efficient node operations (duty cycling). With the DTN store-carry-and-forward approach, mobility issues are no longer seen as obstacles, since nodes can carry messages with them while moving until an appropriate next node is found. In this approach, messages are relayed from one node into another until they reach their destination, or they are discarded.

Despite its attractiveness, DTN routing involves the challenging task of finding the most suitable node to forward messages to. A variety of network information is used to address this problem, namely: (1) dynamic network information (DNI), e.g., location information, traffic information and encounter information; (2) static network information (SNI), e.g., social relations among nodes. Through social network analysis, static network information (e.g., centrality [Fre78, BE06]) that is more stable over time can be leveraged and used by DTN routing protocols to facilitate the forwarding of messages. Centrality can be seen as a quantitative measure of the structural importance of a given node (or vertex) in relation to others within the network (or graph). Typically, a node can be considered as central if it plays an important role in the network's connectivity, e.g., if it is much required within a network for the transportation of information, or if it is more apt to connect to other nodes in the network. In DTNs, central nodes can be seen as good candidates to become relay nodes. Betweenness centrality, which was introduced independently in [Fre77a, Ant71], can be defined as the number of shortest paths passing through a given node. It takes into account the global structure of the network and can be applied to networks with disconnected components. It can be perceived as a measure of the load placed on a given node since it measures how well a node can facilitate communication among others. However, in most networks, information does not only flow along shortest-paths [FBW91, SZ89]. The determination of betweenness centrality has always been a challenging task, since in most cases it requires complete topology knowledge. So, it cannot be directly applied to DTNs.

In DTNs, nodes might misbehave because of their resource scarcity and of the fact of them being controlled by rational entities. Node misbehavior, malicious or selfish, can significantly impact network performance (see Chapter 5). DTN routing decision making becomes much simpler with the use of reputation and trust. Trust can be seen as the belief a node has in the peer's qualities, whereas reputation can be seen as a peer's perception about a node [MPC15]. To manage and organize decentralized and self-managed systems, incentive schemes (defined in Section 2.4.6) can be used, hence compensating for the nonexistence of a central or dedicated entity for, e.g., managing reputation and trust.

In a DTN, nodes represent individuals, vehicles, or other entities, and edges the relationship between two entities. Some information may be private, such as the entities owning and managing DTN nodes and their relationships. DTN applications would benefit from mechanisms that enforce the entities' identities and/or relationship anonymity due to the sensitive or confidential nature of the entities' identities and their behaviors. A DTN node may disclose private information by sending private data to other nodes. Privacy-preservation techniques allow protecting privacy through masking, modification and/or generalization of the original data without sacrificing the data utility [MBPC17].

### 1.1.3 Research questions

Lastly, for the two case studies envisioned in this thesis, i.e., the static case, consisting in WMSNs, and, the dynamic case, consisting in DTNs, the following list of research questions arise:

- How to design routing protocols that take into account multipath routing, ETX and the route coupling issue, i.e., how to efficiently route messages in WMSNs?
- How to design efficient routing protocols that take into account multiple WMSNs QoS requirements? Will these routing protocols be energy-efficient?
- How to collect reputation information in a DTN, as conversely to traditional networks, the source node can not count the arrival of ACKs associated with data packets that were sent nor monitor directly wireless channels to check if the next-hop node properly forwarded the data packet?
- How to evaluate each node's ability to provide correct information to the reputation system in DTNs?
- How to devise methods that allow nodes to exchange sensitive information meanwhile keeping privacy?
- How to design routing algorithms that allow nodes to compare their routing metrics without disclosing them, i.e., how to efficiently and privately forward messages in DTNs?[2]

[2] Since many previous work already addressed security aspects in WANETs [Pat10], it is not addressed in this thesis.

- How will the proposed protocol(s) perform compared with existing ones?

## 1.2 Objectives

The objectives of this PhD thesis are:

- To design and implement efficient routing protocols for WANETs. In particular, the proposed approaches should:

  1. be energy efficient meanwhile satisfying WMSNs QoS requirements such as throughput, delay, energy efficiency.
  2. use the available network information in DTNs to find the most suitable next node to forward messages to.

- To design and implement a robust and distributed reputation system and a secure routing protocol for DTNs.

## 1.3 Contributions

In summary, the contributions of this thesis are the following:

- A novel energy efficient routing protocol for WMSNs that uses a cross-layer approach to address the false routing failures problem involving the MAC and routing layers. It also uses probe packets to identify routing failures, which reduces packet loss and unnecessary route maintenance operations.

- A new multi-objective routing protocol that takes into account multiple QoS requirements of WMSNs applications.

- An analysis of the impact of nodes' misbehavior in DTNs, since the latter consumes network resources and degrades the network's performance and availability.

- A new reputation system for DTNs that is both robust against false ratings and efficient at detecting nodes' misbehavior. It can be integrated with any DTN routing protocol.

- A new privacy-preserving opportunistic routing protocol for DTNs that ensures privacy by protecting each node's sensitive information even if it has to be processed elsewhere.

Sustaining this thesis, the following articles were accepted and published:

**[MPC13a]** Naércio Magaia, Paulo Pereira, and Miguel P Correia. Nodes' misbehavior in vehicular delay-tolerant networks. In Proceedings of Conference on Future Internet Communications (CFIC), pages 1–9, May 2013.

**[MPC13b]** Naércio Magaia, Paulo Pereira, and Miguel P Correia. Selfish and malicious behavior in delay-tolerant networks. In Proceedings of Future Network and Mobile Summit (FutureNetworkSummit), pages 1–10. IEEE, 2013.

**[MPG15]** Naércio Magaia, Paulo Pereira, and António Grilo. High Throughput Low Coupling Multipath Routing for Wireless Multimedia Sensor Networks. Ad Hoc & Sensor Wireless Networks, 25(3-4):165–198, 2015.

**[MHN$^+$15]** Naércio Magaia, Nuno Horta, Rui Neves, Paulo Rogério Pereira, and Miguel Correia. A multiobjective routing algorithm for wireless multimedia sensor networks. Applied Soft Computing, 30:104 – 112, 2015.

**[MFPC15]** Naércio Magaia, Alexandre P. Francisco, Paulo Pereira, and Miguel Correia. Betweenness centrality in delay tolerant networks: A survey. Ad Hoc Networks, 33:284 – 305, 2015.

**[MPC15]** Naércio Magaia, Paulo Rogério Pereira, and Miguel P Correia. Cyber Physical Systems: From Theory to Practice, chapter Security in Delay-Tolerant Mobile Cyber-Physical Applications. CRC Press, 2015.

**[MBPC17]** Naércio Magaia, Carlos Borrego, Paulo Pereira and Miguel P Correia. PRIVO: A PRIvacy-preserVing Opportunistic routing protocol for Delay Tolerant Networks. In IFIP Networking, June 2017.

**[MPC17]** Naércio Magaia, Paulo Pereira, and Miguel P Correia. REPSYS: A Robust and Distributed Reputation System for Delay-Tolerant Networks. 20th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (accepted). November 2017.

**[RMP17]** Miguel Rodrigues, Naércio Magaia and Paulo Pereira. LBHD: Drop Policy for Routing Protocols with Delivery Probability Estimation. In 12th Iberian Conference on Information Systems and Technologies, June 2017.

And, the following articles were submitted:

**[MGPed]** Naércio Magaia, Pedro Dinis Gomes, and Paulo Rogério Pereira. FinalComm: Leveraging dynamic communities to improve forwarding in DTNs. Submitted to an international conference.

**[RMPed]** Miguel Rodrigues, Naércio Magaia and Paulo Pereira. Drop Policies for DTN Routing Protocols with Delivery Probability Estimation. Submitted to an international journal.

## 1.4 Structure

The rest of this PhD thesis is structured as follows.

In Chapter 2, a review of related literature is presented. Specifically, a review of routing protocols for WMSNs and DTNs that will be used on the following chapters of this thesis, surveys of betweenness centrality in DTNs and security mechanisms in WANETs are presented.

In Chapter 3, the design, implementation and evaluation of a novel energy efficient routing protocol for WMSNs, known as High Throughput Low Coupling Multipath extension to the Dynamic Source Routing (HTLC-MeDSR), is presented.

In Chapter 4, a new multi-objective routing protocol for WMSNs is presented. A comparison of the proposed approach with HTLC-MeDSR is also presented.

In Chapter 5, a study of nodes' misbehavior in DTNs is presented. Misbehaving nodes consume network resources, thus reducing the network's performance and availability. However, different protocols are more resilient to different types of node misbehavior, and that depends on different factors that are analyzed.

Chapter 6 presents a new robust and distributed reputation system for DTNs, which is based on a Bayesian approach that uses the Beta distribution. It uses all the available information and is both robust against false ratings and efficient at detecting nodes' misbehavior.

In Chapter 7, a new privacy-preserving opportunistic routing protocol, where a DTN is modeled as a time-varying neighboring graph where the edges correspond to the neighboring relationship among pairs of nodes, is presented. It ensures privacy by protecting each node's sensitive information even if it has to be processed elsewhere.

Finally, Chapter 8 presents concluding remarks and future work.

# Chapter 2

# Background and related work

A WANET (e.g., WMSN or DTN) can be seen as a collection of wireless nodes dynamically forming a temporary network without any existing network infrastructure or centralized administration. Since forwarding decisions are made individually by each node, these networks require for their correct operation that each node gives its own contribution.

In this chapter, a review of routing protocols for WMSNs and DTNs is presented. A review of multi-objective routing approaches is also presented. Moreover, two surveys are presented. On the one hand, betweenness centrality metric definitions and variants, its standard algorithms and a discussion on how DTN routing protocols make use of the metric and its algorithms to aid message forwarding are presented. On the other hand, a survey of security mechanisms for DTNs is presented.

## 2.1 Notations

This section introduces notation commonly used for static and dynamic networks. These notations will be used throughout this thesis.

### 2.1.1 Static networks

For static networks, a notation similar to [BE05, HHM08] is used. It is assumed that $G = (V, E)$ is a weighted, undirected graph. Each vertex (or node) $v \in V$ can be identified by an integer value $i = 1, 2, \ldots, |V|$. Each edge (or link) $e \in E \subseteq V \times V$ is identified by a pair $(i, j)$ representing a connection between vertex $i$ and vertex $j$ to which a weight $\omega(i, j) = \omega_{i,j}$ might be associated by $\omega : E \to \mathbb{R}^+$.

**Definition 2.1.1.** Walk. In a graph, a walk is a sequence of vertices $v_1, v_2, \ldots, v_k$, such that $(v_i, v_{i+1}) \in E$ for $i = 1, 2, \ldots, k-1$, where $v_1$ and $v_k$ are the walk's end vertices. The length of a walk is its number of edges. Two non-adjacent vertices are connected if there is at least one walk connecting them. Given a pair of distinct vertices $(s, t) \in V \times V, s \neq t$, a walk where all vertices and edges are distinct is considered a path $p_{st}$. The vertices $s$ and $t$ are called endpoints of $p_{st}$ and the vertices in Int$(p_{st}) = p_{st} \backslash \{s, t\}$ are the internal vertices of $p_{st}$. The shorthand $s \to t$ indicates that $s$ is connected to $t$, and its transitive closure $s \overset{+}{\to} t$ indicates that $s$ is connected to $t$ through a path of one or more edges in $G$.

**Definition 2.1.2.** Subgraph. $H = (V_H, E_H)$ is a subgraph of $G = (V, E)$, denoted $H \subset G$, if and only if (iff) $V_H \subset V$ and $E_H \subset E$.

**Definition 2.1.3.** Local subgraph. $H$ is a local subgraph with respect to a vertex $v \in V$, iff all vertices in the subgraph can be directly reached from $v$.

**Definition 2.1.4.** Geodesic path (or shortest path). A geodesic path between a pair of vertices $s$ and $t$, is one with the minimum length[1] $d_{st}$. A path length is the number of edges connecting vertices $s$ and $t$. If no paths exist between vertices $s$ and $t$, then $d_{st} = \infty$. Let $S_{st}$ denote the set of shortest paths between vertices $s$ and $t$, and $S_{d_{st}(v)}$ denote the set of shortest paths between vertices $s$ and $t$ that passes through vertex $v$. A vertex $v \in V$ lies on a shortest path between vertices $s, t \in V$, iff $d_{st} = d_{sv} + d_{vt}$.

**Definition 2.1.5.** Vertex-diameter (*VD*). Let $\mathbb{S}_G$ be the union of all the $S_{st}$'s, for all pairs $(s, t) \in V \times V$ of distinct nodes $s \neq t$. The vertex-diameter *VD*$(G)$ of $G$ is the size of the shortest path in $G$ with the maximum size, that is, it is the maximum number of vertices among all shortest paths in $G$, and is given by *VD*$(G) = \max\{|p| : p \in \mathbb{S}_G\}$.

**Definition 2.1.6.** Predecessors. The predecessors of a vertex $v$ on the shortest path from $s$ is $P_s(v) = \{u \in V : (u, v) \in E, d_{sv} = d_{su} + \omega(u, v)\}$.

### 2.1.2 Dynamic networks

For dynamic networks, a notation similar to [CFQS10, TMM+10] is used. Consider a set of entities (or vertices, nodes) $V$, a set of relations (or edges, links) $E$ among these entities, and an alphabet $L$ incorporating possible properties that such relation might have (e.g., terrestrial link, bandwidth of 8 MHz); specifically, $E \subseteq V \times V \times L$. It is assumed that entities' relations happen along a time span $\mathcal{T} \subseteq \mathbb{T}$ called the system's lifetime. The temporal domain $\mathbb{T}$ is commonly assumed to be: (1) $\mathbb{N}$ for discrete-time systems, (2) $\mathbb{R}^+$ for continuous-time systems. Thus, the dynamics of the system can be described by a temporal graph (or time-varying graph) $\mathcal{G} = (V, E, \mathcal{T}, \rho, \zeta)$, where

- $\rho : E \times \mathcal{T} \to \{0, 1\}$, named presence function, meaning that an edge is available at a particular time.
- $\zeta : E \times \mathcal{T} \to \mathbb{T}$, named latency function, meaning the amount of time necessary to cross a particular edge at a particular time (edge's latency can vary in time).

**Definition 2.1.7.** Journeys. A journey is composed by a sequence of pairs $\mathcal{J} = ((e_1, T_1), (e_2, T_2), \ldots, (e_k, T_k))$, so that $\{e_1, e_2, \ldots e_k\}$ that is a walk in $G$ is also a journey in $G$ iff $\rho(e_i, T_i) = 1$ and $T_{i+1} \geq T_i + \zeta(e_i, T_i), \forall i < k$. $departure(\mathcal{J})$ and $arrival(\mathcal{J})$ denote a journey's $\mathcal{J}$ starting date $T_1$ and last date $T_k + \zeta(e_k, T_k)$, respectively. Thus, journeys can be assumed as paths over time from a source to a destination having:

- Topological length, which is the number $|\mathcal{J}| = k$ of pairs, that is, the number of hops,
- Temporal length, which is the end-to-end duration: $arrival(J)$–$departure(J)$.

**Definition 2.1.8.** Distance. In temporal graphs, similarly to the length of a journey, the distance is also measured in terms of hops and time:

[1] Many geodesic paths may exist between two vertices.

- The topological distance $(d_{st,T})$ from a node $s$ to a node $t$ at time $T$ is given by $min\{|\mathcal{J}| : \mathcal{J} \in \mathcal{J}^{*}_{st},\ departure(\mathcal{J}) \geq T\}$. For a given time $T$, a shortest journey is one whose departure is $T^{'} \geq T$ and topological length = $d_{st,T}$. $\mathcal{J}^{*}_{st}$ denotes the set of all possible journeys starting at node $s$ and ending at node $t$.

- The temporal length $(\widehat{d}_{st,T})$ from a node $s$ to a node $t$ at time $T$ is given by $min\{arrival(\mathcal{J}) : J \in \mathcal{J}^{*}_{st},\ departure(\mathcal{J}) \geq T\} - T$.

  - For a given date $T$, a foremost journey is one whose departure $T^{'} \geq T$ and arrival is $T + \widehat{d}_{st,T}$.
  - For any given date $T$, a fastest journey is one whose departure is $T^{'} \geq T$ and temporal length is $min\{\widehat{d}_{st,T} : T^{'} \in \mathcal{T} \cap (T, +\infty)\}$.

**Definition 2.1.9.** Temporal graphs as a sequence of footprints. Given a temporal graph $\mathcal{G} = (V, E, \mathcal{T}, \rho, \zeta)$, a footprint of this graph from $T_1$ to $T_2$ is the static graph $G^{[T_1,T_2)} = \left(V, E^{[T_1,T_2)}\right) \forall e \in E, e \in E^{[T_1,T_2)}$ iff $\exists T \in [T_1, T_2), \rho(e, T) = 1$. Specifically, the footprint aggregates all interactions of a given time-window (or sub-interval) $\tau$ into static graphs. Considering that $\mathcal{T}$ is partitioned in consecutive sub-intervals $\tau = [T_0, T_1), [T_1, T_2), \ldots, [T_i, T_{i+1}), \ldots$; so that, $[T_k, T_{k+1}) \Leftrightarrow \tau_k$ denotes a sequence of footprints of $\mathcal{G}$ according to $\tau$ is $SF(\tau) = G^{\tau_0}, G^{\tau_1}, \ldots$.

## 2.2 Routing protocols

The routing protocols introduced in this section will be used in the evaluation sections of the following chapters.

### 2.2.1 WMSNs

Routing protocols in WSNs can be classified according to the network structure, protocol operation, how routing information is acquired and maintained [AKK04]. In terms of network structure, routing protocols can be divided into flat-based routing, hierarchical-based routing and location-based routing. In flat-based routing, nodes typically have similar roles, whereas in hierarchical-based routing nodes have different roles. In location-based routing, location information is used to route data in the network. According to the protocol operation, these protocols can be classified as multipath-based, query-based, negotiation-based, QoS-based, or coherent-based routing techniques. In multipath-based routing, multiple paths are maintained between a source-destination pair. In query-based routing, the destination node sends a query through the network and the node with the data that matches the query sends it. In negotiation-based routing, high level data descriptors are used to eliminate redundant data transmissions through negotiation. In QoS-based routing, certain QoS metrics have to be satisfied while routing data through the network. In coherent-based routing, sensors cooperate in processing data flooded throughout the network. According to how routing information is acquired and maintained, they can be classified into proactive, reactive, and hybrid. In proactive protocols, nodes compute routes before they are needed. In reactive protocols, nodes compute routes on demand. Hybrid protocols combine ideas of both.

Next, some routing protocols for WMSNs, which will be used in the evaluation sections of the following chapters, are surveyed.

The Dynamic Source Routing (DSR) protocol [JHM07] is an on-demand source routing protocol. In source routing, each data packet contains complete routing information to reach its destination. When a node wants to send a data packet, and no route to that destination is available on its route cache, it starts a route discovery process. Route discovery is the mechanism by which a source node discovers a route to a destination, typically by flooding Route REquest (RREQ) packets targeting the destination. When a neighbor of a source receives a RREQ packet, it first checks whether the packet is intended for it or not. If it is the destination, it sends a reply back to the source after copying the accumulated routing information contained in the RREQ packet into a Route REPly (RREP) packet. Route maintenance is the mechanism by which a node is able to detect any change in the network topology. The DSR protocol is a single path routing protocol.

The Robust Multipath Source Routing (RMPSR) protocol [WZ04] is a multipath extension to DSR. The basic idea behind RMPSR protocol is to discover multiple nearly disjoint routes between a source and a destination. To increase the probability of discovering multiple disjoint routes, the path selection criteria [WH01] include the following properties: disjoint nodes, small distance between the primary (shortest) and the other paths, and small correlation factor. The correlation factor of two node disjoint paths is defined as the number of links connecting the two paths. A route set consists of a primary route and several alternative routes. The primary route connects a source and a destination node, and alternative routes connect an intermediate node to a destination. The destination node collects multiple copies of RREQ packets of the same session within a time window, then builds multiple nearly disjoint route sets, and returns primary routes to the source node, and alternative routes to corresponding intermediate nodes.

The RMPSR protocol uses a per-packet allocation scheme to distribute video packets over two primary routes of two route sets. If one transmitting primary route is broken, the intermediate node that detects a broken link will send a Route Error (RERR) packet to the source node. Upon receiving the RERR packet, the source node removes the broken primary route from its route cache, and switches the transmission to another primary route.

The RMPSR protocol was designed taking into account the QoS requirements for video applications. It does not deal with issues like route coupling, as it selects nearly disjoint routes. In addition, the traffic allocation algorithm does not take into account the link quality while selecting the routes to send data, which means that routes with high packet loss may be selected, further degrading performance.

The Ad hoc On-demand Multipath Distance Vector (AOMDV) protocol [MD01] offers a multipath, loop-free extension to Ad hoc On-demand Distance Vector (AODV) [PBRD03]. It ensures that alternate paths at every node are disjoint. Therefore it achieves path disjointedness without using source routing.

To support multipath routing, route tables in AOMDV contain a list of paths for each destination. All the paths to a destination have the same destination sequence number. Once a route advertisement with a higher sequence number is received, all routes with the old sequence number are removed. Two additional fields, hop count and last hop, are stored in the route entry to help address the problems of loop freedom and path disjointedness, respectively.

Because AOMDV implements multipath discovery, the loop freedom guarantee from AODV no longer holds. AOMDV addresses this issue as follows. The hop count field contains the length of the longest path for a particular destination sequence number, and is only initialized once, at the time of the first advertisement for that sequence number. Hence, the hop count remains unchanged until a path for a higher destination sequence number is received. It follows that loop freedom is ensured as long as a node never advertises a route shorter than one already advertised, and never accepts a route longer than one already advertised.

To ensure that paths in the route table are link-disjoint, a node discards a path advertisement that has either a common next hop or a common last hop relative to one already in the route table. It was observed that, as long as each node adheres to this rule, all paths for the same destination sequence number are guaranteed to be link-disjoint. Node-disjoint paths can be obtained with the following additional restriction: for a particular destination sequence number, every node always advertises the same designated path to other nodes. Route maintenance in AOMDV and AODV are similar.

The AOMDV protocol attempts to find link disjoint or node disjoint routes between a source-destination pair. AOMDV does not address the route coupling issue even though it uses disjoint routes, because it does not guarantee that the routes are non-interfering. Furthermore, it does not take into account the link quality of the selected routes, similarly to RMPSR.

The Interference-Aware Multipath Routing for Video Delivery (IAMDV) protocol [WZ10] is an on-demand interference-aware routing protocol for a single source-destination pair, that tries to build two disjoint paths without the need of any special hardware support for locatization. IAMDV consists of two rounds of route request/replies. In the primary round, the protocol discovers the shortest path between the source-destination pair, and while the RREP packet travels back to the source node, the protocol tries to block neighbors of the nodes in the shortest path, that are not supposed to participate in the routing process. In the second round, the protocol attempts to find two paths that are distant from the shortest path by twice the radio communication range.

IAMDV prioritizes video traffic by splitting video streams into MPEG's I-, P- and B-frames, and sending the most important frames through the path with the smallest hop count, and the less important frames (P and B) through an alternate path. IAMDV takes into account the route coupling issue during the route discovery process, but it does not consider the link quality over the discovered/selected routes, as the protocol only considers the hop count metric for the selection of the path to forward the most important traffic.

In [MPG11], the authors proposed a Multipath Extension to Dynamic Source Routing protocol (MeDSR) . The basic idea is to build disjoint route sets for the source-destination pair. As in [WH01], to increase the probability of discovering multiple disjoint routes, when a node receives a RREQ packet, if it is the first time this RREQ packet is received or the path included in this message is node disjoint relative to the path included in a previously cached copy of the same RREQ packet, then the node will cache it and broadcast it again. In other cases, the node will discard this message. The route sets are built at the destination node, since the destination node knows the entire path of all available routes. The route sets consist of a primary route, connecting a source-destination pair, and the information about all the neighbors of the nodes in the route. The neighborhood information is collected by all nodes upon the route discovery process and is added to the RREP packet as it travels back to the source node.

Some procedures were implemented to manage RREQ and RREP packets, at the destination and source node respectively. The destination node collects RREQ packets and builds route sets, and the source node collects paths from received RREP packets and uses them during the multipath selection process. During the multipath selection process, the paths are grouped according to their correlation factor, and the ones with the smallest correlation are selected. The MeDSR protocol is a multipath routing protocol that takes into account the route coupling issue, but it does not consider link quality over the selected routes, nor does it attempt to distinguish collisions from routing failures.

| Routing Protocol | *-path | Route coupling | Multimedia QoS requirements | Link quality |
|---|---|---|---|---|
| DSR [JHM07] | single-path | N/A | No | No |
| RMPSR [WZ04] | multi-path | No | Yes | No |
| AOMDV [MD01] | multi-path | No | No | No |
| IAMDV [WZ10] | multi-path | Yes | Yes | No |
| MeDSR [MPG11] | multi-path | Yes | Yes | No |

Table 2.1: WMSN Routing Protocols

Table 2.1 summarizes the WMSN routing protocols and their characteristics.

### 2.2.2 Multi-objective routing approaches

The authors of [XSG06] proposed a multi-objective routing algorithm that identifies a set of Pareto Optimal routes, which represent different trade-offs between energy consumption and communication latency, for both single and multipath routing problems. One of the reasons behind the selection of the objectives is that sensor nodes are powered by batteries which makes power conservation an important goal. By minimizing the number of hops in a path, communication latency can be minimized since in most situations, latency is a consequence of the number of intermediate nodes along a communication path. WSN and WMSN have different QoS requirements, as multimedia traffic generally requires a minimum bandwidth.

In [YCH12], a performance comparison of two Multi-Objective Evolutionary Algorithms (MOEA), namely the Non-dominated Sorting based Genetic Algorithm-II (NSGA-II) [DPAM02] and the Multi-Objective Differential Evolution (MODE) [XSG03] algorithm, is presented. MOEAs are used to find optimal routes between a source and a destination nodes taking into account conflicting objectives, like dissipated energy and end-to-end delay in a fully-connected wireless network.

The authors of [COC11] proposed a QoS based Multi-Objective Optimization algorithm aiming at ensuring certain QoS levels in Wireless Mesh Networks (WMN). Some of the QoS parameters optimized are bandwidth, packet loss rates, delay and power consumption. The authors presented a linear programming formulation, not making clear how the presented formulation is used by the multi-objective optimization algorithm.

The authors of [REF14] proposed a multi-objective routing optimization approach that uses a real-valued genetic algorithm (RVGA), aiming at prolonging the average network lifetime. It obtains benefits of better convergence properties [FES03] by maintaining an unconstrained Pareto archive without employing an independent search population. The proposed approach, whose objectives are to minimize the total energy consumption and to maximize the time required for nodes to recharge or replace their batteries, accomplishes its goal by combining a $k$-shortest paths based search space pruning and an edge metric consisting of an association between a pair of nodes' energy cost with its link. Energy efficient routing protocols are important as they prolong nodes' battery lifetime. Routing protocols that use the ETX metric can find energy efficient paths, which allow the overall reduction of the network's energy consumption.

In [HEZ15], a multi-objective routing optimization, which uses an improved Strength Pareto Evolutionary Algorithm (SPEA2) [ZLT01], was proposed for WSNs aiming at minimizing both energy consumption and delay.

The authors of [MRD16] proposed a modified NSGA-II algorithm to address the QoS routing problem in WMNs. The objectives considered were the minimization of ETX and the transmission delay. The dynamic crowding distance was implemented in NSGA-II to preserve diversity in non-dominated solutions. Moreover, a decision-making procedure based on analytic hierarchy process was used to find the best optimal solution from the set of Pareto-solutions obtained by the algorithm.

| Publication | MOEA algorithm | Scenarios considered | Metrics considered |
|---|---|---|---|
| [XSG06] | MODE | Wireless Sensor Network | Energy consumption, delay |
| [YCH12] | MODE, NSGA-II | Full-connected network | Energy consumption, delay |
| [COC11] | NSGA-II | Wireless Mesh Network | Bandwidth, packet loss, energy consumption and delay |
| [REF14] | RVGA | Wireless Mesh Sensor Network | Energy consumption, battery lifetime |
| [HEZ15] | SPEA2 | Wireless Sensor Network | Energy consumption and delay |
| [MRD16] | MNSGA-II | Wireless Mesh Network | ETX and transmission delay |

Table 2.2: Summary of multi-objective routing approaches

Table 2.2 provides a summary of multi-objective routing approaches taking into account the algorithms used, scenarios considered and the metrics used as objective functions.

### 2.2.3 DTNs

Some taxonomies have been proposed to classify routing protocols in DTNs. According to [SRT$^+$10], routing protocols in DTNs are classified as deterministic (or scheduled), enforced or opportunistic. Deterministic routing happens if contact information is known a priori. Enforced routing is used to deliver messages to disconnected parts of a network (i.e., islands) by means of ferries [ZAZ04] or data mules [SRJB03b]. In opportunistic routing, no additional information (connectivity or mobility) is known a priori, nor special purpose nodes, such as data mules or ferries, are used. Opportunistic routing can be sub classified into three basic routing primitives, namely replication, forwarding and coding. In the message replication scheme, a relay node carrying a message may decide to spawn a new copy of the message and forward it to a newly encountered node. This scheme can be further sub classified in greedy, if a new copy of a message is spawn and forwarded to any node encountered that does not contains it; controlled, if there is a context (e.g., time-based, probability-based or copy-based) associated with each given message keeping track of the number of copies created; and utility-based, if a set of parameters related to the nodes in question (i.e., the current carrier and the candidate relay) are evaluated in order to assess the candidate node suitableness as a relay for a given message destined to a certain target node. In the message forwarding scheme, a relay node carrying a message may decide to pass that message over to another node it encounters, and by doing so it relinquishes its copy of the message and ceases to be one of its carriers. In the message coding scheme, a message may be coded and processed at the source (i.e., source coding) or as it traverses the network (i.e., network coding). Another approach can be used to analyze DTN routing, despite the common goal of finding a path to a destination taking into account the available information. The rationale behind it is to treat DTN routing as a resource allocation problem, thus having an intentional effect on DTN routing instead of an incidental one. The idea behind resource allocation is to forward or replicate a message to a relay based on the available resources in order to maximize the likelihood of message delivery, whenever two nodes meet. Resource allocation routing can use any of the three basic routing primitives. Taking into account the delivery semantics [CS13], routing protocols are divided in unicast, multicast and anycast. Unicast schemes deliver messages from a single source to its single destination. Multicast schemes deliver messages from a single source to a group of destinations. Anycast schemes deliver messages from a single source to any node within the ones composing a group.

Other taxonomies were also proposed taking into account the amount of social information used. According to [BLO$^+$14], routing protocols are divided in social-oblivious and social-aware schemes. This classification is based on the amount of social information employed while making routing decisions. In social-oblivious schemes, message replicas are randomly diffused, hoping that one will reach the destination. In social-aware schemes, probabilistic delivery estimation is made by nodes buffering messages to a certain destination. The idea is to relay messages to the most promising next hop based on its successful delivery probability. The authors of [WLX14] surveyed applications, taxonomy and design-related issues in social-aware routing protocols for DTNs. Social-aware routing protocols are classified in self-reported or detected. Self-reported routing protocols are those where routing decisions are made taking into account prior completely known social information. If nodes' social behavior is detected by means of an online method, and forwarding decisions are made based on that, these protocols are called detected routing protocols. The authors of [ZXSW13] surveyed and classified social-based routing protocols for DTNs according to the positive or negative effects of their social characteristics. Positive social characteristics are those that improve the DTN routing performance. Meanwhile, if nodes attempt to maximize their own utility or conserve their resources (that may be limited during operation) they tend to behave selfishly, thus presenting a negative social characteristic. Social-aware or social-based routing protocols (hereinafter called *social routing protocols*) can be further classified as single-property or multi-property (also called hybrid) depending on the number of social properties used. Moreover, in [WLX14, ZXSW13], a taxonomy to classify incentive mechanisms, i.e., mechanisms to stimulate cooperation among nodes in DTNs, was proposed. Incentive mechanisms are categorized as: reputation-based, remuneration-based (also known as credit-based) and game theory based (hereinafter called game-based). In reputation-based schemes, nodes use others' reputation records to make forwarding decisions. Good or bad reputation is gained by forwarding or not messages from other nodes, respectively. Nodes with bad reputation (misbehaving) are excluded from the network. Some form of credit (or reward) is used to control message forwarding in remuneration-based schemes. Nodes that forward others' messages are rewarded. The earned credit is used to obtain forwarding services from other nodes. In game-based schemes, forwarding decisions are modeled by game theory, where each node follows a strategy aiming at maximizing its benefits and minimizing its resource consumption, which, for instance, can be accomplished by maximizing its delivery probability or perhaps by minimizing its end-to-end delay. A well-known strategy is Tit-For-Tat, in which a node forwards as many messages for a neighbor as the neighbor has forwarded its.

Next, some routing protocols for DTNs, which will be used in the evaluation sections of the following chapters, are surveyed.

Direct Delivery [SPR04] and First Contact [KOK09] are single copy DTN routing protocols where only one copy of each message exists in the network. In Direct Delivery, the message is kept in the source and delivered only to the final destination, if the nodes meet. In First Contact, the message is transfered to the first node encountered and deleted from the previous carrier. The message is forwarded until it reaches the intended destination.

The Epidemic [VB00] routing protocol is an unlimited-copy routing protocol as nodes may replicate messages to any node they come in contact with. When two nodes come into communication range, they exchange a summary vector containing information about messages that they have seen. The receiving node decides whether it accepts the message or not. This decision may be taken, for example by not carrying messages for a certain destination node, or of a certain size. For example, buffer size, hop count and TTL fields may limit the amount of resources consumed through Epidemic Routing.

In many real environments, encounters between nodes are not random, but follow a predictable pattern. Mobile Ubiquitous LAN Extensions (MULEs) [SRJB03a] are mobile agents (vehicles or animals) that when in close range pick, buffer and drop off data, carrying data between remote locations. To exemplify a pattern, the data MULEs in [BDD$^+$05] meet with higher probability certain data MULEs. The Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) protocol [LDS03, LDS07] uses a probabilistic metric: delivery predictability, that attempts to estimate, based on node encounter history, which node has the higher probability of successful delivery of a message to the final destination. When two nodes are in communication range, a new message copy is transferred only if the other node has a better probability of delivering it to the destination.

MaxProp [BGJL06] attempts to transfer all messages not held by the other node, when it is in communication range. The protocol uses acknowledgments to clear the remaining copies of a message in the network when it is received by the destination node. When nodes discover each other, MaxProp exchanges messages in a specific priority order, taking into account message hop counts and the delivery likelihood to a destination based on previous encounters. New messages are assigned higher priority, and the protocol attempts to avoid reception of duplicate messages.

In the Resource Allocation Protocol for Intentional DTN (RAPID) [BLV07], routing packets are opportunistically replicated until a copy reaches the destination node. The protocol models DTN routing as a utility-driven resource allocation problem. The routing metric is a per-packet utility function. When nodes are in communication range, RAPID replicates the packet that results locally in the highest increase in utility. The corresponding utility $U_i$ of packet $i$, is defined as the expected contribution of $i$ to the given utility routing metric. RAPID is composed of three core components: (1) a selection algorithm determines which packets to replicate given their utilities when nodes are in communication range, (2) the inference algorithm, that given a routing metric estimates the utility of a packet, and (3) a control channel, that propagates metadata required by the inference algorithm.

Spray and Wait [SPR05] is an $n$-copy routing protocol with two phases: (1) spray phase, where a message created by the source node is initially spread by the source to encountered nodes until the $n$ copies are exhausted; (2) wait phase, where every node containing a copy of the message performs a direct delivery to the destination. There are two variants of the protocol: normal mode, where a node gives one copy of the message to each node it discovers that does not have the message; and binary mode, where half of the $n$ copies are given in each encounter.

| Routing Protocol | Abbreviations | *-copy | Estimation-based |
|---|---|---|---|
| Direct Delivery [SPR04] | DD | single-copy | No |
| First Contact [KOK09] | FC | single-copy | No |
| Epidemic [VB00] | Epidemic | unlimited-copy | No |
| PRoPHET [LDS03] | Prophet | unlimited-copy | Yes |
| MaxProp [BGJL06] | Maxprop | unlimited-copy | Yes |
| RAPID [BLV07] | Rapid | unlimited-copy | Yes |
| Spray and Wait [SPR05] | SnWNormal/SnWBinary | $n$-copy | No |

Table 2.3: DTN Routing Protocols

Table 2.3 summarizes the DTN routing protocols and their characteristics.

## 2.3 The betweenness centrality metric

This section introduces definitions and variants of the betweenness centrality metric. It also introduces its standard algorithms and a discussion on how DTN routing protocols make use of this social metric to aid message forwarding.

### 2.3.1 Concepts

In networks, the importance of a vertex or edge can be determined by the number of paths in which it participates. Centrality denotes the order of importance that vertices or edges have in a network by assigning real values to them. Since shortest paths are defined for both vertices and edges, centrality can be computed for a vertex $v$ or an edge $e$ (i.e., an element $x$) as presented below.

**Shortest-Path**

The *shortest-path betweenness centrality* [BE05] of an element $x$, which can be a vertex $v$ or an edge $e$, is based on the number of shortest paths that contain $x$. Let $\delta_{st}$ denote the fraction of shortest paths between the pair of vertices $s$ and $t$ containing vertex $v$, i.e.,

$$\delta_{st}\left(v\right) = \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{2.1}$$

where $\sigma_{st} = \left|S_{d_{st}}\right|$ and $\sigma_{st}(v) = \left|S_{d_{st}(v)}\right|$. The ratio $\delta_{st}(v)$, also called pair-dependency of $s, t$ on $v$, can be considered as the probability of any communication between vertices $s$ and $t$ involving vertex $v$. The shortest-path betweenness centrality $c_B(v)$ is defined as

$$c_B\left(v\right) = \sum_{s \neq v \in V} \sum_{t \neq v \in V} \delta_{st}\left(v\right) \tag{2.2}$$

From equation 2.2, one can conclude that the shortest-path betweenness centrality of a vertex measures the control over communications between others, since the shortest paths ending and starting in $v$ were excluded. In disconnected networks, any pairs of vertices $s$ and $t$ without any shortest paths between them must add zero to the shortest-path betweenness centrality of every other vertex in the network.

For an edge $e$, the pair-dependency of $s, t$ on $e$ is given by

$$\delta_{st}\left(e\right) = \frac{\sigma_{st}(e)}{\sigma_{st}} \tag{2.3}$$

and, the shortest-path betweenness centrality $c_B\left(e\right)$ of edge $e$ is given by

$$c_B\left(e\right) = \sum_{s \in V} \sum_{t \in V} \delta_{st}\left(e\right) \tag{2.4}$$

**Flow**

It was previously mentioned that shortest path based centrality metrics assume that the flow of information happens along the shortest paths. By considering small-world experiments [DMW03, TM67], one could assume that despite the shortest paths, a more realistic betweenness metric also included paths other than the shortest ones. In [FBW91], a more sophisticated betweenness metric, called *flow betweenness centrality*[2], was proposed also including contributions from non-shortest paths.

According to [New05], flow betweenness centrality of a vertex $v$ is defined as the amount of flow through $v$ when the maximum flow [CLRS09] is transmitted from $s$ to $t$, averaged over all $s$ and $t$. Since there might not be a unique solution to the flow problem, flow betweenness centrality can be more adequately defined as the maximum possible flow through a vertex $v$ over all possible solutions to the $st$ maximum flow problem[3], averaged over all $s$ and $t$ [FBW91]. It can be seen as a measure of betweenness of vertices in a network in which a maximum amount of information is uninterruptedly pumped between all sources and targets. Flow betweenness centrality cannot be computed directly by counting paths as the set of edge-independent paths among pairs of nodes are not unique.

Let $W$ denote the matrix of maximum flows among nodes, that is, the number of edge-independent paths among them, and $W[v]$ be the principal submatrix of $W$, that is, the matrix resulting from $W$ by removing column and row $v$. Additionally, let $W[v]^*$ be the matrix obtained by deleting node $v$ from the original network, and recalculating the flow matrix. The flow betweenness centrality is given by

$$c_{FB}\left(v\right) = \sum_{st} \frac{w[v]_{st} - w[v]^*_{st}}{w[v]_{st}} \tag{2.5}$$

Other betweenness metrics can be obtained from equation 2.5 by changing matrix $W$. Hence, to compute the shortest-path betweenness centrality, $W$ becomes the shortest path count matrix in which $w_{st}$ gives the number of shortest paths from $s$ to $t$. Note that the value returned by 2.5 is twice the one returned by 2.2.

### Current-Flow

In the *current-flow betweenness centrality* [BE05] metric, which is another alternative to the shortest-path betweenness centrality metric, the flow of information follows the behavior of an electrical current flowing through an electrical network. An electrical network is defined by a connected undirected graph $G = (V, E)$, together with a conductance function $c \ : E \rightarrow \mathbb{R}$. A supply function $b \ : E \rightarrow \mathbb{R}$, specifies an external electrical current entering and leaving the circuit.

Similar to shortest-path betweenness centrality, that counts the fraction of shortest $s$-$t$-paths through a vertex, current-flow betweenness of a vertex characterizes the portion of unit $s$-$t$-supplies through that vertex. The throughput of a vertex $v$, for a fixed $s$-$t$ pair, forms the current-flow equivalent of $\sigma_{st}(v)$ through $v$, i.e., with respect to a unit $s$-$t$-supply $b_{st}$, the throughput of vertex $v \in V$ is

$$\varsigma_{st}\left(v\right) = \frac{1}{2}\left( - \left|b_{st}\left(v\right)\right| + \sum_{e \ni v} \left|x(\overrightarrow{e})\right| \right) \tag{2.6}$$

where $x(\overrightarrow{e})$ is a (electrical) current function and $-\left|b_{st}\left(v\right)\right|$ sets to zero the throughput of a vertex with non-zero supply. This guarantees that a given unit $s$-$t$-supply is not considered for the throughput of its source node $s$ and sink node $t$, for the current-flow betweenness. Thus, the current-flow betweenness centrality $c_{CB} \ : V \rightarrow \mathbb{R}$ for an

[2] It is called flow betweenness centrality because of the association between the number of edge-independent paths among pairs of nodes and the quantity of material that could flow from one node to another through all possible edges [FF87].

[3] The maximum flow problem can be solved using standard algorithms [CLRS09].

electrical network $N = (G = (V, E)\, c)$ is

$$c_{CB}\left(v\right) = \frac{1}{\left(n-1\right)\left(n-2\right)} \sum_{s,t \in V} \varsigma_{st}\left(v\right),\ \forall v \in V \tag{2.7}$$

where $\frac{1}{(n-1)(n-2)}$ is a normalizing constant. Current-flow betweenness centrality measures the portion of throughput through vertex $v$ taken over all possible source-destination pairs.

### Random-Walk

Due to the lack of global knowledge, sometimes it may not be possible for a vertex to compute shortest paths. For these cases, an alternative way of traversing the network can be used by means of a random-walk model. The random walk model consists of walking from vertex to vertex, through the network's edges, i.e. an edge is randomly selected from a vertex $v$ to be followed, and the process is repeated from the new vertex.

It is assumed here that the graph is unweighted, connected and undirected. If, for example, a vertex $s$ wants to send a message to a vertex $t$ but neither $s$ nor its adjacent vertices knows how to reach $t$ through the shortest path, each vertex that gets the message for $t$, then selects at random one of its adjacent vertices to send the message.

It was demonstrated in [BE05] that the random-walk betweenness centrality $c_{RWB} \ : V \to \mathbb{R}$ is equivalent to the current-flow betweenness centrality, that is $c_{RWB}\left(v\right) = c_{CB}\left(v\right),\ \forall v \in V$. For a more detailed discussion, please refer to [BE05].

### Ego

An ego network, also known as the neighborhood network (or first order neighborhood) of the ego, can be defined as a network consisting of a single actor (ego) along with the actors it is connected to (alters) and all links among the latter. Ego networks allow an easier collection of data if compared to collecting data from the entire network, because the ego usually provides complete information of the alters (including how they are connected). By sampling such information, statistically significant conclusions about the entire population can be attained [EB05].

As previously stated, centrality measures allow finding the most important actors within a network and betweenness centrality studies the degree to which an actor is among all other actors within the network. If an actor is between two other actors, it follows that no alters on the path connecting the actors share a connection, or else it would form a shortest path. Hence, there is a connection between the betweenness centrality of the actor in the whole network and the one in the ego network (even though it may be difficult to quantify this association) [EB05]. Previous works have provided evidence of the usefulness of betweenness centrality in ego networks [Bur95].

To compute ego betweenness centrality (EBC), first it is necessary to compute the betweenness of a single actor. Due to the ego networks' structure, the shortest paths in the network are either of length 1 or 2. Every single pair of non-adjacent alters must have a shortest path of length 2 which passes through the ego. Note that shortest paths of length 1 do not contribute to the betweenness computation.

Let $\mathcal{A}_m$ be the adjacency matrix of $G$, then $\mathcal{A}^2_{m_{ij}}$ contains the number of walks of length 2 connecting vertex $i$ and vertex $j$. The shortest paths can be obtained by counting the number of paths of length 2 of non-adjacent pairs of actors. So,

$$\mathcal{A}_m^2 \left[\mathit{1} - \mathcal{A}_m\right]_{ij} \tag{2.8}$$

where $\mathit{1}$ is a matrix of all 1's, and 2.8 gives the number of shortest paths of length 2 between $i$ and $j$. The ego betweenness centrality is given by the sum of the halved reciprocal entries $\mathcal{A}_m^2 \left[\mathit{1} - \mathcal{A}_m\right]_{ij}$ such that $\mathcal{A}_{m_{ij}} = 0$.

Computing ego betweenness centrality of the entire network is one order of magnitude faster than computing, for example, the shortest-path betweenness centrality.

### Temporal

Similar to the shortest-path betweenness centrality metric used in static networks, the *temporal betweenness centrality* [TMM$^+$10] of a vertex $v$ could be defined as the fraction of shortest journeys that pass through $v$. However, besides the shortest journeys that pass through a vertex, it is also important to consider for how long a vertex along the shortest path holds a message before forwarding it, i.e., the fastest journeys among the shortest ones. Therefore, the temporal betweenness centrality of a vertex $v$ at time $T$ is:

$$c_{\mathcal{T}B,T}\left(v\right) = \frac{1}{(n-1)(n-2)} \sum_{s \neq v \in V} \sum_{t \neq v \in V} \frac{\psi_{st,T}\left(v\right)}{\psi_{st,T}} \tag{2.9}$$

where $\psi_{st,T}\left(v\right)$ returns the number of fastest journeys among the shortest ones from $s$ to $t$ passing through vertex $v$.

The temporal betweenness centrality for vertex $v$ over the entire temporal graph $\mathcal{G} = G^{[T_{\min}, T_{\max})}$ is:

$$c_{\mathcal{T}B}(v) = \frac{1}{|SF\left(\tau\right)|} \sum_{\tau=0}^{|SF(\tau)|} C_{\mathcal{T}B,T}\left(v\right)\left((T \times \tau) + T_{\min}\right) \tag{2.10}$$

where $|SF\left(\tau\right)|$ is the number of graphs in the sequence.

Table 2.4 presents a summary and comparison of betweenness centrality metrics based on the metrics, the type of network, the main idea, the type of network knowledge (global or partial), drawbacks and comments.

## 2.3.2 Variants

In this section, variants of betweenness centrality proposed in the literature are presented.

### Canonical-path betweenness

The authors of [GSS08] proposed a simple variant of betweenness centrality, called canonical path betweenness centrality (or simple canonical centrality), in which only a single canonical shortest path between any source-target pair is considered. The reasoning behind this variant are road networks, where multiple routes do exist in practice, but they usually share most edges. As a result, in general, $\delta_{st}\left(v\right)$ is one or zero. Also, unique shortest paths are enforced by perturbing the edge weights in some route planning methods [GSS08].

| Betweenness metric | Network Type | Main idea | Network knowledge | Drawbacks | Comments |
|---|---|---|---|---|---|
| Shortest path | Static | The flow of information happens along the shortest paths. | Global | The flow of information may not take the shortest-path (e.g. the small-world experiments). | It measures the control over communications between others. |
| Flow | Static, Dynamic | Although preferring shortest paths, the flow of information tries to exploit all possible paths. | Global | The flow of information may not be maximum and not follow optimal flow paths from source to target nodes. | It is based on the idea of maximum flow. It is a measure of betweenness of vertices in a network in which a maximal amount of information is continuously pumped between all sources and targets. |
| Current-flow | Static | The flow of information follows the behavior of an electrical current flowing through an electrical network. | Global | It can only be applied to electrical networks. | It is equivalent to random-walk betweenness. The current flows along all paths from source to target, but more on along the shortest ones (i.e., the ones in which the resistance is smaller). |
| Random-walk | Static, Dynamic | Uses the random-walk model to traverse the network. | Partial | It includes contributions from many paths that are not optimal in any sense. | It is suitable to a network in which information wanders around at random until it finds its target. |
| Ego | Static | It consists in summing the reciprocals of entries given by number of shortest paths of length 2 between a pair of non-adjacent vertices. | Partial | It is difficult to normalize the metric scores with respect to the ego network size. | There is no direct connection between the betweenness centrality computed for the entire network and the EBC. |
| Temporal | Dynamic | It is the fraction of fastest journeys among the shortest ones that pass through a given vertex. | Global | Similar to the shortest-path version. | It is based on the concept of shortest journeys. It measures the control over communications between others over time. |

Table 2.4: A summary and comparison of the betweenness centrality metric

### $\epsilon$-betweenness

In [CKSS02], the authors considered terrorist networks models [Gar01, Ste01, Res06], which may be large, dynamic and characterized by uncertainty[4]. Terrorist networks are considered: large, as the networks are unknown, i.e., the set of actors being monitored is likely a superset of those actually engaged in illicit activities, and dynamic, as the knowledge we have of them changes over time. Network's dynamics, i.e., mobility and nodes joining or leaving the network, may reflect inaccuracies in the shortest path calculations (besides the uncertainty in the shortest path length between a pair of nodes, the path itself may also change) causing perturbations in the betweenness centrality values. So, it may be necessary to recalculate betweenness centrality values as the network evolves through time. Some algorithms [DI01, DI04] have been proposed that support network's dynamics, thus avoiding, for example, the recalculation of shortest paths between all pairs of nodes.

A path $p_{st}$ is called an $\varepsilon$-shortest path if $|p_{st}| \leq (1+\varepsilon)d_{st}$. The $\varepsilon$-betweenness centrality is defined as

[4] In covert networks, there may be a deliberate effort to hide illicit activity, thus dynamicity and uncertainty also apply.

$$c_{B(\varepsilon)}\left(v\right) = \sum_{s \neq v \in V} \sum_{t \neq v \in V} \frac{\sigma_{st}^{\varepsilon}\left(v\right)}{\sigma_{st}^{\varepsilon}} \tag{2.11}$$

where $\sigma_{st}^{\varepsilon}\left(v\right)$ is the number of $\varepsilon$-shortest paths that include vertex $v \in V$, and $\sigma_{st}^{\varepsilon}$ is the number of $\varepsilon$-shortest paths between $s$ and $t$ in $G$. No analytical or empirical results on the stability of the metric was provided.

### Bounded-distance betweenness

In [BE06, Bra08], the authors limited the length of paths based on the idea that very long paths were only occasionally used, consequently not contributing to the betweenness centrality of a node. This metric was called bounded-distance betweenness centrality (also known as $k$-betweenness), where $k$ gives the maximum length of paths counted. The bounded-distance betweenness centrality of a vertex $v$ is defined as the sum of dependencies of pairs at most $k$ hops apart, that is,

$$c_{B(k)}(v) = \sum_{\substack{s \neq v \in V \\ d_{st} \leq k}} \sum_{\substack{t \neq v \in V \\ d_{st} \leq k}} \delta_{st}(v). \tag{2.12}$$

The bounded-distance betweenness centrality $(c_{B(k)})$ only considers contributions from shortest paths whose lengths are bounded by a constant $k$. For $k = n - 1$, it is equal to equation 2.2, and, for $k = 2$, it is similar to EBC (Section 2.3.1) differing in that shortest paths of length two with a non-neighbor as intermediate are also taken into account.

### Distance-scaled betweenness

Another variant of betweenness mentioned in [BE06, Bra08] counts paths of all lengths, but weights all shortest paths inversely in proportion to their length as in

$$c_{B(k)}\left(v\right) = \sum_{s \neq v \in V} \sum_{t \neq v \in V} \frac{\delta_{st}\left(v\right)}{d_{st}} \tag{2.13}$$

This metric is called *length-scaled betweenness* since the dependencies are scaled by a factor depending only on the length of the shortest path, being the same for all its inner vertices.

### $\alpha$-weight betweenness

The authors of [OAS10] proposed a variant of the betweenness centrality metric for weighted networks that incorporates both the number of ties between nodes (i.e., communication, cooperation, friendship, or trade) and their weights. The weight of a tie can have different meanings depending on the context. For example, in social networks, it can be seen as a function of duration, emotional intensity, intimacy, or exchange of services, whereas in non-social networks, it quantifies the capacity or capability of the tie, such as the number of seats among airports, or the number of synapses and gap functions in a neural network.

It is commonly assumed when analyzing shortest paths that intermediate nodes may increase the cost of interaction. If a high number of intermediate nodes is considered, the necessary interaction time between nodes increases.

Intermediate nodes are also in the position of powerful third-parties, being able to distort or delay information between nodes.

Let $\alpha$ be a tuning parameter which determines the relative importance of the number of ties compared to tie weights. So, the length of the shortest path between two nodes is given by

$$d_{st}\left(\omega\alpha\right) = \min\left(\frac{1}{\left(\omega_{si}\right)^{\alpha}} + \ldots + \frac{1}{\left(\omega_{it}\right)^{\alpha}}\right) \tag{2.14}$$

Equation 2.14 is an extension of the implementation in [Bra01, New01] of the Dijkstra's algorithm [Dij59] by taking into account the number of intermediate nodes. Both the tie weight and the number of intermediate nodes affect the identification of shortest paths. If $\alpha = 0$, the definition falls back to $d_{st}$, whereas if $\alpha = 1$, the definition falls back to Dijkstra's algorithm. If $0 < \alpha < 1$, a shortest path composed of weak ties is preferred over a longer one with short ties. On the other hand, if $\alpha > 1$, the influence of extra intermediate nodes is insignificant in comparison to the strength of the ties and paths with additional intermediaries are favored.

#### $k$-path betweenness

Similarly to the random-walk betweenness, $k$-path betweenness [KAS$^+$12] is based on the random traversal of a message from a source $s$. The following assumptions were made: (1) messages' traversals are only along single-paths, and (2) messages' traversals are only along paths of at most $k$ edges, where $k$ is network dependent.

The $k$-path betweenness centrality of a vertex $v$ is defined as the sum over all possible source nodes $s$ of the probability that a message originating from $s$ goes through $v$, assuming that the message's traversals are only along random simple paths of at most $k$ edges.

Let $p_{sl}$ be an arbitrary simple path with start vertex $s$ and having $l \leq k$ edges, i.e., $p_{sl} = \{s, u_1, u_2, \ldots, u_{l-1}, u_l\}$, and let $N(u_i)$ denote the set of outgoing neighbors of $u_i, \forall i \ : 0 \leq i \leq l$. For every vertex $v$ of $G$, $c_{PB(k)}\left(v\right)$ is given by

$$c_{PB(k)}(v) = \sum_{\substack{s \neq v \in p_{sl} \\ d_{sl} \leq l}} \sum_{1 \leq l \leq k} \frac{\chi[v]}{\prod_{i=1}^{l} \left|N(u_{i-1}) - \{s, u_1, u_2, \ldots, u_{i-2}\}\right|}, \tag{2.15}$$

where $\chi[v \ : v \in p_{sl}]$ is 1 if $v$ lies on $p_{sl}$, and 0 otherwise.

Table 2.5 presents a summary and comparison of betweenness centrality variants based on the type of variant, the betweenness metric, the main idea and relevant comments.

### 2.3.3 Algorithms

Betweenness centrality is one of the most widely used centrality metrics in social and complex networks analysis and it is based on shortest paths enumeration. Since it requires the computation of all shortest paths between a given pair of nodes, its exact determination is computationally-expensive. Betweenness computation requires $\mathcal{O}\left(n^3\right)$ time and $\mathcal{O}\left(n^2\right)$ space, where $n$ is the number of vertices in the network [Bra01].

In this section, algorithms used to compute standard betweenness centrality are presented. These algorithms can be either exact [Bra01] or approximate, and the latter can be subdivided according to the type of techniques used, namely random sampling [GSS08, BP07, RK14], adaptive sampling [BKMM07] and local techniques

| Variant | Betweenness metric | Main idea | Comments |
|---|---|---|---|
| Canonical-path betweenness | Shortest-path | Only a single canonical shortest path between any source-target pair is considered. | Used in road networks. |
| $\epsilon$-betweenness | Shortest-path | Consists in dynamically updating betweenness centrality in face of network's changes. | Used in terrorist networks analysis. |
| Bounded-distance betweenness | Shortest-path | Considers only contributions from shortest paths whose lengths are bounded by a constant $k$ | NA |
| Distance-scaled betweenness | Shortest-path | The longer a path, the less valuable it may be to control it. | NA |
| $\alpha$-weight betweenness | Shortest-path | It incorporates both the number of ties and their weights in weighted networks. | There are also variants for degree and closeness centrality that incorporate the tuning parameter. |
| $k$-path betweenness | Random-walk | It is based on a similar assumption about the random traversal of a message from a source $s$. It is assumed that the message's traversals are only along random simple paths of at most $k$ edges. | Nodes with high $k$-path centrality have high node betweenness centrality. |

Table 2.5: A summary and comparison of the variants of betweenness centrality

[Hin11].

### Exact Computation

**Brandes (2001).** In [Bra01], the author proposed an algorithm to evaluate simultaneously all centrality metrics based on shortest paths, thus reducing the algorithm's time and space requirements. The proposed approach integrates well with traversal algorithms that solve the single-source shortest-paths (SSSP) problem.

It can be seen from equation 2.2, that in order to determine betweenness centrality two steps are necessary: in the first step, it is necessary to compute the length and number of shortest paths between all pairs; in the second step, it is necessary to sum all pair-dependencies. The author observed that the second step of the betweenness centrality computation was responsible for its complexity.

For $s \neq v \in V$, the combinatorial shortest path counting is given by

$$\sigma_{sv} = \sum_{u \in P_s(v)} \sigma_{su} \tag{2.16}$$

By applying 2.16 in traversal algorithms like the Breadth First Search (BFS) [CLRS09] and Dijkstra's, and if the priority queue is implemented with a Fibonacci heap [FT87], the algorithms run times become $\mathcal{O}(m)$ and $\mathcal{O}(m + n \log n)$, respectively (where $m$ is the number of edges in the network).

To reduce the complexity of the second step of the algorithm, i.e., the need for explicit summation of all pair-dependencies, the concept of *dependency* of a vertex $s \in V$ on a single vertex $v \in V$, was introduced in [Bra01] as

$$\delta_s(v) = \sum_{t \in V} \delta_{st}\,(v) \tag{2.17}$$

and it was also observed that these dependencies obey a recursive relation.

**Theorem 2.3.1** ([Bra01]). *The dependency of $s \in V$ on any $v \in V$ obeys*

$$\delta_s(v) = \sum_{w \,:\, v \,\in\, P_s(w)} \frac{\sigma_{sv}}{\sigma_{sw}} \left(1 + \delta_s(w)\right) \tag{2.18}$$

**Algorithm 2.3.1.** *First, for each vertex $s \in V$ do a SSSP computation, maintain during the process the lists of predecessors $P_s(v)$. Then, for every $s \in V$ compute the dependencies $\delta_s(v)$ for all other $v \in V$ using the list of predecessors and the information along the directed acyclic graph of shortest paths. Finally, in order to obtain the centrality index of a vertex $v$, compute the sum of all dependencies values.*

Thus, for weighted and unweighted graphs, betweenness centrality can be computed in $\mathcal{O}\left(nm + n^2 \log n\right)$ and $\mathcal{O}(nm)$ times, respectively and $\mathcal{O}(n+m)$ space. For additional details, please refer to [Bra01].

### Approximation Techniques

Taking into account that, for large-scale graphs, the exact centrality computation is computationally-expensive, some approximation techniques have been proposed, which are introduced in the following sections.

### Random Sampling Techniques

**Brandes and Pich (2007).** The authors of [BP07] presented an experimental study of estimators for centrality metrics based on a restricted number of SSSP computations from selected source vertices (a generalized approach of [Epp04]). Source vertices, also known as *pivots*, are those from which the shortest path computations are initiated.

Let $X_1, X_2, \ldots, X_k$ be independent random variables, so that

$$\bar{X} = \frac{X_1 + X_2 + \ldots + X_k}{k} \tag{2.19}$$

and $\mu = E[\bar{X}]$ is the expected mean.

**Theorem 2.3.2** ([Hoe63]). *If $X_1, X_2, \ldots, X_k$ are independent, $a_i \le X_i \le b_i$, $i = 1, 2, \ldots, k$, then for $\xi > 0$*

$$\Pr\left\{\left|\bar{X} - \mu\right| \ge \xi\right\} \le e^{-2k^2\xi^2 / \sum_{i=1}^{k}(b_i - a_i)^2} \tag{2.20}$$

By 2.17, the contribution of the source vertex $s_i \in V$ to the centrality of a vertex $v \in V$ is given by $\delta_s(v)$. In order to extrapolate, for a single estimate, from the average contributions of $k$ source vertices, let

$$X_i(v) = \frac{n}{n-1} \delta_s(v) \tag{2.21}$$

be the random variable. To apply the bounds given by 2.20, let $a_i = 0$, $b_i = \frac{n}{n-1}(n-2)$, and $\xi = \varepsilon(n-2)$. Since the expectation of estimate $\frac{1}{k}\left(X_1(v) + X_2(v) + \ldots + X_k(v)\right)$ is the sum of all dependencies values on $v$, 2.20 guarantees that the error is bounded from above by $\varepsilon(n-2)$ with probability at least $e^{-2k\left(\frac{\varepsilon(n-1)}{n}\right)^2}$.

The authors of [BP07] concluded that in order for the contributions of $X_i(v)$ to be independent, the pivots needed to be selected at random. A drawback of this approach happens to unimportant nodes near a pivot, since it produces large overestimates of the betweenness centrality values. For example, if a one-degree node is selected as a pivot, the betweenness centrality of a two-degree node connecting the former to the rest of the network is overestimated by a factor of $\frac{n}{k}$.

**Geisberger et al. (2008).** In [GSS08], the authors proposed a generalized framework, which uses canonical centrality, for unbiased approximation of betweenness centrality to address the overestimates' problem of unimportant nodes near the pivots.

Let the proposed estimator be parameterized by

- $\ell : E \to \mathbb{R}$ on the edges, named *length function*.
- $f : [0,1] \to [0,1]$, named *scaling function*.

Let $P = (e_1, e_2, \ldots, e_k)\,,\ k = |E|$ be a path, such that $\ell\,(P) = \sum_{i=1}^{k} \ell\,(e_i)$. The algorithm performs, in each interaction, one of $2n$ possible (forward or backward) shortest path searches with uniform probability $1/2n$. A *scaled contribution* can be defined as

$$\delta_v = \begin{cases} \dfrac{f\big(\ell(S_{d_{sv}})/\ell\big(S_{d_{st}}\big)\big)}{\sigma_{st}} & \text{for forward search} \\ 1 - \dfrac{f\big(\ell(S_{d_{sv}})/\ell\big(S_{d_{st}}\big)\big)}{\sigma_{st}} & \text{for backward search} \end{cases} \tag{2.22}$$

So, $v$ gets the following contributions

$$\delta_v = \begin{cases} \sum\limits_{t \in V} (\delta_{st}\,(v) \ :\ S_{st} \in S_{st}(v)) \ := \delta_s(v) & \text{for forward search} \\ \sum\limits_{s \in V} (\delta_{st}\,(v) \ :\ S_{st} \in S_{st}(v)) \ := \delta_t(v) & \text{for backward search} \end{cases} \tag{2.23}$$

**Theorem 2.3.3** ([GSS08]). *If $X = 2n\delta\,(v)$ is an unbiased betweenness centrality estimator, then $E\,(X) = c_B\,(v)$.*

Hence, by averaging $k$ independent runs of $X_i\,(v)$, an approximation $\bar{X}$ of $c_B\,(v)$ can be obtained. The authors of [GSS08] proposed two implementations of their framework, namely a linear and bisection scaling. In the linear scaling, the contribution of the samples depends linearly on the distance to the sample, whereas in the bisection scaling, a sample only contributes on the second half of the path. According to the authors, both approaches perform better than the one proposed in [BP07], and the bisection approach even produced a good approximation for less important nodes with a small number of pivots.

**Riondato et al. (2014).** The authors of [RK14] proposed two efficient algorithms for betweenness centrality estimation, based on the random sampling of the shortest paths, which offer probabilistic guarantees on the quality of the approximation.

With a probability at least of $1 - \varphi$, both algorithms work as follows. The first algorithm estimates the betweenness of all vertices, and ensures that the approximate betweenness values are within an additive factor $\varepsilon$ from the real values. The second algorithm focuses on the top-*K* vertices with the highest betweenness. It returns a superset of the top-*K* vertices, while ensuring that the approximate betweenness value is within a multiplicative factor $\varepsilon$

from the real value. According to the authors, it is the first algorithm that can compute such approximation for the top-*K* vertices.

In order to derive the appropriate sample size necessary to achieve the desired approximation, Vapnik-Cherno venkis (VC) dimension theory [RK14] notions and results are used. A range set associated with the problem at hand is defined, and the upper and lower bounds to its VC-dimension are proven. So, the resulting sample size is independent from the number of vertices in the network, depending only on the vertex-diameter.

Let $\mathcal{I}_v(s,t) \subseteq \mathbb{S}_G$ be the set of all shortest paths, from $s$ to $t$, that $v$ is internal to

$$\mathcal{I}_v(s,t) = \{p \in S_{d_{st}} \ : \ v \in \mathrm{Int}(p)\} \tag{2.24}$$

The betweenness centrality of a vertex $v \in V$ in the normalized form is defined as

$$c_B\left(v\right) = \frac{1}{n\left(n-1\right)} \sum_{p_{st} \in \mathbb{S}_G} \frac{|\mathcal{I}_v(s,t)|}{\sigma_{st}} \tag{2.25}$$

**Theorem 2.3.4** ([HPS10, LLS00]). *Let $\mathcal{R}$ be a range set on a domain $D$ with $VC(\mathcal{R}) \leq d$, and let $\phi$ be a distribution on $D$. Given $\varepsilon, \varphi \in (0,1)$, let $S$ be a collection of $|S|$ points from $D$ sampled according to $\phi$, with*

$$|S| = \frac{c}{\varepsilon^2}\left(d + \ln\frac{1}{\delta}\right) \tag{2.26}$$

*where c is an universal positive constant. Then, $S$ is an $\varepsilon$-approximation to $(\mathcal{R},\phi)$ with probability at least $1-\varphi$.*

In order for the algorithm to compute a set of approximations for the betweenness centrality of the (top-*K*) vertices in a graph through sampling, with probabilistic guarantee on the quality of the approximations, let $d = \lfloor \log_2 \mathit{VD}(G) - 2 \rfloor + 1$. With 2.26, the resulting sample size $r$ is

$$r = \frac{c}{\varepsilon^2}\left(\lfloor \log_2 \mathit{VD}(G) - 2 \rfloor + 1 + \ln\frac{1}{\varphi}\right) \tag{2.27}$$

**Algorithm 2.3.2.** *Repeat $r$ times the following steps: First, sample a pair $s, t$ of distinct vertices uniformly at random. Second, compute the set $S_{d_{st}}$ of all shortest paths between $s$ and $t$. Third, select a path p from $S_{d_{st}}$ at random. Fourth, increase by $\frac{1}{r}$ the betweenness estimation of each vertex in* $\mathrm{Int}(p)$. *If the sampled vertices s and t are not connected, the third and fourth steps can be skipped.*

The unbiased estimator $\tilde{c}_B(w)$ for the betweenness $c_B\left(w\right)$ of a vertex $w$ is the sample average

$$\tilde{c}_B\left(w\right) = \frac{1}{r} \sum_{p_{st} \in S} |\mathcal{I}_w(s,t)| \tag{2.28}$$

where $S$ is the set of paths sampled in the algorithm. The desired accuracy and confidence are achieved with 2.27. For additional proof, please refer to [RK14].

**Adaptive Sampling Techniques** The adaptive sampling technique was proposed in [BKMM07] for estimating the size of the transitive closure of a directed graph. The proposed algorithm presented adaptive sampling of source vertices. To be precise, the information acquired from each sample depends on the number of samples.

**Bader et al. (2007).** In [BKMM07], the authors proposed an approximate algorithm for computing betweenness centrality of a single vertex, for both weighted and unweighted graphs, which is based on an adaptive sampling technique. The proposed approximation is a sampling algorithm, since centrality is estimated by means of sampling and SSSP computations of a subset of vertices, and, it is an adaptive algorithm, since the information acquired from each sample depends on the number of samples. So, this approach significantly reduces the number of SSSP computations for high centrality vertices.

The authors noted that through scores' extrapolation from a fewer number of path computations, centrality can be estimated in contrast to [Bra01] which estimates centrality scores of all vertices in the graph. Nonetheless, betweenness centrality scores are difficult to estimate, and the quality of the approximation was found to be dependent on the source vertices.

Let $a_i = \delta_{v_i *}(v)$ denote the dependency of the vertex $v_i$ on $v$, and let $A = \sum a_i = c_B(v)$ denote the quantity to estimate.

**Algorithm 2.3.3.** *Repeatedly sample a vertex $v_i \in V$; using a graph traversal algorithm, do a SSSP from $v_i$ and maintain a running sum $\chi$ of the dependency scores $\delta_{s*}(v)$. Sample until $\chi > cn$ (c is a constant and it is $\geq 2$). If k is the total number of samples, then the estimated betweenness centrality score of v is*

$$c_B(v) = \frac{n\chi}{k} \tag{2.29}$$

**Theorem 2.3.5** ([BKMM07])**.** *For $0 < \epsilon < 0.5$, if the centrality of a vertex $v$ is $\frac{n^2}{\gamma}$ for some constant $\gamma \geq 1$, then with probability greater or equal to $1 - 2\epsilon$ its centrality can be estimated to within a factor of $\frac{1}{\epsilon}$ with $\epsilon\gamma$ samples of source vertices.*

The authors of [BKMM07] demonstrated through experimental evaluation that their algorithm performed similarly for other vertices, besides those with high centrality (showed from theoretical results). For detailed discussion of the algorithm, please refer to [BKMM07].

### Local Techniques

**Hinne (2011).** The authors of [Hin11] proposed a local strategy to derive an approximation and the corresponding error bound's analysis of the true centrality metric using only the vertices directly adjacent to a target vertex. For example, the estimation of the vertex's centrality could be obtained by examining the vertex, its neighbors and its neighbor's neighbors.

Let the normalized version of equation 2.2 be defined as

$$c_B(v) = \frac{1}{(n-1)(n-2)} \sum_{s \neq v \in V} \sum_{t \neq v \in V} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{2.30}$$

To obtain a local approximation of 2.30, the term $\sigma_{st}(v)$ can be decomposed as $\sigma_{st}(v) = \sigma_{sv}\sigma_{vt}$. By applying this decomposition to the summation terms in 2.30,

$$\frac{\sigma_{st}(v)}{\sigma_{st}} \propto \sum_{\substack{v_0 \to v \to v_1 \\ v_0 \neq v_1}} \frac{\sigma_{v_0 v_1}(v)}{\sigma_{v_0 v_1}} \tag{2.31}$$

where $v_0$ and $v_1$ are the predecessors and successors of $v$, respectively. So,

$$c_B(v) \propto \tilde{c}_B^H(v) = \sum_{\substack{v_0 \to v \to v_1 \\ v_0 \neq v_1}} \frac{\sigma_{v_0 v_1}(v)}{\sigma_{v_0 v_1}} \tag{2.32}$$

where $\tilde{c}_B^H(v)$ is the local approximation of betweenness centrality, and $H$ is the local subgraph of $v$. The approximation error is given by

$$\left| c_B\left(v\right) - \tilde{c}_B^H(v) \right| = \sum_{s \not\to v \not\to t} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{2.33}$$

Since local subgraphs are smaller than the graph itself, there is a trade-off between computation and accuracy in local approximations. Specifically, smaller subgraphs lead to faster computations, decreasing as a result the approximation accuracy.

Table 2.6 presents a summary and comparison of the algorithms used to compute betweenness centrality based on the type of algorithm, the main idea and their complexity.

| Publication | Type | Main idea | Complexity |
|---|---|---|---|
| Brandes (2001) [Bra01] | Exact computation | Compute centrality of all graph's vertices in the same asymptotic time bounds as $n$ SSSP computations | $O(nm)$ – unweighted graphs<br>$O(nm + n^2 \log n)$ – weighted graphs |
| Brandes and Pich (2007) [BP07] | Random Sampling | The selection at random of source vertices is superior to deterministic strategies. | $O(km)$ – unweighted graphs<br>$O(k(m + n \log n))$ – weighted graphs<br>$k = \log n/\epsilon^2$ |
| Geisberger et al. (2008) [GSS08] | Random Sampling | Obtain good betweenness centrality estimates of unimportant nodes | |
| Bader et al. (2007) [BKMM07] | Adaptive Sampling | Reduce the number of SSSP computations of vertices with high centrality | |
| Riondato et al. (2014) [RK14] | Random Sampling | Compute betweenness estimation, based on the random sampling of the shortest paths, which offer probabilistic guarantees on the quality of the approximation | $O(r\ (n + m))$ – unweighted graphs<br>$O(r(m + nlogn))$ – weighted graphs |
| Hinne (2011) [Hin11] | Local Techniques | Gives a local centrality estimate based on a subgraph of vertices around a specific vertex | $O\left(nk^{2d}\right)$, $k^{2d} < m$ where $k$ is the average degree, and $d$ denotes distance |

Table 2.6: A summary and comparison of the algorithms used to compute betweenness centrality

### 2.3.4 DTN routing protocols

DTNs routing protocols face the troublesome task of finding a suitable next node to forward messages, due to the network's dynamics. This problem is augmented when additional requirements, such as good delivery probability or low end-to-end delay, are foreseen. Many routing protocols have been proposed up until now and in some, social network analysis is leveraged to enhance the delivery of messages. Some of the routing protocols hereby presented make use of many social metrics, where at least one is betweenness centrality. Let it be noted that betweenness centrality is computed over some inherent social network (see column 3 of Table 2.7), and not directly over the DTN.

### SimBet

The authors of [DH07] proposed a DTN routing protocol called *SimBet* that exploits two social metrics for data forwarding, namely betweenness centrality and social similarity[5]. If neither the sender nor its contacts know how to reach the destination node, the message is forwarded to a node structurally more central as its odds of discovering a suitable carrier are higher. Unlike previous works, no assumptions of control of node movements, or knowledge of future movements is made. It is assumed that only a single copy of each message exists in the network, which reduces resource consumption if compared to multi-copy strategies.

The authors argued that metrics based on direct or indirect encounters were not appropriate for discovering suitable carriers for routing messages, since some networks contained cliques (i.e., groups of nodes – clusters – that interacted more among them than with members of other clusters). So, node's centrality was estimated in the network in order to identify bridge nodes, i.e., message carriers among disconnected groups. Based on concepts from graph theory and network analysis, centrality is used to quantify a vertex's importance within graphs.

As previously stated, betweenness centrality measures the extent to which a node has control over information flowing between others. Thus, high betweenness centrality nodes are regarded as having a capacity to facilitate interactions between nodes they link, i.e., having a capacity of facilitating communication to other nodes in the network. It was also previously referred that a well-known disadvantage of centrality metrics is their computational complexity for large networks. Due to this, the authors used *ego networks* which do not require complete knowledge of the network as ego network analysis is performed locally by each individual node.

It was shown in [Mar02] that betweenness centrality based on egocentric measures is not equivalent to its sociocentric counterpart, despite the node's ranking based on both metrics being identical. Consequently, a comparison of locally calculated betweenness values between two nodes can be made, and the one with the higher value may be found. The betweenness values show 'how much a node connects nodes that are themselves not directly connected'. SimBet calculates betweenness centrality using an egocentric network representation of nodes with which the ego node has come into contact. Therefore, SimBet's ego betweenness centrality is given by equation 2.8.

### SimBetAge

In [ALV$^+$09], the authors proposed a DTN routing protocol for highly dynamic socially structured networks called *SimBetAge*. The routing scheme proposed in [DH07] was exploited while simultaneously taking into account social relations' strengths and gradual aging, i.e., the progression of the social network over time. Similarity and betweenness centrality were modeled over weighted graphs instead of unweighted ones as in [DH07].

A more realistic view of a social network is modeled by a weighted time dependent graph $G\left(T\right) = (V, E, \omega\left(e,T\right))$, where $G\left(T\right)$ is a fully connected graph, the weight $\omega$ is called *freshness* of an edge. If $e = (u,v)$ and $T \in \mathbb{T}$, $\omega\left(e,T\right) = 0$ means that the nodes in $e$ have never been connected; $\omega(e,T) = 1$ means a permanent connection between them. The freshness of a single edge can be perceived as an indicator of the likelihood of two nodes $u$ and $v$ being connected at a given time $T$, due to its representation as a logistic growth function (i.e., the contacts become fresher at each new encounter), and exponential decay function (i.e., the contacts become older

[5] Similarity expresses the amount of common features of a group in social networks. In sociology, the probability of two individuals being acquainted increases with the number of common acquaintances between them [MSLC01]. In computer networks, similarity between nodes $i$ and $j$ can be defined as the number of common neighbors among them. Therefore, the more common neighbors they have, the more similar they are.

with time). The freshness of a path $\omega\left(p, T\right)$ is defined by the product of all freshness values in it.

The authors proposed a new metric called *egocentric flow betweenness* defined as

$$c_{EFB}\left(v\right) = \sum_{u,w \in V^*(v)} \frac{\left(\omega_{vu}\omega_{vw}\right)^2}{\omega_{vu} + \sum_{u \neq v \in V^*(v)} \sum_{w \neq v \in V^*(v)} \omega_{vu}\ \omega_{vw}}, \tag{2.34}$$

i.e., $c_{EFB}\left(v\right)$ is the sum over the age of all paths between pairs of nodes $u, w \in V^*(v)$ passing through $v$ divided by the age of all possible paths between them, and weighted with the age of the edges between $v$ and $u$, and $v$ and $w$. If two nodes $u$ and $w$ want to compare their utilities to a destination $t$ in an ego-centric manner:

- *Betweenness* should be used, if both nodes are far away from $t$;
- *Similarity* should be used, if $t$ is at most two hops away of any of them;
- *Directed betweenness* should be used otherwise, as it considers only the paths containing $t$ instead of all possible paths in the neighborhood of $v$. Directed betweenness is defined as

$$c_{DB}\left(v, t\right) = \sum_{\substack{u,w \in V^*(v) \\ t \in V^*(v)\ \vee\ u = t}} \frac{\left(\omega_{vu}\ \omega_{vw}\right)^2}{\omega_{vu} + \sum\limits_{u \neq v \in V^*(v)} \sum\limits_{w \neq v \in V^*(v)} \omega_{vu}\ \omega_{vw}} \tag{2.35}$$

**Bubble Rap**

In [HCY11], a DTN routing protocol for pocket switched networks (PSN) called *Bubble Rap* was proposed. A PSN is a network without infrastructure composed of a multitude of devices carried by persons. Therefore, two social metrics, namely community and centrality, are exploited for data forwarding, instead of mobility due to the network's unpredictability and highly dynamic topological structure. Node mobility is used by MANET and DTN routing algorithms to build and update their routing tables. In sociology, a community can be defined by a group of people living in the same location. So, people from the same community tend to interact more often between themselves, than with a randomly chosen member of the population [Oka05]. Another important aspect to consider within a community is the degree of interaction among its members. Usually, some members tend to interact more than others. For example, the postman meets customers more often, in comparison to a network engineer. As a consequence, there are more popular members (hubs) which have higher centrality values, and these popular hubs are a better choice for relays than unpopular ones.

In [HCY11], the authors assumed that each node belonged to at least one community, and that it had a global (for the whole system) and local (for its local community) centrality values. Also, a node could belong to a single node community or to multiple communities, thus having multiple local centrality values.

A PSN is modeled as a temporal network (or a time evolving network) due to its characteristics. Thus, betweenness centrality of a node in a temporal graph is obtained by counting the number of times a node acted as a relay for other nodes on all the shortest delay deliveries[6], over a large number of emulations of unlimited flooding with different uniformly distributed traffic patterns.

In order to approximate centrality, the authors found out that the degree per unit-time (e.g., the number of unique nodes seen per *t* hours) and the node centrality had a high correlation value (for *t* = 6, the correlation

[6] The delivery with shortest delay is when the same message is delivered to the destination over different paths.

coefficient is 0.9511). This correlation led them to conclude that what mattered was the frequency of interaction, not the number of known persons. They compared the average unit-time degree with a greedy ranking algorithm called RANK, and found out that they performed similarly. RANK, which is similar to the greedy strategy in [ALPH01], assumes that each node only knows its ranking and the ranking of those it encounters. But, it does not know the ranking of the other nodes it does not encounter, neither does it know which node has the highest rank in the system. However, since in a distributed manner, it is difficult to compute the average unit-time degree individually throughout the whole experiment, two approaches were proposed, namely: (1) the single window (S-Window) approach, in which upon an encounter, nodes compare how many unique nodes they met in the previous unit-time slot; (2) the cumulative window (C-Window) approach, which consists in calculating the average value on all previous windows (e.g., from yesterday until now), and then calculating the average degree for every *t* hours. C-Window is similar to a statistical technique called exponential smoothing [Win60].

The authors used two centralized community detection algorithms, namely *K*-CLIQUE [PDFV05] and weighted network analysis (WNA) [New04], to identify local community structures as it would be helpful in designing good strategies for information dissemination. They use the two since each has useful features and they complement each other.

Bubble Rap Forwarding works as follows: if a node wants to send a message to a destination, this node (the sender) first bubbles (forwards) this message up based on the global centrality until it finds a node which is in the same local community as the destination of the message. Then, the local centrality is used instead of the global one, and the node continues to bubble up the message based on the local centrality through the local community until the destination is found or the message expires.

**PQBCF**

The authors of [NLSD13] proposed a peer-to-peer (P2P) query algorithm based on betweenness centrality forwarding (PQBCF) for Social Opportunistic Networks (SONs). In SONs, mobile devices are carried by people and consequently the mobility model exhibits social characteristics. Since betweenness centrality quantifies the importance of nodes in message delivery throughout the network, nodes with higher betweenness values can be seen as more active and having more opportunities of encountering more nodes. So, they are naturally good candidates to act as relay nodes.

To share or publish contents in SONs, a message dissemination scheme commonly used is the P2P inquiry/response[7] that works as follows: first, a query node sends an inquiry message throughout the network searching for a node with the response message; then, the response message is forwarded to the query node.

In PQBCF, a node looking for some data, e.g., an audio file, generates an inquiry message containing the description of the required data. To reduce the expected query latency, assuming that many nodes contain the requested data, multiple copies of the inquiry message are created. With more copies, the query delay reduces, but the network overhead also increases as a result of the additional number of message copies occupying nodes' buffers, as well as more message transmissions. Therefore, a tradeoff between the query delay and overhead should be obtained by the inquiry/response scheme. PQBCF achieved this by calculating the number of copies of the inquiry message in the network based on the expected query delay, the mobility and the nodes density.

[7] In [BBQ$^+$05], an inquiry/response scheme was proposed combining content-based routing with probabilistic-based routing.

During the inquiry message dissemination, betweenness centrality is used as the metric for relays selection. If the inquiry message is passed to a node with information matching the inquiry message, it creates a response message, calculates the number of copies of the response message, and sends it back to the inquiry node.

The betweenness centrality of a node $v$ is defined as

$$c_B(v) = \frac{2}{(n-1)(n-2)} \sum_{s \neq v \in V} \sum_{t \neq v \in V} \frac{g_{st}(v)}{g_{st}} \tag{2.36}$$

where $g_{st}$ is the number of all the messages successfully delivered between a pair of nodes $s$ and $t$, and $g_{st}(v)$ represents the number of messages that passed through node $v$ during the forwarding process. The ratio $\frac{g_{st}(v)}{g_{st}}$ indicates the importance of node $v$ in delivering messages between $s$ and $t$. If the destination node is fixed, the importance of node $v$ in delivering messages to the destination node $t$ is given by

$$c_{B_t}(v) = \frac{2}{(n-1)} \sum_{s \neq v \in V} \frac{g_{st}(v)}{g_{st}} \tag{2.37}$$

Each node maintains a table with the necessary information to calculate $c_B(v)$ and $c_{B_t}(v)$ which can be obtained in a distributed manner.

### GrAnt

Multi-agent systems in which the behavior of an agent (also known as artificial ant) is inspired by the behavior of real ants, are called ant systems. The Ant Colony Optimization (ACO) metaheuristic [DD99], a particular class of ant algorithms which use artificial swarm intelligence [BDT99], is inspired on an experience by Gross et al. [GADP89] using an ant system. Some example problems where ant algorithms have been used are classical traveling salesman and routing in telecommunications networks.

In [VMDV11], the authors proposed Greedy Ant (GrAnt), a prediction-based routing protocol for DTNs, which uses a greedy transition rule of the ACO metaheuristic aiming at exploiting, if available, good previous solutions, and to select the most suitable message forwarder.

To cope with DTN, the following modifications were proposed allowing to differentiate GrAnt from traditional ACO algorithms: (1) to increase the possibility of reaching the destination, forwarder ants, which are ant agents responsible for discovering paths to the destination nodes, are encapsulated into data messages; (2) to find a path to an unknown destination, a dynamic number of forwarder ants, whose computation takes into account the utilities of the already established message forwarders and the success of the message delivery, is used; (3) to provide exploitation of good solutions already found or to forward the message to the most promising node, a greedy ACO transition rule is used while considering heuristic functions and pheromone concentration. The pheromone concentration indicates how useful a global solution was, which serves as a history of the best previous movements of the ants. The heuristic function values indicate an explicit influence towards more useful local information; (4) besides the best path, redundant ones are also allowed due to the dynamics of DTNs. An event-driven evaporation happens if and only if a node detects a new path being constructed to the destination. And, since the pheromone deposited by ants is based on information about nodes in each constructed path, the evaporation process prevents the occurrence of undue convergence of the algorithm to the same subset of paths.

The GrAnt protocol provides modules for (*i*) routing, by determining which route a message should follow to reach its destination. The forwarding decision consists in adopting a greedy transition rule that considers the pheromone at a link in the path to the destination (or local heuristic information, in the absence of pheromones) and the heuristic function associated with an intermediate node in the path to the destination, (*ii*) scheduling, by deciding in which order messages must be transmitted, and (*iii*) buffer management, by indicating which message(s) must be dropped whenever the buffers occupancy limit has been reached.

The heuristic function is based on two criteria: $\mathrm{Social}_{vt}$, representing the social proximity between nodes $v$ and $t$, and $\mathrm{BetwU}_{vt}$, representing the betweenness utility of node $v$ in relation to the destination $t$. The node betweenness utility computation is slightly different from [Fre78]. In order to have a high betweenness utility to a destination $t$, a node $v$ must appear with high frequency in paths between any source node and the destination $t$. So, differently from [Fre78] and [DH07], no shortest path verification is required by betweenness utility and no list of all previous encounters is exchanged, respectively.

### Kim et al. (2014)

In [KHYJ14], the authors proposed a routing scheme by using DNI (i.e., node's local contact history) and SNI (i.e., the expanded ego-network betweenness centrality). Since computing the real betweenness involves global network knowledge, it is in general impractical in DTNs due to the lack of network-wide end-to-end connectivity. Therefore, each node computes betweenness by means of its local expanded ego network built using the node's social network composed of information of its neighbors and of its neighbors' neighbors. The result is used as an estimate of its true betweenness over the entire network due to their high correlation [KHYJ13].

Routing is composed of two strategies, namely, edge weight and centrality based strategies. In the former, each node calculates the edge weight [BS10] using DNI. If the edge weight is high between a pair of nodes it means that there is a high future contact probability. Similarly to [BS10], a node carrying a message to a remote destination forwards it to a given relay node if the edge weight between the relay and the remote destination is higher than the one between him and the destination node. Centrality based strategies are used to improve routing efficiency. Each node constructs its own social network, and calculates the expanded ego betweenness centrality. As some nodes might get very low edge weights since they hardly meet with other nodes, and if these isolated nodes are the destination, proper relays might be difficult or even impossible to find by the source node using the former strategy. Messages would probably be discarded due to Time-To-Live (TTL) expiration. Therefore, the node carrying the message also forwards the message to another node if it presents a higher value of betweenness centrality even though it presents a lower edge weight value, since nodes with higher betweenness centrality values are more socially related to other nodes.

Additionally, a message management scheme was also proposed to reduce the overall delivery cost. A node carrying a message can delete the message from its buffer after forwarding it to another node with an edge weight higher than all edge weights in this node's social network.

### LocalCom

Previous works [DH07, HCY11, CHC$^{+}$07] confirmed that with high probability nodes in DTNs tend to meet more a certain group of nodes than other nodes outside this group, and that the grouping structure remains stable over time. Hence, it is of interest to utilize the grouping structure of DTNs to facilitate message forwarding.

In [LW09], the authors proposed LocalCom, a community-based epidemic forwarding scheme for routing, which efficiently detects the community structure, using limited information, and improves the forwarding efficiency based on the community structure. In LocalCom, the statistics of the separation period is selected in order to shorten nodes' knowledge. Based on the frequency and length of the node's contacts, each node calculates the average separation period towards its neighbors. LocalCom also applies the Gaussian similarity function [Lux07] to represent the closeness in the relationship. A closer relationship is reflected by a shorter average separation period. At the same time, an irregularity in the relationship is reflected by the variance of the separation period. Therefore, closeness and irregularity metrics are used to deduce the similarity metric, which shows the relationship between each pair of nodes in the network. Similarity also captures the core temporal and spatial encounter information.

Differently from [DH07, HCY11], a distributed scheme was developed and it only requires local information to form communities in LocalCom. It uses an extended clique, which is based on virtual links, to represent underlying community structures. A virtual link allows the representation of a neighboring relationship between a pair of nodes, if at least one path with up to $k$ hops exists between them.

High similarity and short hop-count distances that are some of the desirable properties within a community can aid intra-community communication based on the single-copy source routing. So, packets will be directly forwarded along a virtual link. Through flooding, inter-community packet forwarding is performed using nodes that have direct neighboring relationship with nodes in other communities (also called gateways). Since not all gateways are necessary, some pruning is performed to avoid unnecessary redundancy. Bridges, which are the actual forwarding nodes are selected from gateways using two marking and pruning schemes: static pre-pruning and dynamic pruning. The former is conducted by each gateway based on local information. Nodes marked as bridges during the former further define their role dynamically based on additional information received.

In order to forward a packet between nodes residing in different communities: first, the inter-community forwarding mechanism is used to forward the packet to the current communities' bridges, and then, the bridges forward the packet to other communities they are connected to. Each gateway calculates its centrality for the communities it connects. The betweenness centrality of a gateway $v$ in community $A$ connecting community $B$ is given by

$$c_{GB}\left(v\right) = \sum_{s \in A} \sum_{t \in B} \frac{\sum_{p \in P_{st}(v)} \left( \prod_{(i,j) \in p} \omega_{ij} \right)}{\sum_{p \in P_{st}} \left( \prod_{(i,j) \in p} \omega_{ij} \right)} \tag{2.38}$$

where $P_{st}$ represent the set of all paths between $s$ and $t$ in the neighboring graph, and $P_{st}\left(v\right)$ denotes the subset of $P_{st}$ containing all the paths from $s$ to $t$ that pass through $v$. The numerator and denominator are the sum of all path weights in $P_{st}\left(v\right)$ and the sum of all path weights in $P_{st}$, respectively.

Each gateway should calculate its centrality values, that is, one distinct for each community it connects, and send through the virtual links among them the centrality value to all other nodes in its local community. Therefore, each gateway knows all other gateways in its community connecting to other communities, also knowing their centrality values.

### CAOR

The authors of [XWH14] proposed the community-aware opportunistic routing (CAOR) algorithm for Mobile Social Networks (MSNs) [WXH13] using two social metrics, namely community and centrality. A MSN can be seen as a social DTN since it is composed of mobile nodes with social characteristics[8]. Based on this social characteristic, a home-aware community model was proposed in which mobile users with a common interest form, by themselves, a community where the frequently visited location is their common *home*. Similarly to [WXH13], the authors assume that each home supports a real or virtual throwbox [INC09], i.e., a local device that can temporarily store and transmit messages.

The rationale behind CAOR is to turn the routing between lots of mobile nodes to the routing between a few community homes. Therefore, message delivery can be turned into the delivery within and between these communities. Two centrality metrics are used to measure the importance of nodes during message delivery, specifically: intra-community centrality and inter-community betweenness metric. The former consists in measuring the capability of each community member to meet and deliver messages to other members, and the node with the largest intra-community centrality in a community has the best capability to deliver messages. The latter consists in measuring the ability of a node set to be taken as a communication bridge between communities. Here, the delivery delay is used to evaluate the inter-community betweenness of a set of nodes.

Let an MSN be composed of $|V|$ nodes $V = \{v|v \in V\}$ moving among $|L|$ locations $L = \{l|l \in L\}$ such that $(|L| \ll |V|)$. For two overlapped communities $C_l$ and $C_{l'}$ and an arbitrary relay set $S(S \subseteq C_l \cap C_{l'})$, the inter-community betweenness is given by

$$c_{B_{l,l'}}(S) = \frac{1}{\sum_{v \in S} \lambda_{v,l}} + \frac{\sum_{v \in S} \frac{\lambda_{v,l}}{\lambda_{v,l'}}}{\sum_{s \in S} \lambda_{v,l}} \tag{2.39}$$

where $\lambda_{v,l}$ and $\lambda_{v,l'}$ are parameters of the exponential distribution followed by the interval of node $v$'s visits to homes $l$ and $l'$, respectively. In other words, the inter-community betweenness is the expected delay that it takes for a relay node to cooperatively deliver messages by means of an opportunistic routing scheme from one community to another.

The optimal betweenness, which corresponds to the relay set with the smallest betweenness for the message delivery from the community home $l$ to $l'$, is given by

$$c_{\tilde{S}_{l,l}} = \underset{S \subseteq C_l \cap C_{l'}}{\operatorname{argmin}} \, c_{B_{l,l}}(S) \tag{2.40}$$

The CAOR algorithm consists of an initialization and routing phases. The initialization phase builds $|L|$ community homes from a network with $|V|$ nodes, thus simplifying the network. Then, under the home-aware community model, the routing phase delivers messages based on the optimal opportunistic routing rule, that is, the message sender always delivers messages to the encountered relay that has a smaller minimum expected delay to the destination than itself.

[8] For instance, in many real MSNs, mobile users with common interests tend to visit some location (real or virtual) that is related to this interest.

### Hoten

In [YMF15], a forwarding metric, known as Hoten (HOTspot ENtropy), which consists of three social metrics, namely betweenness centrality, similarity and personality, was proposed to improve the performance of routing in opportunistic networks. The authors focused on the integration of social structure into data forwarding algorithms since existing algorithms, such as SimBet, Bubble Rap and People Rank [MMDA10], did not fully exploit social structures extracted from real world traces (e.g. human walks [GC10]). Similarly to [LHK$^+$08], the authors confirmed the existence of two known phenomena by analyzing GPS traces of human walks, i.e., on the one hand, people always move around a set of well-known locations, called *public hotspots* (instead of purely random walks), and, on the other hand, each people shows preference for some particular locations, called *personal hotspots*. They also assumed hotspots were more stable than the social structure of existing algorithms, as for example, public hotspots were formed by overlaying personal hotspots together and personal habits were stable over time and across situations [AA21].

Information theory [CT12] is used to compute the nodes' social metrics since the entropy represents the degree of disorder or randomness in a system, i.e., the bigger the entropy value is, the more disordered the system is. To compute betweenness centrality, the authors used the relative entropy[9] [AB09] (also called Kullback-Leibler divergence) between the public hotspots and the personal hotspots. Similarity between two nodes was computed by exploiting the inverse symmetrized entropy of the personal hotspots between them. The entropy of personal hotspots of a node is used to estimate its personality.

Let $K$ denote the total number of hotspots in the network and let $n_i$ denote the number of stay points in hotspot $i$. The weight of the hotspot $i$ is given by $\omega_i = \frac{n_i}{\sum_{i=1}^{K} n_i}$. In the same way, let $n_{\mathtt{p}_i}^{j}$ denote the number of $i$th person's stay points in $j$th hotspot. The weight of $j$th hotspot influenced by the $i$th person is given by $\omega_{\mathtt{p}_i}^{j} = \frac{n_{\mathtt{p}_i}^{j}}{\sum_{i=1}^{K} n_{\mathtt{p}_i}^{j}}$.

Let $X_i$ be a random variable denoting the distribution of personal hotspots of node $i$, and let $Y$ be a random variable denoting the distribution of public hotspots. So, Y = $\omega_1,\ \omega_2,\ \ldots, \omega_k$ and $X_i = \omega_{\mathtt{p}_i}^{1}, \omega_{\mathtt{p}_i}^{2}, \ldots, \omega_{\mathtt{p}_i}^{k}$. The betweenness centrality of node $v$ is given by

$$c_B\left(v\right) = \left[\sum_{j=1}^{k} \omega_{\mathtt{p}_v}^{j} \log\left(\frac{\omega_{\mathtt{p}_v}^{j}}{\omega_j}\right)\right]^{-1} \tag{2.41}$$

If equation 2.41 is compared with equations 2.2 and 2.8, one can conclude that it has low time complexity $\mathcal{O}(k)$ since it (*i*) is only related to the top $k$ hotspots and (*ii*) is independent of the number of nodes in the network.

The Hoten routing algorithm works as follows: when a node meets with another node, the node delivers to the other node any message it carries destined to the other node, and removes the message from its messages' queue. If a message is not destined to the other node, both nodes swap their Hoten forwarding metrics (also known as Hoten utility) for that message. If the node's Hoten utility is smaller than that of the other node for the given message, the node delivers the message to the other node and removes the message from its message queue, thus taking a single copy approach.

[9] Relative entropy can be used to differentiate the divergence between two random variables[YMF15].

### Other approaches

In [LSHG16], a packet forwarding algorithm based on pheromones was proposed. The latter is calculated analyzing the connection and disconnection times between pairs of nodes. The pheromone of the destination node is computed periodically and then spread to the surrounding neighbors via an attenuation model [LSHG16]. Every node that receives it, must update itself and spread it again using the same attenuation model. A node must decrease the pheromone if he does not receives it within a certain period of time. A node detaches from the community when the pheromone reaches zero. Communities are constructed taking into account the pheromone to the destination node. In this approach, packets are forwarded to nodes along the direction were the pheromone is stronger until they reach their destination. The betweenness centrality of a node corresponds to the pheromone held by the candidate relay to the destination node.

The authors of [GSK15] proposed a node scheduling approach for DTNs consisting of two metrics, namely community and centrality. Each node uses a global centrality metric consisting of ego betweenness and degree centrality, if no community information about the destination of the message is available, hoping to find the destination or another node belonging to the destination's community.

In [XZD$^+$16], a social-aware data forwarding approach for smartphone-based DTNs was proposed. Two utility functions were combined to derive the social strength among users and their importance. On the one hand, the social context information of the nodes is used to calculate the social similarity utility between a node and destination, and on the other hand, the social connection of networks is used to calculate the betweenness centrality utility of a node.

Table 2.7 presents a summary and comparison of DTN Routing protocols using betweenness centrality based on the social metrics used, the type of graph used, the main idea of the routing protocol, the type of standard betweenness centrality algorithm, optimizations, the performance evaluation and DTN scenarios and/or applications. Note that the performance evaluation presented, when available, is limited to the surveyed social routing protocols.

| Publication | Social metrics | Graph Type | Main idea | Type of algorithm | Optimizations | Performance evaluation | Scenarios / Applications |
|---|---|---|---|---|---|---|---|
| SimBet [DH07] | Egocentric betweenness and Similarity | Unweighted graph | The message is forwarded to a node structurally more central. | Local Technique | NA | Delivery performance close to Epidemic, but without the overhead. | Disconnected Delay-Tolerant MANETs |
| SimBetAge [ALV+09] | Egocentric flow betweenness and Similarity | Aged Graph | It is an extension of SimBet that takes into account the progression of the social network over time. | Local Technique | NA | It outperforms SimBet in terms of delivery rate. | PSNs |
| Bubble Rap [HCY11] | Betweenness centrality (degree centrality per unit time) and Community | Weighted Temporal Graph | Nodes bubble up messages first using global centrality and then using local centrality. | NA | Controlled message replication. Original carrier deletes the message once the destination community is identified. | Delivery ratio close to SimBet, but much lower resource utilization. | PSNs |
| PQBCF [NLSD13] | Betweenness centrality | NA | A query node sends an inquiry message throughout the network searching for a node with the response message using betweenness centrality as the metric for relay's selection. | NA | The number of inquiry messages is a tradeoff between the query delay and overhead. | Inquiry success ratio and delay better than flooding for high message generation frequency. No social routing protocol was considered. | SONs |
| GrAnt [VMDV11] | Betweenness utility | NA | The next node is chosen using pheromone concentration if available, or local information captured from DTN nodes. | NA | NA | Achieves higher successfully message delivery and lower overhead than Epidemic in community-based movement model. No social routing protocol was considered. | DTNs |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| LocalCom [LW09] | Community, similarity and betweenness centrality | Neighboring Graph | The intra-community forwarding mechanism is first used to forward the packet to the current communities' bridges, and then, the bridges forward the packet to other communities they are connected to | Exact Computation | Community level broadcast if the source and destination are in different communities. | Being simple flooding the upper bound in terms of the delivery ratio, LocalCom outperforms other protocols, (Bubble Rap included). But in terms of number of forwards (overhead), Bubble Rap represents the lower bound for all the scenarios considered. | DTNs |
| Kim et al. [KHYJ14] | Expanded ego betweenness centrality | Weighted Contact Graph | Each node first uses DNI to choose a proper relay node. Then, SNI is used to enhance routing efficiency. | Local Technique | Message delivery cost is reduced by deleting messages forwarded to nodes with the highest edge weight. | More delivery efficient routing in comparison to Epidemic. | DTNs |
| COAR [XWH14] | Betweenness centrality and community | Contact Graph | Build home-aware communities and use optimal opportunistic routing rule to route messages among these communities. | NA | Each home only forwards its messages to the node in its optimal relay set. | It outperforms Bubble Rap and SimBet in terms of delivery rate and average delay. | MSNs |
| Hoten [YMF15] | Betweenness centrality, similarity and personality | NA | Each node only forwards a message if the Hoten utility for a given destination is smaller. | NA | Single-copy approach. | It outperforms SimBet and PeopleRank in terms of delivery rate. | DTNs |

Table 2.7: A summary and comparison of DTN Routing protocols that use betweenness centrality

### 2.3.5 Discussion

In WANETs, such as DTNs, the dynamics of the network constitutes a challenging task to routing protocols and as a result of that end-to-end connectivity between any pairs of nodes might never exist. However, by using a store-carry-and-forward approach, DTN nodes can carry messages with them while moving until an appropriate node is found. In this approach, messages are relayed from one node into another until they reach their destination, or they are discarded. In order to find the most suitable forwarding node, static and dynamic network information is used. Among the available network information, static network information has been adopted by a considerable number of social routing protocols due to its stability tendency over time, hence leveraging the use of social metrics. Still, despite the advantages of using social metrics, single-property social routing protocols may experience difficulties finding the destination node. If, for example, centrality-based metrics are being used, the node carrying a message may not select, as the next message carrier intermediate nodes having lower centrality than the current carrier. But, depending on the network topology, intermediate nodes with low centrality may also have high odds of encountering the destination node. This led most of the surveyed DTN routing protocols to be hybrid, although some properties might be non-social.

In this subsection, a survey of betweenness centrality concepts, variants and standard algorithms is presented. Additionally, a survey of DTN routing protocols that use betweenness centrality, and a discussion on how the metric, its algorithms are used by the protocols is also provided. Previous work has shown that centrality metrics, which are used to point out the (relative) importance of vertices and edges in networks, are of considerable relevance for DTN routing protocols. Since mathematically these metrics are simple to grasp, their actual calculation is by far much more elaborate, due to the network's size and dynamics. Because of that approximate algorithms are more common means of calculation as an alternative to the exact computation.

The surveyed protocols can be organized in three groups based on the type of algorithms, namely: (*i*) approximate algorithms, (*ii*) exact algorithms, and (*iii*) alternative heuristics.

Ego networks, which fit in the first group, are used to reduce the complexity associated with the computation of betweenness centrality using partial network knowledge. Since ego network analysis is performed locally by each individual node, egocentric betweenness centrality can be seen as a local technique similar to the one described in [Hin11]. In [KHYJ14], the authors used an expanded ego network, i.e., an ego network where the second degree neighbors of a given node are also considered. In [DH07], betweenness centrality is only updated upon hello message reception from new nodes.

The two betweenness centrality metrics (egocentric flow betweenness and directed betweenness) proposed in [ALV$^+$09] were envisaged for highly dynamic social networks, as instead of the number of shortest paths, their calculation takes into account all possible paths in a network. The difference between them is that in the latter only paths containing the destination are considered. Also, since both metrics consider nodes in the neighborhood of a given node, these algorithms are based on a local technique. A Socially-Aware Multi-Phase Opportunistic (SAMPhO) [VYL14] routing protocol was proposed, in which ego betweenness is used according to the conditions of the social environment in the centrality-based forwarding phase. The authors of [GCPR14] proposed two distributed EBC protocols (EBC broadcast and gossip) for distributed SONs. EBC broadcast and gossip differ from each other in the update phase of the adjacency matrix. In the former, the adjacency matrix is kept updated by each node, by doing communications with all nodes in its ego network. In the latter, the adjacency matrix is kept updated through specific gossip techniques [Jel11]. A bridging centrality metric [HKRZ08] that is calculated by multiplying betweenness centrality by a bridging coefficient [HKRZ08] is used in [WZGX14]. But, instead of using the shortest-path version that requires global network knowledge, ego betweenness centrality was used.

The routing protocol proposed in [LW09] is the only one using an exact algorithm (hence, belonging to the second group) for the betweenness centrality computation on a weighted graph. This routing protocol uses three social metrics, namely similarity, community, and betweenness centrality. Similarity, which is based on closeness and irregularity metrics, is used to build the neighboring graph. With the graph, a distributed scheme is used to identify communities which are used during the intra-community forwarding. If the source and destination nodes are in different communities, flooding is used for intra-community forwarding and betweenness centrality for inter-community through bridges.

Some of the routing protocols proposed use alternative heuristics to compute betweenness centrality. In [NLSD13], betweenness centrality is computed using the number of successfully delivered messages. No global network knowledge is necessary as when nodes meet, they synchronize their reserved ratios of successful message delivery values and update their betweenness centrality values to a given destination. In [VMDV11] and [VMDV12] a metric called betweenness utility was proposed to measure the importance of a given node in delivering messages to a certain destination node. Differently from [Fre78], no shortest path verification is required by the betweenness utility, nor a list of all previous encounters is exchanged, in contrast to [DH07]. The hybrid protocol proposed in [VMDV12] infers the most suitable next node to forward messages by means of opportunistic social information. It also determines the best path to forward each message while limiting message byte redundancy. In [HCY11], the authors approximate centrality using the degree per unit-time, as the two metrics are highly correlated. In [ZLG$^{+}$11], a centrality metric was proposed that uses the expected number of packets which can be transmitted from a given node to others within the time constraint, as the centrality metric in [HCY11] only considers the frequency of contacts and disregards their duration. Equally, in [JHMB11] a generalized model of the centrality metric was proposed that allows the calculation of the expected delivery performance metrics (delivery latency or delivery cost) of a given message. In [XWH14], the expected delivery delay is used to evaluate the inter-community betweenness of a set of nodes, and in [YMF15] relative entropy, from information theory, is used to compute betweenness centrality.

Previously, six definitions of betweenness centrality were presented. They can be used in static or dynamic networks, and use partial or global network information. The most appropriate centrality metrics for DTNs are flow, random-walks, ego and temporal betweenness centrality, since they do not require global knowledge of the network (see Table 2.4). Among them, flow and ego betweenness have been implemented in DTN routing protocols, as shown in Table 2.7. Please note that SimBetAge uses an egocentric version of flow betweenness centrality, thus not requiring global network knowledge. Most of the works analyzed compare their approach with Epidemic routing, a non-social routing approach, as it is considered as the upper bound in terms of delivery ratio. While designing routing protocols, there is always a tradeoff between delivery ratio and overhead. For example, on the one hand, there is LocalCom that outperforms Bubble Rap in terms of delivery ratio by using flooding to increase its probability of successful packet delivery, and the schemes considered could only achieve a delivery ratio of 30% to 40% when the TTL expiration was set to three days in the Reality scenario (MIT Reality Mining [EP05]) [LW09]. CAOR significantly outperforms SimBet and Bubble Rap in another scenario (MSN trace from the WiFi campus of Dartmouth College [KHAY07]) in terms of delivery ratio and average delay (which was not considered as an evaluation metric in [LW09]). Specifically, when compared with SimBet and Bubble Rap, CAOR increases the delivery ratio by about 89.5% and 35.8%, and reduces the delivery delay by about 49.6% and 22.7%, respectively [XWH14]. On the other hand, there is Bubble Rap that because of using a more conservative replication strategy presents a lower overhead in comparison to LocalCom [LW09]. But both use betweenness centrality during forwarding as means to reach the destination. Still, the message delivery efficiency, used by Kim et al. [KHYJ14], incorporates both message delivery ratio and overhead, and because of that can be seen as a good indicator of a protocol overall performance.

In addition, another important aspect that was taken into account in SimBetAge was the fact that social relations and the roles of individual nodes change over time. Likewise, some relations are stronger than others resulting from, for example, a higher contact frequency.

Despite the efforts of innumerous researchers, the use of social metrics by DTN routing in still under research. An interesting point of research is for DTN routing protocols to consider in addition to the shortest paths, the fastest ones in terms of end-to-end duration. This concept is similar to the one defined in the temporal betweenness centrality.

As previously mentioned, betweenness centrality has shown its relevance to problems such as identifying important nodes that control flows of information between separate parts of a network and identifying casual nodes to influence other entities' behavior. It has been also used to analyze social and protein networks, to identify and analyze behavior of key bloggers in dynamic networks of blog posts, to identify significant nodes in wireless ad hoc networks, to study online expertise sharing communities, to study the importance and activity of nodes in mobile phone call networks and interaction patterns of players on massively multiplayer online games and to measure network traffic in communication networks. In relation to DTNs, many social routing protocols that use betweenness centrality have been proposed to enhance routing in PSNs, SONs, MSNs, Disconnected Delay-Tolerant MANETs and so on. With the exception of GrAnt and cGrAnt [VMDV12], the remaining surveyed routing protocols exploit social characteristics of mobile nodes in those networks. Additionally, betweenness centrality in DTNs has shown its relevance to problems such as the construction of a mobile backbone, the offloading of data in wireless social mobile networks, and information dissemination and content placement in opportunistic networks.

## 2.4 Security mechanisms in WANETs

This section reviews security threats and requirements due to the challenges posed by the decentralized and self-organized nature of WANETs. It reviews trust and incentive schemes that are used to stimulate cooperation among nodes in such networks. In addition, a cryptographic mechanism and some routing protocols that ensure privacy are presented and surveyed, respectively. Lastly, this section also surveys incentives schemes applied to DTN routing protocols.

### 2.4.1 Security threats

In WANETs, forwarding decisions are made individually by each node (or entity), which may incur in the consumption of nodes' limited resources, e.g., battery power, bandwidth, processing, and memory, all over the network. These networks share a tricky notion of being self-organized and self-managed, but they require for their correct operation that each node gives its own contribution.[10] Some concerns arise in these networks (1) in the establishment of trust between entities, or (2) to stimulate their cooperation, or even (3) due to the fairness of their contributions.

There are two possible types of nodes' misbehavior: selfish and malicious. When a node manifests a *selfish* behavior, it aims to maximize its benefits by using the network while saving its own resources (e.g., battery power). As such, cooperation enforcement schemes can be leveraged to foster cooperation. If a node manifests a *malicious* behavior, it tends to maximize the damage caused to the network. A way to deal with such misbehavior is by detecting and isolating those nodes from the network.

Selfish behaviors can be classified as individual or social selfishness [WLX14, ZXSW13]. A node presents individual selfishness if it only aims at maximizing its own utility, hence disregarding a system-wide criteria. Social selfishness is manifested when nodes only forward messages of others to whom they have social ties with.

Besides selfishness, entities on a self-organized environment are also prone to other forms of attacks such as flooding and cheating. A flooding attack consists in trying to exhaust the network's or the others' resources, e.g., by initiating an enormous amount of requests, in order to render the network useless. A cheating (or retention) attack consists in gaining an unfair advantage over other nodes by holding essential system's data.

Yet another way of classifying attacks is based on the nature of the attacks and the type of attackers. As such, they can be classified as passive or active [Sta07]. Passive attacks happen if an unauthorized party gains access to a message without modifying its contents. There are two types of passive attacks, namely: (1) the release of message contents [SOM10], which happens if message contents are made available or disclosed to unauthorized parties, and (2) traffic analysis [LVW09], which allows an attacker to infer communication patterns, thus guessing the nature of the communication that was taking place. Active attacks are characterized by an unauthorized party modifying the contents of the message.

Table 2.8 summarizes potential attacks to WANETs.

### 2.4.2 Security requirements

As a wireless medium is accessible (or open) for everyone within communication range, misbehaving nodes might attempt to compromise message contents, justifying the existence of security requirements [Sta07] such as authentication, confidentiality, integrity, availability and privacy, in such environments.

#### Authentication

Authentication assures that the communication is genuine. There are two possible cases: (1) the single message case, e.g., a warning message, where the objective is to assure the source's legitimacy to the destination of the message, i.e., that the message is from the source that it claims to be from; (2) the ongoing interaction case, e.g., the connection of an entity to another, where the objective is twofold: first, it is necessary to assure the entities' authenticity, i.e., that each entity is who it claims to be, and second, it is also necessary to assure that a third party cannot interfere with the connection, even if masquerading as one of the legitimate parties. In WANETs, authentication assures the authenticity of all the related processes such as sensing, communication and actuation.

[10] It is assumed that nodes have equivalent privileges and responsibilities.

| Attack pype | Attack Description |
|---|---|
| Individual selfishness [WLX14, ZXSW13] | A selfish node only aims at maximizing its own utility, disregarding the system-wide criteria. |
| Social selfishness [WLX14, ZXSW13] | Nodes only forward messages of others to whom they have social ties with. |
| Eavesdropping [Sta07, SOM10] | Message content is made available or disclosed to unauthorized parties. |
| Compromised-key attack [WYX$^+$10] | An attacker obtains a cryptographic key and uses it to read an encrypted communication. |
| Traffic analysis [Sta07, LVW09] | Extracting unauthorized information by analyzing communication patterns (but not their content). |
| Routing loop attacks [CSC11] | Modifying routing packets so they do not reach their destination. |
| Wormhole attacks [CSC11] | A set of malicious nodes creates a worm link to connect distant network points with low-latency, causing disruption in normal traffic load and end flow. |
| Black-hole attacks [CSC11] | A malicious node drops all packets forwarded to it and responds positively to incoming route requests despite the fact of not having proper routing information. |
| Gray-hole attacks [CSC11] | Special case of a black-hole attack where a malicious node selectively drops packets. |
| Denial-of-Service (DoS) [CSC11] | Attacks that obstruct the normal use or management of a service. |
| False information or false recommendation [CSC11] | Colluding and providing false recommendation/information in order to isolate good nodes while keeping bad ones connected. |
| Incomplete information | Consists in not cooperating to provide proper or complete information. |
| Packet modification/insertion | Consists in the modification or malicious insertion of packets. |
| Newcomer attacks [CSC11] | A malicious node registers as a new user to discard its bad reputation or distrust. |
| Sybil attacks [CSC11] | Consists in using multiple network identities. |
| Blackmailing [CSC11] | Consists in using a majority-voting scheme trying to cause routing topology change. |
| Replay attacks [CSC11] | Consists in maliciously or fraudulently repeating or delaying valid transmitted packets. |
| Selective misbehaving attacks [CSC11] | Consists in selectively misbehave to other nodes. |
| On-off attacks [CSC11] | Disrupting services by behaving correctly/incorrectly in alternation. |
| Conflicting behavior attacks [CSC11] | Behave differently to different nodes to cause contradictory opinions. |
| Credit forgery attack (or layer injection attack) [ZLL$^+$09] | Forge valid credit in order to reward itself for work it did not do or for more than it has done. |
| Nodular tontine attack (or layer removal attack) [ZLL$^+$09] | Remove one or more layers of a multilayer credit generated by previous forwarding nodes. |
| Submission refusal attack [ZLL$^+$09] | If the source and last intermediate node collude, the latter may refuse to submit the received credit from a virtual bank, and receive another compensation from the source node. |

Table 2.8: A summary list of potential attacks to WANETs

### Confidentiality

Confidentiality assures the protection of the transmitted data from passive attacks, since, and as previously stated, the wireless medium is accessible for everyone within communication range. Regarding the content of the data transmission, several levels of protection can be identified, in which the broadest level assures the protection of all user data transmitted between two nodes over a period of time [Sta07]. For instance, if a TCP connection between two systems is considered, the broadest service could prevent the release of any user data transmitted over the connection, meanwhile the narrower service could include the protection of a single message or a specific fields within a message. Confidentiality may also aim to assure protection against traffic analysis. The idea is that an attacker shall not be able to perceive traffic flow information such as the source and destination, frequency, length, or even other traffic's characteristics. In WANETs, confidentiality assures that sensitive information cannot be disclosed to unauthorized third parties during its propagation through the network.

#### Integrity

Integrity assures that the transmitted data is received as sent, without duplication, insertion, modification, reordering, or replay. Similarly to confidentiality, integrity can apply to: (1) a messages' stream, (2) a single message, or even (3) certain fields of the message. And as before, the stream of messages' protection is the best approach.

In WANETs, integrity assures that transmitted data cannot be modified while in transit through the network.

#### Availability

Availability refers to a system's property of being accessible and usable when requested by an authorized entity, in accordance to the system's design. Loss or reduction of availability can be caused by a variety of attacks. An example of such attacks is DoS (see Table 2.8). With automated countermeasures, e.g., authentication and encryption, or some sort of physical action, it is possible to prevent/recover from the loss of availability resulting from such attacks. In WANETs, availability assures that the network shall be available when a node needs to send a message.

#### Privacy

Privacy can be understood as the users' willingness to disclose or not his/her information, to others (family, friends, or even the general public). As a property, privacy is the confidentiality of personal information.

The lack of infrastructure in WANETs leverages nodes' forwarding decisions, allowing the opportunist exploitation of any other nodes in their vicinity to help messages reach their intended destinations. In such environments, message forwarding relies on the nodes' participation in the network.

If some nodes are deemed untrusted, privacy issues may rise due to passive attacks, which are in the nature of eavesdropping on transmissions.

In WANETs, privacy is more related to specific applications' requirements. For example, sensitive information of an entity controlling/owning a DTN node cannot be disclosed to other entities.

### 2.4.3 Privacy in DTNs

According to the literature [WYLC09], privacy breaches can be classified as identity disclosure, link disclosure and attribute disclosure. Identity disclosure is the case when the identity of the individual associated to the node is revealed. Link disclosure happens when the sensitive relationship between the individuals is disclosed. Attribute disclosure is the case when the sensitive data associated with the node, owned by an individual, is compromised. Moreover, there are several types of sensitive information such as node attributes, specific link relationships between nodes, nodes degrees, neighborhoods of some target nodes, etc.

Anonymization methods [WYLC09] can be used to protect privacy if sensitive information needs to be processed elsewhere. There are three main anonymization methods, namely: (*i*) $k$-anonymity privacy preservation via edge modification, that modifies graph structure by successive deletions and additions of edges so that each node in the modified graph is indistinguishable with at least $k - 1$ other nodes in terms of a given network property; (*ii*) edge randomization, that modifies the graph structure by randomly adding/deleting edges or by switching edges;

and (*iii*) cluster-based generalization, where nodes and edges are clustered into groups and anonymized into a super-node.

It is commonly assumed in WANETs that nodes are willing to share their private information for the sake of the network's performance. For instance, in DTNs, some routing protocols that address privacy issues in DTNs have been proposed [LHT$^+$10, SOM10, CMS12, RV12, PH12, SCRB16]. Routing approaches such as [LHT$^+$10, SOM10, CMS12] ensure attribute privacy. The location used by the source node to send messages is protected in [LHT$^+$10]. The context, e.g. personal information, residence, work, hobbies, interest profiles, etc., which is used for forwarding is protected in [SOM10, CMS12]. In [RV12], an adaptive mechanism for achieving user anonymity that ensures identity privacy is proposed. Identity privacy can be compromised if an attacker combines external knowledge with observed network structure [WYLC09]. In [PH12], an approach that ensures link privacy has been proposed where instead of transmitting the list of friends of the sender as a list of nodes, a modified and obfuscated one is transmitted.

Other privacy techniques have been proposed in the literature. For instance, with homomorphic encryption – proposed by Rivest et al. in 1978 [RAD78] – a node can carry out computations on encrypted values, without needing to decrypt them first. In [ZEPP13] and [SCRB16], privacy-preserving routing protocols based on additive homomorphic encryption (Paillier cryptosystem [Pai99]) were proposed. The former, which was proposed for peer-to-peer networks, allowed a node to calculate its similarity to other nodes using multivariate polynomial evaluation, meanwhile the latter, which was proposed as a secure geographical routing protocol for DTNs, allowed nodes to compare their habitats in order to choose the best forwarder for every message, respectively.

### 2.4.4 Homomorphic encryption

In cryptography, finding common elements in two private sets without exposing the sets themselves is known as the Private Set Intersection (PSI) problem [FNP04]. For instance, an algorithm that solves the PSI problem would allow a trusted node to send an encrypted version of some data to be processed by an untrusted node and the latter would perform computations on this encrypted data without knowing anything of the data's real value, and send back the result. The trusted node would expect the decrypted result to be equal to the intended computed value, as if it was performed on the original data. For example, with homomorphic encryption a node can carry out computations on encrypted values, without decrypting them first.

If addition operators are considered, the scheme is additively homomorphic. Likewise, if multiplication operators are considered, the scheme is multiplicatively homomorphic. An additive homomorphic encryption scheme is the one in which two numbers encrypted with the same key $\mathcal{E}(a)$ and $\mathcal{E}(b)$ can be added without being first decrypted, i.e., one can efficiently compute $\mathcal{E}(a+b)$ without decrypting them.

In the Paillier cryptosystem [ZGH07], which is an additive homomorphic encryption scheme, when entity $i$ wants to send message $m$ to entity $j$, entity $i$ selects random primes $p$ and $q$ and constructs $n = pq$; plaintext messages are elements of $\mathbb{Z}_n$ and cyphertext are elements of $\mathbb{Z}_{n^2}$. Entity $i$ picks a random $g \in \mathbb{Z}^*_{n^2}$ and verifies that $\exists \mu$ where $\mu = \left(L(g^\lambda \bmod n^2)\right)^{-1} \text{mod n}$ , $L\left(x\right) = (x-1)/n$ and $\lambda = \text{lcm}(p-1, q-1)$. If $\nexists \mu$ then a new random $g \in \mathbb{Z}^*_{n^2}$ must be picked. Entity $i$'s public key $p_k$ is $(n, g)$ and private key $s_k$ is $(\lambda, \mu)$.

To encrypt a message $m$, entity $j$ picks a random $r \in \mathbb{Z}^*_n$ and computes the cypher text $c\!= \mathcal{E}\left(m\right) = g^m \cdot r^n \bmod n^2$, therefore cyphering with $p_k$. To decrypt $c$, entity $i$ computes $\mathcal{D}\left(c\right) = \Big(L(c^\lambda \bmod n^2)\Big)^{-1} \cdot \mu \bmod n =$

$m$, therefore deciphering with $s_k$.

Let $\mathcal{E}\left(a\right) = g^a \cdot r_1^n \bmod n^2$ and $\mathcal{E}\left(b\right) = g^b \cdot r_2^n \bmod n^2$. Entity $j$ can compute the sum this way: $\mathcal{E}\left(a+b\right) = \mathcal{E}\left(a\right) \cdot \mathcal{E}\left(b\right) \bmod n^2 = g^{a+b} \cdot \left(r_1 r_2\right)^n \bmod n^2$.

Let $\mathcal{E}\left(a\right) = g^a \cdot r_1^n \bmod n^2$ and $k$ be a non-encrypted constant. Entity $j$ can compute the multiplication by a non-encrypted constant $\mathcal{E}\left(k \cdot a\right) = \mathcal{E}\left(a\right)^k \bmod n^2 = g^{k \cdot a} \cdot \left(r_1\right)^n \bmod n^2$.

### 2.4.5 Trust

According to the *Oxford Dictionary of English* [Ste10], trust is a "firm belief in the reliability, truth, or ability of someone or something". In the literature, many other definitions of trust can be found according to various disciplines. For example, in line with *social sciences* [Coo01], trust can be defined "as the degree of subjective belief about the behavior of a particular entity"; in line with economics – as shown in the Prisoner's Dilemma – trust is based on the assumption that humans are rational and strict utility maximizers of their own interest, therefore selfish. Even though, the emergence of altruistic behavior can be seen in initially purely selfish mechanisms. In line with *communication and networking* [CSC11], trust relationships among participating nodes are important, as with them cooperative and collaborative environments can be built, which improve system objectives in terms of scalability, re-configurability, reliability, dependability, or security.

Trust management is necessary in order for participating nodes without previous interactions to form a WANET with an acceptable level of trust relationships between them [CSC11]. As stated in [BFL96], "trust management provides a unified approach for specifying and interpreting security policies, credentials, and relationships." Some applications of trust management in decision-making situations are intrusion detection, authentication, access control, key management, and the isolation of misbehaving nodes for effective routing [CSC11].

*Trust management* is composed of trust establishment, trust update and trust revocation. A *trust establishment* process consists of the representation, evaluation, maintenance, and distribution of trust among nodes. A *trust update* process consists in modifying a node's trust values as a consequence of their collaboration with others, thus favoring trustworthy nodes and penalizing untrustworthy ones. *Trust revocation* consists in dropping/cancelling trust relationships among nodes.

In general, the main properties of trust in WANETs [CSC11] are: (1) *dynamicity*, since trust establishment is based on temporally and spatially local information; (2) *subjectivity*, due to network's dynamics, a trustee node may be assigned different levels of trust as a result of different experiences; (3) *incomplete transitivity*, since, for instance, Alice may trust Bob, and Bob may trust Charlie, but it is not guaranteed that Alice trusts Charlie. Trust's transitivity among two entities (trustor and trustee) and a third party is guaranteed if a trustor trusts the trustee and the trustee's recommendation of the third party; (4) *asymmetry*, entities with different capacities (e.g., energy or computation power) may not trust each other; (5) *context-dependency*, trust types depend on the foreseen task.

The properties presented above should be taken into account during the design of a trust-based framework. Other important aspects to consider for WANETs are: (*i*) an entity decision procedure of trust should be fully distributed and based on a cooperative evaluation with uncertainty and incomplete evidence, since a Trusted Third Party (TTP) may be unreliable in such environments; (*ii*) trust's determination should be flexible to membership changes and to deployment scenarios, therefore in a highly customizable manner; (*iii*) selfishness should be taken into account by a trust decision framework, hence no assumption that all nodes are cooperative should be made;

(*iv*) also because of network's dynamics, trust should be established in a self-organized and reconfigurable manner.

Additional care should be taken to ensure that a trust management system is not easily subverted, attacked or compromised. It is important to mention that trust management schemes are devised to detect misbehaving nodes (i.e., selfish nodes along with malicious ones). In addition, if the available information or evidence does not provide a certain level of trust, the trust engine should be robust enough to gracefully degrade.

### 2.4.6 Incentive schemes

Incentive schemes [SNR$^+$07] (also known as cooperation enforcement schemes/mechanisms) can be leveraged to manage and organize decentralized and self-managed systems, therefore compensating for the nonexistence of a central or dedicated entity. As a consequence, it is possible to deal with the security challenges previously mentioned. Even though, a cooperative behavior may result in an increase of the nodes' resource consumption, e.g., forwarding nodes may incur in additional energy and bandwidth usage during packets' transmissions and receptions. It was demonstrated in the context of MANETs that cooperation can succeed over competition [BH03]. So, the idea is to guarantee that a cooperative behavior is overall more beneficial than a passive or malicious uncooperative behavior. It is often desirable that these mechanisms distinguish uncooperative behaviors due to valid reasons such as energy shortage, crashing, etc., from malicious uncooperative ones.

The use of incentive schemes in MANETs has been exhaustively researched [JHB03, BBC$^+$01, BH00, SNCR03, AE03, WEWL06, WLW04, ZLLY07]. Incentive schemes have been envisaged for many application domains, namely infrastructure-based P2P applications (e.g., file sharing, distributed processing, and data backup), wireless networks (e.g., DTNs, WSNs, MANETs, wireless ad hoc backup, and nomadic computing) and Web commerce (e.g., auction sites, review and recommendation sites) [SNR$^+$07].

Incentive schemes can be categorized as reputation-based, remuneration-based and game-based schemes. The following sections discuss in detail each one of these categories.

#### Reputation-based schemes

There is a distinction between trust and reputation, despite their occasional interchangeable use in literature. Reputation is one entity's opinion about another. Reputation-based schemes are those in which the decision to interact depends on the other node's reputation.

**Architecture.** The architecture of the reputation management system, an integral part of the reputation mechanism, can be centralized, decentralized or hybrid [SNR$^+$07]. In the centralized approach, a central authority collects nodes' information, derives and provides the scores for all participants. In the distributed approach, since no central authority is available, nodes' ratings are stored in a distributed fashion. The evaluation of reputation is commonly based on subsets of information (e.g., neighbor nodes' information), which may be prone to inconsistencies if compared to the centralized approach. Nevertheless, a distributed management system scales better, and is more common in WANETs, than the centralized one. A hybrid approach is a combination of both.

**Operations.** Reputation-based mechanisms are composed of three phases: collection of evidence, cooperation decision and cooperation evaluation [SNR$^+$07]. In the first phase, a node collects reputation information by observing, experiencing and/or by means of recommendations of third parties. In the second phase, a node evaluates the collected information in order to decide if it should cooperate or not, based on the other node's reputation. Some evaluation methods used in this phase are: voting schemes, average ratings, Bayesian based computation, flow mode, etc [SNR$^+$07]. In the last phase, the degree of cooperation between nodes is evaluated. It is done after their interaction, and consists in rewarding nodes that presented a good behavior by adequately increasing their local reputation. Consequently, nodes with bad reputation are isolated, hence not receiving others' services.

**Attacks and counter-measures.** Misbehaving nodes can cause a variety of attacks such as individual and social selfishness, DoS, functionality attacks (e.g., subversion attacks), and single or group attacks to the reputation system (e.g., a liar or collusion[11], respectively). For example, the CORE [MM02] mechanism can be used to guard against the impact of liars. In [MGLB00], the Watchdog and Pathrater are used to identify misbehaving nodes and selecting paths to avoid them, respectively. For a more detailed discussion of attacks and countermeasures, please refer to [Sen10].

### Remuneration-based schemes

Remuneration-based schemes are those in which cooperating nodes should receive an equivalent complement (remuneration), and misconduct is punished with a penalty. The exchange of services, for some sort of payment, calls for a TTP (e.g., bank) to manage the process.

**Architecture.** A remuneration-based mechanism is composed of four operations: negotiation, cooperation decision, cooperation evaluation and remuneration [SNR$^+$07]. During negotiation, nodes decide on the terms of their interaction. This can be done only between them, or between them and a TTP. The cooperation decision, i.e., if a node can or not cooperate, is taken based on the outcome of the negotiation. The cooperation evaluation is twofold: on the one side, the service requesting party decides based on the acceptability of the service to the request; on the other side, the service providing party, decides based on the acceptability of the remuneration. At last, the collaborating node is remunerated. There are three types of remuneration envisaged in this scheme: virtual currency units, real money and bartering units [SNR$^+$07].

**Fair exchange.** A fair exchange protocol offers ways to guarantee that at the end of the exchange held by two or more nodes, either all of them have received what they were expecting or none of them has received anything. The correctness of the exchange depends on the availability of a neutral TTP. There are two types of protocols: online and offline fair exchange protocols [SNR$^+$07]. In the former, the TTP constitutes a bottleneck as it mediates every interaction between the nodes. In the latter, the TTP is used as an intermediary if and only if one of the nodes has doubts about the fairness of the exchange.

### Game-based schemes

Game-based schemes are those in which forwarding decisions are modeled using game theory. Game theory provides guidelines on how to model situations such as social dilemmas (e.g., the prisoner's dilemma). It also

[11] According to the Oxford Dictionary of English [Ste10], collusion is a secret or illegal cooperation or conspiracy in order to deceive others.

provides insights on how individual node-to-node interactions, without a centralized entity, can still spawn cooperation towards a more efficient outcome. Classical (rational) game theory assumes that nodes (i.e., players) have well-defined and consistent goals, that can be described by an utility function, which can be seen as a measure of the players' satisfaction resulting from a certain game outcome. So, in these schemes, each node follows a strategy aiming at maximizing its benefits and minimizing its resource consumption. For example, a node decides to forward a message if the (direct or indirect) result of that action maximizes its delivery probability, or possibly it minimizes its end-to-end delay. Below, some definitions are presented.

**Definition 2.4.1.** Game. A game constitutes a formal description of a strategic interaction between players.

**Definition 2.4.2.** Player. A player is an entity entitled to its own decisions and subsequent actions. It can also be interpreted as a node or as a group of nodes making decisions.

**Definition 2.4.3.** Action. An action is the act of performing a move in the game.

**Definition 2.4.4.** Payoff. A payoff (or utility), typically represented by a (positive or negative) number, reflects the desirability of an outcome to a player. As a consequence, it incorporates the player's attitude towards the risk.

**Definition 2.4.5.** Strategy. A strategy represents a set of actions that can be performed by the player during the game.

**Definition 2.4.6.** Payoff Matrix. A payoff matrix is a matrix that represents: the players, their strategies and the payoffs for each player with every possible combination of actions.

A plethora of ways to classify games is available nowadays [OR94, TvS12]. Some classification examples are: (1) according to the level of cooperation; (2) according to the symmetry of the payoff matrix (or according to the dependency between the strategy and the player); (3) according to the sum of the players' payoffs; (4) according to the amount of information known in advance; (5) according to how the plan of action is chosen; and (6) according to the number of players.

In relation to (1), games can be classified as cooperative or non-cooperative. In non-cooperative games, the game focuses on the strategy of the node where it has to make a decision if it is going to cooperate or not with another random node. If the cooperation decisions are taken by a group of nodes, these types of games are called cooperative games. In relation to (2), games can be classified as symmetric or asymmetric. Symmetric games (also called matrix games) are those in which the strategy options and payoffs do not depend on the players, but only of the other strategies employed. All players have the same strategy set [CRVW04]. The games where the strategies of the players are not identical are known as asymmetric ones (also called bi-matrix games). In relation to (3), games can be classified as zero sum or non-zero sum games. In zero-sum games, the sum of all players' payoffs is zero, for any possible outcome. Thus, a player's benefit is equal to the loss of other players. However, if for any outcome, the sum of all players' payoffs is greater or less than zero, they are called non-zero sum games. In relation to (4), games are classified as of perfect or imperfect information. If all previous players' moves are known, the game is of perfect information. A concept similar to a perfect information game is a complete information game, in which all players' strategies and payoffs are known, excluding their actions. In relation to (5), games can be classified as strategic or extensive. If each player chooses (once and for all) its action's plan, and all players' decisions are simultaneously made, it is called a strategic game. In extensive games, each player can choose its plan of action while taking decisions, which does not have to happen at the beginning of the game. In relation to (6), games can be classified as two-player or multi-player. As the name suggests, the two-player games are those played by exactly two players. Instead, if there are more than two players, the game is called a multi-player one.

**Social dilemmas** Classical game theory is based on the following key assumptions: (1) player's perfect rationality, i.e., players have well-defined payoff functions, being fully aware of their own and their opponents' strategy options and payoff values; and (2) that this is common knowledge, meaning that all players are aware of their own rationality, and of the rationality of other players; and that all players are aware that all players are aware that all are rational, etc., ad infinitum. A strategy profile of a game is said to be a Nash Equilibrium (NE), if and only if no player has an unilateral incentive to deviate and play another strategy, since there is no way it could be better off given the others' choices.

Social dilemmas occur in certain situations where the game has a single NE, which is not Pareto efficient[12] so that the sum of individual utilities (social welfare) is not maximized in equilibrium. One of game theory's main tasks is to provide guidelines on how to solve social dilemmas, and to provide insights on how player-to-player interactions (excluding the intervention of a central entity) may still generate an aggregate cooperation towards a more efficient outcome in many real-life situations.

The most popular dilemmas of cooperation are: the snowdrift game, the stag-hunt game, and the prisoner's dilemma [SF07].

**Stag-Hunt.** In *A Discovery in Inequality* of 1755, Rousseau described the Stag-Hunt game's story. In it, each hunter prefers stag S over hare H, and hare over nothing. According to game theory, the highest income is achieved if each hunter chooses hunting stag, but the chance of a successful stag hunt increases with the number of hunters. So, there is nearly no chance of catching a stag alone, but the odds of getting a hare does not depend on others. Hence, the payoff matrix for two hunters is given by:

| | | Hunter$_2$ | |
|---|---|---|---|
| | | H | S |
| Hunter$_1$ | H | (1,1) | (2,0) |
| | S | (0,2) | (3,3) |

**Prisoner's Dilemma.** In 1950, Tucker aiming at showing the difficulty of analyzing certain kinds of games previously studied by Dresher and Flood, came up with the Prisoner's Dilemma story: "Two burglars ($B_1$ and $B_2$) are arrested after their joint burglary and held separately by the police. However, the police does not have sufficient proof in order to have them convicted, therefore the prosecutor visits each of them and offers the same deal: if one confesses (called defection $D$ in the context of game theory) and the other remains silent (called cooperation $C$ – with the other prisoner), the silent accomplice receives a 3-year sentence and the confessor goes free. If both stay silent then the police can only give both burglars 1-year sentence for minor charges. If both confess, each burglar receives a 2-year sentence."

[12] Pareto efficient is a NE refinement concept used to provide equilibrium selection in cases where the NE concept alone could provide multiple solutions to the game.

Let $P$ – Punishment for mutual deflection, $T$ – Temptation to defect, $S$ – Sucker's payoff and $R$ – Reward for mutual cooperation, such that the matrix payoff is:

| | | $B_2$ | |
|---|---|---|---|
| | | D | C |
| $B_1$ | D | (P,P) | (T,S) |
| | C | (S,T) | (R,R) |

$S < P < R < T$ is the ranking ordering satisfied by the matrix elements.

**Snowdrift.** Two drivers trapped on opposite sides of a snowdrift have the following options: (1) cooperation, by getting out and shoveling; (2) defection, by remaining in their cars. If both drivers decide to shovel, each of them gets the benefit $b$ of getting home and both share the work's cost $c$. Therefore, each receives a mutual cooperation reward $R = b - c/2$. If they both choose to defect, none of them gets home and they both obtain no benefit ($P = 0$). If only one of them shovels, then both get home but the defector's income becomes $T = b$, as it was not reduced by shoveling, while the cooperator gets $S = b - c$.

Snowdrift is mathematically equivalent to the hawk-dove and the chicken games.

### Strategies

**Tit-for-Tat.** Rapoport proposed the tit-for-tat strategy for the Axelrod computer tournament in 1984. This strategy, for the Iterated Prisoner's Dilemma, starts by cooperating in the first step and subsequently repeating the opponent's previous action. In the long run, the tit-for-tat strategy cannot be exploited, since it retaliates defection (never being also the first to defect) by playing defection until the co-player decides on cooperating again. Thus, the extra income gained by the opponent during its first defection is returned to tit-for-tat. It is also a forgiving strategy as it is willing to cooperate again, defecting only if the opponent defects. This strategy effectively helps to maintain cooperative behaviors in multi-player evolutionary Prisoner's Dilemma games [SF07]. Tit-for-Tat modified versions were proposed to overcome the drawbacks of the strategy's determinism (e.g., in noise environments) such as the Tit-for-Two-Tats, which only defects if its opponents has defected twice in a row, and the Generous (Forgiving) Tit-for-Tat, which cooperates with some probability even if the co-player has previously defected.

**Win-Stay-Lose-Shift.** The concept of Win-Stay-Lose-Shift (WSLS) was introduced by Thorndike in 1911. WSLS strategies make use of a heuristic update rule depending on a direct payoff criterion (aspiration level) that dictates when the player should change its planned action. The player maintains its original action, if the recent rounds' average payoff is above the aspiration level, changing to a new one if not. By doing so, the aspiration level differentiates between winning and losing situations. A random choice can be performed, if there are multiple changing alternatives. An example of WSLS strategy is Pavlov [KK89], which was demonstrated to be able to defeat Tit-for-Tat in noisy environments (e.g., for the Iterated Prisoner's Dilemma), due to its ability to correct mistakes and exploit unconditional cooperators.

### 2.4.7 Incentive schemes applied to DTNs

In this subsection, somes incentive schemes to handle selfishness in DTNs are reviewed.

#### Reputation-based schemes

**RCAR.** The authors of [DD12] proposed a reputation-based extension to the Context Aware Routing (CAR) protocol [MM09], called RCAR, to address the problem of black-holes in DTNs.

In RCAR, every node keeps a local notion of reputation, therefore avoiding the overhead and technical complication associated with a centralized reputation management system in WANETs. Upon message forwarding, each node estimates the likelihood of selecting forwarding nodes based on the node's reputation, that is, based on past interactions with possible forwarding nodes.

The reputation management system employs both data and acknowledgment messages. The data messages' format incorporates the list of nodes ($nlist$), i.e., the list of nodes a message has passed through, as well as a list of digital signatures ($slist$) that is used to prove the integrity and authenticity of every node in $nlist$. The update mechanism, employed by the proposed reputation management system, does not disseminate updates based on broadcast/multicast mechanisms, which are expensive.

Reputation is maintained by means of three mechanisms: acknowledgments, list of nodes and aging. Since each message contains a list of forwarding nodes the message has traversed, upon message reception, nodes should update the reputation of the forwarding nodes in $nlist$. Despite that, the sender waits for an acknowledgment from the destination node, and only increases the reputation of the forwarding node upon reception of the acknowledgment. At last, the aging mechanism is used to decrease the reputation of all nodes. As messages can get lost, there is no way a node can know the reasons behind it, e.g., message drop due to a black-hole node (which node misbehaved?), or due to buffer overflow, or even due to Time-To-Live (TTL) expiration. To fulfill DTN requirements, the aging mechanisms decrease period is dynamically updated using Kalman filters [Kal60].

**Give2Get.** In [MS12], the authors proposed two strategy proof forwarding protocols for Pocket Switched Networks (PSNs) [HCS$^+$05] of selfish individuals, namely, Give2Get Epidemic and Give2Get Delegation. The proposed protocols are strategy proof, which means that the strategies of following the protocol are Nash Equilibria. So, no individual has any incentive to deviate.

In Epidemic Forwarding [VB00], nodes use every contact opportunity to forward messages. When a node is in contact with another one, and it has a message that the other does not have, the message is relayed to the other node. If selfish nodes that simply drop messages are considered in the network (also called message droppers), epidemic forwarding performance degrades [MPC13b].

In Delegation Forwarding [ECCD08], nodes have associated to them a forwarding quality, which may depend on the message's destination. A message, upon its generation, is associated with the forwarding quality of the sender. Then, when a relay node gets in contact with another node, it checks whether the forwarding quality of the other relay node is higher than that of the forwarding quality of the message. If so, it creates a replica of the message, labels both messages with the forwarding quality of the other relay node, keeps one of them and forwards the other message to the other relay node. If not, the message is not forwarded.

The idea behind Give2Get (G2G) Epidemic Forwarding is that the protocol works correctly even if all nodes in the network are selfish. That can be accomplished if no selfish node has a better choice than following the protocol truthfully, thus being a NE. G2G Epidemic Forwarding consists of three phases: message generation, relay and test. In the message generation phase, the message is modified, that is, the message sender is hidden from every possible relay except the destination, so that the relay candidate has no interest in not accepting it[13]. In the relay phase, nodes collect proof of relay (POR) to show to the source and previous relays during the test phase. In the test phase, nodes present evidence of their correct behavior, making it impossible for relays to drop messages. If no evidence is presented by the relay, a proof of misbehavior (PoM) is generated and the relay is removed from the network.

G2G Delegation Forwarding makes use of G2G Epidemic Forwarding techniques with the intention of stopping message droppers. Since G2G Epidemic Forwarding techniques are not enough for the algorithms to be NE, due to the fact of selfish nodes (1) being able to lie about the forwarding quality (i.e., being liars), and (2) being able to change the forwarding quality of messages, e.g., set it to zero, therefore getting rid of the message sooner (i.e., being cheaters). Two approaches have been devised for G2G Delegation Forwarding: Delegation Destination Frequency (DDF) and Delegation Destination Last Contact (DDLC). In the former, a node forwards a message to another node if the other node has contacted the destination of the message more frequently than any other node in the list of nodes inside the message. In the latter, a node forwards a message to another node if the other node has contacted the destination of the message more recently than any other node in the list of nodes inside the message.

**Other approaches.** In [WCZ11], a user-centric reputation-based incentive protocol for DTNs that allows each node to manage its reputation evidence was proposed. In it, the next hop nodes can demonstrate each successful transmission, where each relay node keeps its relay evidence to be later on validated by the source.

In [LD13, WZCS13], Bayesian approaches leveraging on the Dempster-Shafer Belief Theory [Sha76] were proposed for DTNs. In the former Bayesian aproach, a trust-based framework that can be integrated with single-copy data forwarding protocols was proposed. It uses a watchdog component and a special message to monitor the forwarding behavior of nodes. In the latter Bayesian aproach, a user-centric and social-aware reputation based incentive scheme was proposed. Similarly to [WCZ11], each node also manages its reputation evidence and demonstrates it whenever necessary. Two concepts have been introduced, namely self-check and community-check. They were defined for reputation evaluation in relation to the forwarding competency of the candidate and the sufficiency of the evidence that the node presents, and for speeding up reputation establishment and forming consensus views towards targets in the same community.

The authors of [BH11] proposed a reputation mechanism for opportunistic networks that uses social-network information to detect and penalize misbehaving nodes therefore stimulating them to participate in the network. This approach is limited to social-based DTNs. In [HOSC$^+$12], a collaborative watchdog mechanism aiming at reducing the detection time of selfish nodes by propagating second-hand information was proposed for MANETs and DTNs.

In [AF12], an iterative trust management and distributed malicious node detection mechanism for DTNs was proposed. It uses an iterative trust and reputation mechanism that enables each node to evaluate others based on

[13] The authors made the following assumptions: (1) every node is selfish and accepts messages destined to it; (2) there are no Byzantine nodes in the network; (3) selfish nodes do not collude; (4) nodes are capable of making use of public key cryptography.

their past behavior. The authors of [ZDG$^+$14] proposed a probabilistic misbehavior detection scheme for secure DTN routing. A trusted authority, which is periodically available, judges nodes' behavior based on the collected routing evidence and performs probabilistic checks. A trusted third party to judge and punish nodes based on their behavior is required.

In [YMH$^+$16, LZQ16], social trust routing schemes were proposed. In [YMH$^+$16], the trust routing based on social similarity scheme is built on the observation that nodes move around and contact each other according to their common interests or social similarities. In [LZQ16], the social trust model exploits the contact status, forwarding ability and common attributes. In addition, a trust based routing algorithm and buffer management algorithm for the secure routing strategy that considers network coding in data dissemination was also proposed.

The authors of [DRXM15] proposed a cooperative watchdog system (CWS) to support selfish nodes detection. Each time a node participates in a contact opportunity, a reputation score is assigned to him. CWS updates each nodes reputation score by means of the classification, evaluation of the neighbor and decision modules. The proposed classification module does not learn as new observations are available.

#### Remuneration-based schemes

**SMART.** The authors of [ZLL$^+$09] proposed a secure multilayer credit-based incentive (SMART) scheme to stimulate bundle forwarding cooperation among selfish nodes, which can be implemented in a distributed manner without relying on any tamperproof hardware in a DTN environment. In SMART, intermediate nodes, without the involvement of the sender, can transfer/distribute credit since a forwarding path cannot be predicted by the sender (unlike in MANETs where forwarding paths are known a priori), and due to nodes' mobility, intermediate nodes and the sender may be disconnected.

SMART assumes the existence of two main entities: an Offline Security Manager (OSM), which is responsible for key distribution, and a virtual bank (VB), i.e., a special network component, such as a roadside unit in vehicular networks [LPP$^+$07] or the information publisher in social networks [GA08], which takes charge of credit clearance. DTN nodes submit collected coins to the VB, by exploiting opportunistic links to these special network components. Upon joining the network, every node should register with the OSM. During the clearance phase, nodes should submit the collected layered coins to the VB in order to receive their rewards.

SMART is based on the concept of a layered coin – composed of multiple layers, where each layer is generated by the source/destination or an intermediate node – providing virtual electronic credits to charge for and reward the provision of data forwarding in DTN environments. The first layer, also known as the base layer, is created by the source node, and it indicates: the payment rate (credit value), the remuneration conditions, and the class of service (CoS) requirements, besides other reward policies. In the following propagation process, a new layer, also known as the endorsed layer, which is built on the previous one, is created by each intermediate node by appending a non-forgeable digital signature. It implies that the forwarding node agrees in providing forwarding services underneath the predefined CoS requirement, thus being accordingly remunerated in accordance to the reward policy. By checking the signature at each endorsed layer, it is easy to check the propagation path and determine each intermediate node. If the provided forwarding service fulfills remuneration conditions defined in the predefined reward policy, in the rewarding and charging phase, each forwarding node along a single/multiple path(s) will share the credit defined in this coin depending on single/multi-copy data-forwarding algorithms and its forwarding results.

Four aspects were considered in SMART's design, namely: effectiveness, by stimulating cooperation among selfish nodes; security, by being robust to various attacks; efficiency, by not introducing extra communication and transmission overhead; and generality, by being compatible with most existing DTN routing schemes. A trade-off had to be taken into account between security and performance. Security, as intermediate nodes manage all security issues related to a coin, during the forwarding process, a(n) (individual or social) selfish node(s) may attempt to maximize its expected benefit by cheating the system. The final aspect is performance, due to the extra computation and transmission overhead as a result of any security functionality, during the design of a secure credit-based incentive scheme.

**Mobicent.** In [CC10], the authors proposed a credit-based system to support Internet access service for heterogeneous wireless network environments. Two modes of operation are supported by mobile devices in such environments, namely: (1) a long-range low-bandwidth link, e.g., cellular interface, to maintain an always-on connection, used in particular by the source and destination nodes; (2) a short-range high-bandwidth link, e.g., Wi-Fi, to opportunistically exchange large amounts of data with neighboring nodes (which is used by all nodes), since due to node mobility these links tend to be intermittent.

The Mobicent network architecture consists of three components: (*i*) a TTP, that stores key information for all nodes, providing also verification and remuneration services; (*ii*) Helpers, which are mobile or static nodes which help in data relaying using the short-range high-bandwidth link; (*iii*) Mobile clients, that are the destination nodes.

The payment mechanism works as follows: the data payload is encrypted by each relay with a one-time symmetric key before being forwarded. When a client receives the encrypted data and intends to access the decrypted one, it must make a payment to the TTP in exchange for the encrypted keys. This happens since the key is sent along with the data in encrypted form, and the TTP is the only means to recover them. Only relays involved in data forwarding receive payment.

In [CC10], the authors dealt with two forms of selfish actions (or attacks), namely: edge insertion attacks and edge hiding attacks. Let $G = (V, E)$ be a contact graph. Each vertex (or node) $v \in V$ can be identified by an integer value $i = 1, 2, ..., |V|$. Each edge $e \in E$, identified by a pair $\{v_1, v_2\}$, denotes the opportunistic contact between two nodes at time $t(e)$. Thus, $(\{v_1, v_2\}, t_1)$ means that $v_1$ meets $v_2$ at time $t_1$. The former, i.e., an edge insertion attack of a node $v$ consists in creating a Sybil $v^{'}$ so that $G \longrightarrow G^{'} = (V^{'}, E^{'})$, where $V^{'} = V \cup \{v^{'}\}$ and $E^{'} = E^{v \longrightarrow (v, v^{'})} \cup \{(v, v^{'}, t)\}$. The latter, i.e., an edge hiding attack for a node $v$ consists in modifying $G \longrightarrow G^{'} = (V, E - e)$, where $e \in E(v)$.

According to the authors, due to network's dynamics, edge insertion and hiding attacks are extremely difficult to be detected in DTNs. As the proposed scheme provides incentives for selfish nodes to honestly behave (that is, by setting the client's payments and relay's rewards so that nodes behave truthfully), mechanisms to detect selfish actions are not required. And also, by working on top of the DTN routing layer, this scheme ensures that selfish actions do not result in a large reward, not requiring pre-determined routing paths either.

**Other approaches.** In [JMT16], a credit-based congestion-aware incentive scheme for DTNs with a check and punishment mechanism to prevent forwarding nodes from deliberately discarding message, was proposed. It is composed of a message acceptance selection mechanism that allows nodes to decide whether to accept others' messages in accordance to their self congestion degree.

The authors of [CLWW16] proposed two credit-based rewarding schemes, namely earliest path singular rewarding and earliest path cumulative rewarding, to guarantee systematic security in DTNs. The aim of the former mechanism is to motivate nodes to truthfully forward the messages during every contact opportunity, meanwhile the latter rewards each node in the earliest delivery path in a cumulative way.

In [ZC12], a secure and reliable packet forwarding protocol using a threshold credit-based incentive mechanism was proposed. It is based on the modified model of population dynamics to efficiently prevent node compromise attacks, stimulate the cooperation among intermediate nodes, maximize vehicular nodes' interests and realize the fairness of possessing the same opportunity to forward packets for credits.

The authors of [SKW12] proposed a collusion–resistant incentive–compatible routing and forwarding scheme, where each node has to pay a credit to receive a data packet and later have to obtain credit for forwarding that packet. In addition, nodes must pay credit to send new data packets of their own.

In [SKW12], selfish nodes who fail to forward packets will not be able to send anything, and malicious nodes are prevented from flooding the network. Moreover, the mechanism also greatly limits black-hole attacks since it is necessary to pay for receiving the data.

The authors of [RC15] proposed a credit-based mechanism that enables device-to-device data exchange without the support of traditional Internet service providers. The monetary value of a given data message during its journey in the network is represented using a utility function, and selfish behavior among nodes is prevented using a buffer management optimization algorithm.

In [NLX$^+$16], a copy adjustable incentive scheme, which adopts a virtual credit concept to stimulate selfish nodes to cooperate in data forwarding, was proposed. Two types of credit were considered, namely social credit and non-social credit, to reward nodes when they relay data for other nodes inside or outside their community, respectively. Additionally, the number of messages a node can replicate to other nodes is adjusted according to its cooperation level and earned credit.

### Game-based schemes

**Tit-for-Tat.** In [SZ08], the authors propose a pair-wise tit-for-tat (TFT) – a simple, robust and practical cooperation enforcement scheme for DTNs – which incorporates generosity and contrition to tackle bootstrapping or exploitation problems common in basic TFT mechanisms [SNCR03, JS07, MJS06].

Upon the first encounter between two nodes, as they have not previously relayed packets successfully among them, the basic TFT mechanisms would prevent relaying. Generosity enables bootstrapping by allowing an initial cooperation between the nodes up to $\epsilon$, that is, a node is allowed to send $\epsilon$ packets more than it should according to what it had previously relayed. It also handles asymmetric traffic demands by absorbing traffic imbalance up to a $\epsilon$ amount. But any imbalance exceeding $\epsilon$ could lead to lengthy retaliation among neighbors. As a result, generosity is insufficient by itself. Contrition addresses the previous situation by refraining from reacting to a valid retaliation to its own mistake, i.e., preventing mistakes from causing endless retaliation. With contrition, a node realizes that the other node's action in the current interval was due to its own action in the previous interval, and so does not lower service in the future interval. Similarly, contrition cannot work by itself, since it only provides a way to return to stability after perturbation, not providing a way to reach stability.

The authors also proposed an incentive-aware routing protocol in which selfish nodes are allowed to maximize their individual utilities taking into account the TFT constraints. For a given pair of nodes, the TFT constraints state that the total amount of traffic through a link is equal to the total amount of traffic in the opposite direction.

The proposed routing protocol consists of the following components: (*i*) link state (i.e., link capacity, mean and variance of the waiting time on links) that are periodically exchanged by every node, similarly to many link state protocols (e.g., OSPF)[14]; (*ii*) with link state, each source node computes forwarding paths, and uses source routing to send traffic; (*iii*) each destination node sends an ACK via flooding, upon receiving a packet. Then, the source node uses it to update its TFT constraints for the subsequent interval.

**Barter.** The authors of [BDFV10] proposed a mechanism, which is modeled using game-theory, to discourage selfish behavior based on the principles of barter[15]. In the context of the proposed mechanism, exchanges are made in messages. Hence, when two mobile nodes are in communication range with each other, (1) they send messages' descriptions that they currently store to each other, (2) and then they reach a consensus on the subset of messages they want to exchange. Fairness is ensured (*a*) by guaranteeing that the selected subsets have the same size, and (*b*) by using a preference order, in which messages are exchanged in a message-by-message manner. Notice that the number of exchanged messages depends on the length of the shorter list or the duration of the connection. The exchange can be interrupted, if any party cheats. So, any major disadvantage is not experienced by the honest party, since it at most downloaded one less message in comparison to the misbehaving party.

In this scheme, the mobile nodes decide which messages they want to download from each other. If nodes behave selfishly, that is, they only download messages that are of primary interest (or destined) to them, in the long run, and according to the principle of barter, they will not have other messages to exchange for the ones they are interested in. Therefore, messages that are secondary (or not destined) for a given mobile node may still have a barter value for the mobile node. Thus, it can be perceived as an investment to acquire new primary messages.

Two assumptions were made by the authors: (*i*) mobile nodes offer all their valid and only valid messages to download, and (*ii*) two mechanisms are present in the system to prevent the injection of fake messages, specifically: digital signatures, where only nodes with a digital signature, supplied by an authority, can exchange messages among themselves, and a reputation mechanism, that is based on the quality of the message contents.

**Other approaches.** In [ZWZ$^+$15], an incentive-driven and freshness-aware publish/subscribe content dissemination scheme for selfish Opportunistic Networks, where TFT is used as the incentive scheme to deal with selfish behaviors of nodes, was proposed. When nodes are in contact, the exchange order is decided by the content utility that includes the direct and indirect subscribed values, and the objective of the nodes in the network is to maximize the utility of the content inventory stored in their buffer.

In [HHBL16], a Stackelberg game-based scheme for data dissemination in mobile opportunistic networks, which realizes the optimal resource allocation including the task assignment and pricing of data forwarding for the relays, was proposed.

[14] It is assumed in [SZ08] that link state is disseminated faithfully. Thus, the authors focused on making the data-plane incentive compatible. Security of the control plane was left for future work.

[15] According to the Oxford Dictionary of English [Ste10], barter means "exchange (goods or services) for other goods or services without using money

In [WCZ$^+$13], the message exchange process between a pair of nodes in probabilistic routing is modelled as a bargaining[16] game. Each message goes through a series of bargaining games while being transferred from its source to its destination. At each game, a message is traded from its current carrier to a node with even higher delivery probability.

The authors of [CSY15] proposed a game theoretical incentive scheme to encourage nodes to participate in packet forwarding/storage and to follow a certain packet routing strategy to realize a routing performance objective. It assigns credits for packet forwarding/storage proportionally to the priorities specified in the routing strategy and supports adjustable QoS for packet routing between specific sources and destinations.

In [MPS15], a distributed information-based cooperation ushering scheme, leveraging on evolutionary theory [Fre77b], to promote cooperation in message forwarding between nodes, was proposed. Upon an encounter, each node maintains and exchanges information with other nodes about the messages created and delivered in the network. Consequently, they evaluate their own performance and compare that with the approximated network performance to adapt to the most successful forwarding strategy.

The authors of [XSG16] proposed a game-based approach, where a virtual currency is used as payment for relay service, to stimulate selfish nodes to participate in bundle delivery in MSNs. Each bundle carrier selects relay nodes taking into account the current status of its limited resources. A bargain game is employed to model the transaction pricing for a relay service.

Table 2.9 presents a summary and comparison of incentive schemes applied to DTN Routing protocols based on the security mechanism used, the main idea of the routing protocol, the attacks considered, and the overhead.

[16] According to [Ste10], to bargain is “to negotiate the terms of an agreement, as to sell or exchange”.

[18] Despite not being the target attack addressed in [MS12], the authors proposed two mechanisms, namely random checks of conformity and rewarding traitors, to mitigate the presence of colluding nodes or to limit their possible harm.

| Scheme | Security mechanism | Main idea | Attack(s) considered | Overhead |
|---|---|---|---|---|
| RCAR [DD12] | Reputation-based | Every node keeps locally the reputation of every forwarding node it comes in contact with. Nodes with the highest reputation are selected as message forwarders. | Black-hole attack | The overhead is reduced, since RCAR reputation mechanism is integrated in the routing protocol (data and ACK messages). Thus, broadcast/multicast is not used by RCAR. |
| Give2Get [MS12] | Reputation-based | Every node keeps locally evidences of their correct behavior as a relay. If no evidences are presented in future encounters, nodes are removed from the network. | Individual selfishness (message droppers, liars and cheaters)<br>Social selfishness (collusion[18]) | Storage and communication overhead due to the use of PORs. The probabilities of detection of selfish behavior are of more than 90% and 60% for G2G Epidemic and Delegation forwarding, respectively. |
| SMART [ZLL$^+$09] | Remuneration-based | Secure multilayer credit-based incentive scheme to stimulate bundle forwarding cooperation among selfish nodes in a DTN environment. | Credit forgery attack<br>Nodular tontine attack<br>Submission refusal attack | Additional cryptographic (computational and communication) overhead due to the use of layered coins. The performance of the underlying routing protocol, with or without SMART, are very close. |
| Mobicent [CC10] | Remuneration-based | Runs on top of DTN routing layer and provides incentives for selfish nodes to behave honestly, thus not requiring a selfish actions' detection mechanisms. | Edge insertion attack<br>Edge hiding attack | No overhead analysis mentioned in the paper. |
| Tit-For-Tat [SZ08] | Game-based | Simple, robust and practical incentive mechanism for DTNs which incorporates generosity and contrition. | Individual selfishness | Additional control overhead caused by the dissemination of link state and ACK messages. |
| Barter [BDFV10] | Game-based | Incentive mechanism to discourage selfish behavior based on the principles of barter. | Individual selfishness | No overhead analysis mentioned in the paper. |

Table 2.9: A summary and comparison of incentive mechanisms applied to DTNs

# Chapter 3

# High throughput low coupling multipath routing for WMSNs

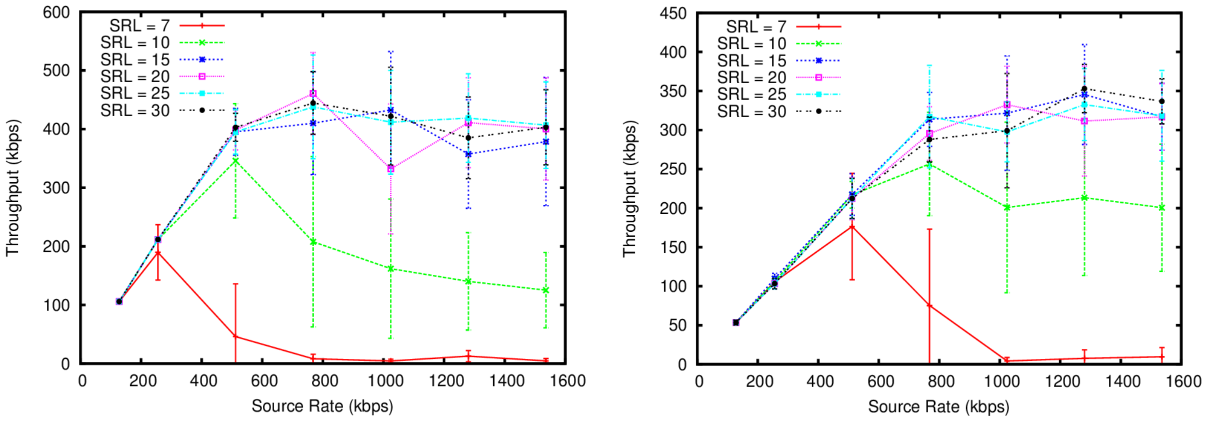
This chapter presents the design, implementation and evaluation of the High Throughput Low Coupling Multipath extension to the Dynamic Source Routing (HTLC-MeDSR) protocol. HTLC-MeDSR proposes a cross layer approach to the FRF problem involving the Medium Access Control (MAC) and the routing layers. At the routing layer, several enhancements to the Multipath extension to the Dynamic Source Routing (MeDSR) protocol [MPG11] are proposed (1) by allowing it to use the ETX metric, and (2) by including second degree neighbors (neighbor of the neighbor) information in the Correlation Factor (CF) calculation. The primary goal of the ETX design is to find paths with high throughput, despite some losses [DCABM05]. The CF allows finding paths with low route coupling if they exist. A scheme to distinguish collisions from route failures is used. By increasing the maximum number of MAC retransmissions, packet loss was reduced and probe packets were used to identify routing failures. Therefore, this scheme reduces unnecessary route maintenance operations.

## 3.1 Short retry limit in multipath routing protocols

The IEEE 802.11 DCF (Distributed Coordination Function) employs the Request to Send / Clear to Send (RTS/CTS) mechanism to reduce the effect of collisions due to the hidden terminal problem. It is known that the RTS/CTS exchange is useful in heavily contending environments, where many transmissions might fail due to collisions [Bia00].

If a node fails to send a RTS packet a certain number of times, called Short Retry Limit (SRL), it discards the corresponding packet and considers that the destination node is no longer available. Then, it triggers the route maintenance procedure. In the current IEEE 802.11 standard, the SRL is statically set to seven tries.

According to [KLY08], the problem of static SRL arises from the fact that a node cannot distinguish between collision losses and mobility-induced errors. Consequently, MANETs suffer from unnecessary overhead from routing maintenance. In Figure 3.1, the throughput achieved by MeDSR with different SRL values is analyzed to examine the effect of the routing instability because of the FRF problem. The simulation model (see Section 3.5 for more details) consisted of one simulation run performed over 5 scenarios with 50 nodes randomly distributed



(a) CBR traffic

(b) Pareto traffic

Figure 3.1: MeDSR throughput analysis in various static scenarios

over an area of 500m x 300m using 6Mbps links and with SRL values of 7, 10, 15, 20, 25 and 30. The source rate was varied from 128 to 1536 Kbps, for Pareto On/Off (50% average On time) and CBR (Constant Bit Rate) traffic.

For the standard value (i.e., SRL = 7), the throughput tends to zero for source rates above 768Kbps as the number of FRF is so high that almost no user traffic goes through. Figure 3.1 also shows that the throughput increases until SRL = 25 for CBR traffic (see Figure 3.1(a)) and SRL = 15 for Pareto traffic (see Figure 3.1(b)). From this results, one can conclude that a large SRL can improve throughput in static routing topologies. An optimal value for SRL for both CBR and Pareto traffic is 15. For Pareto, the maximum throughput was achieved, and for CBR, it was the third best value. No higher throughput gains were obtained for SRL values above 30.

## 3.2 The ETX metric

The ETX metric was proposed with the objective of supporting the selection of paths with high end-to-end throughput. The ETX of a link is considered as the average number of data transmissions (including retransmissions) required to send a packet over that link. For a path, the sum of the ETX for each link in the path is considered (the accumulated ETX).

The ETX of a link is calculated using the Link Quality (*LQ*) and the Neighbor Link Quality (*NLQ*) (see Figure 3.2). Taking node $S$ as reference, *LQ* is defined as the measured probability for a successful data packet transmission from node $A$ to node $S$. Therefore, *LQ* says how good a given link from a neighbor to the reference node is. In Figure 3.2, the source node $S$ calculates *LQ* from node $A$ to itself ($LQ_{SA}$) and for the path from node $S$ to destination $D$, there are 3 values of *LQ*: $LQ_{SA}$, $LQ_{AB}$ and $LQ_{BD}$. It is also important to know the quality of the link on the opposite direction, that is, how many of the data packets sent from node $S$ were received by node $A$. Now, the interest is in the corresponding neighbor's perception of *LQ*, known as *NLQ*. *NLQ* says how good a given link between the reference node to the neighbor is. For example, in Figure 3.2, node $D$ calculates *NLQ* from node $B$ to itself ($NLQ_{BD}$) and for the path from nodes $S$ to $D$ there are 3 values of *NLQ*: $NLQ_{SA}$, $NLQ_{AB}$ and $NLQ_{BD}$. The expected probability for a successful data packet round trip, (i.e. the probability that a node successfully sends a data packet to its neighbor and, on receiving it, its neighbor successfully replies with a response data packet) is $LQ \times NLQ$. Because each attempt to transmit a packet can be considered a Bernoulli trial, the expected number of

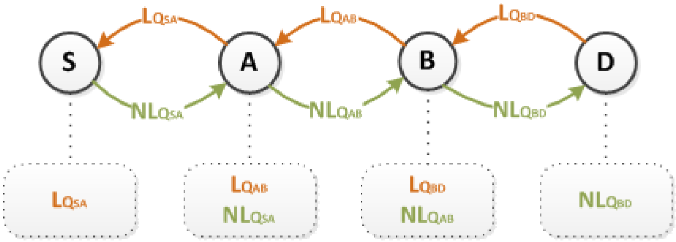


Figure 3.2: The ETX concept

transmissions is given by

$$\text{ETX} = \frac{1}{\mathit{LQ} \times \mathit{NLQ}} \tag{3.1}$$

In HTLC-MeDSR, dedicated link probe packets (known as HELLO packets) are used to measure *LQ* and *NLQ* of every link. Each node broadcasts these packets at an average period $\tau$ (e.g., one second), and they are jittered by up to $\pm 0.1\tau$ to reduce the probability of collisions (accidental synchronization). Every node remembers the probes it received during the last $\omega$ seconds (e.g., five seconds), which allows it to calculate *LQ*. Because probes are broadcast, 802.11a does not acknowledge or retransmit them as in [DCABM05]. To calculate the ETX of a link to a neighbor, a node needs also to know the *NLQ* as it can only determine the *LQ* by itself. In Figure 3.2, node A puts the *LQ* values that it has calculated in the probe packets it broadcasts, which from the reference node $S$ are *NLQ* values. By doing so, every node has all the information to calculate the ETX for each link between itself and all of its neighbors. A probe packet can include direct neighbors' information, *LQ* and ETX information (if they exist). The ETX of a path is the sum of the link's metric along the path, i.e., $\text{ETX}_{SA} + \text{ETX}_{AB} + \text{ETX}_{BD}$.

## 3.3 The correlation factor

The correlation factor (CF) between two paths is defined as the number of shared neighbor nodes between the two, excluding the source and destination nodes' neighbors. If there are no shared neighbor nodes between two paths, it is assumed that the two paths are unrelated (CF = 0). Otherwise, they have a certain value of CF.

Consider, for example, Figure 3.3 where there are two node disjoint paths from the source node 0 to the destination node 11. Two primary paths can be used to forward packets. Nodes on the first primary path are 0, 1, 3, 5, 7, 9 and 11 and on the other 0, 2, 4, 6, 8, 10 and 11. The neighborhood information for the first path is $\{0, 1, 3, 5, 7, 9, 11\}$ and for the other path is $\{0, 2, 4, 6, 8, 10, 11\}$. Excluding the source and destination nodes, these paths do not share nodes. So, the correlation factor for this case is CF = 0 because these two paths are unrelated.

Now, consider Figure 3.4 where there are 100 nodes randomly distributed over an area of 550x550m. It is assumed that each node has information of its first degree neighbors (direct neighbors) and of its second degree neighbors (neighbors of its neighbors) on its neighbors cache. This information can be collected using the HELLO packets, as described in section 3.2. Let nodes 5 and 40 be the source and destination nodes, respectively. By initiating a MeDSR route discovery process, the received RREQ packets contain information about the first and

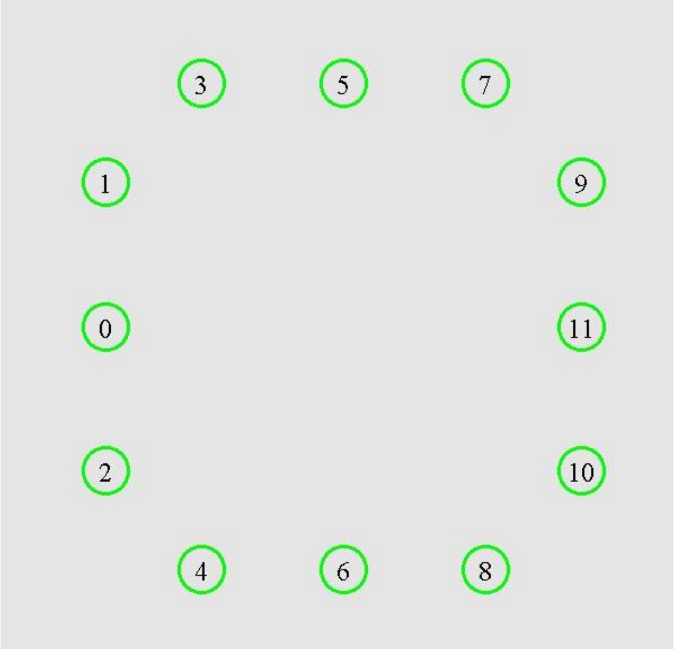


Figure 3.3: The correlation factor concept for a 12 nodes scenario

second degree neighbors of each node on each discovered path. The Dijkstra's Algorithm [CLRS09] is performed at the destination node to find node disjoint paths from node 40 to node 5, and the following paths are found: {40 17 1 3 5}, {40 33 44 7 5}, {40 43 60 19 5}, {40 9 15 23 5}, {40 30 61 28 5}, {40 45 32 35 5}, {40 48 16 37 5}, {40 70 80 58 5}, {40 62 97 85 5}, {40 65 22 27 8 5}, {40 71 54 2 31 5}, {40 89 75 86 41 5}, {40 96 77 76 68 5}, {40 82 0 6 26 4 5}, {40 98 14 79 42 25 5}, {40 39 47 73 50 90 84 5}.

Despite all paths found being node disjoint, if the selected pair of paths to use for multipath forwarding is not carefully chosen, the interference between paths can greatly degrade the performance of multipath routing protocols. The best option would be to select first the unrelated pairs of paths, but they were not found. The combinations of paths with the smallest values of CF are combination$_1$ = {{40 62 97 85 5}, {40 71 54 2 31 5}} and combination$_2$ = {{40 71 54 2 31 5}, {40 98 14 79 42 25 5}} with CF = 18 and CF = 9, respectively. The best option is combination$_2$ because it has the lowest value of CF. Having the best value of CF means that the paths of combination$_2$ have fewer common neighbors and less interference between them as can be seen in Figure 3.4.

There are cases where the combinations of paths have the same value of CF. The same principle as before still holds, but the first degree neighbors of all the nodes in the path are considered as "nodes of the path". Therefore, the neighborhood information is obtained from the second degree neighbors. This new metric is known as *Second degree Correlation Factor* (*SdCF*). If the previous example is considered, SdCF values of combination$_1$ and combination$_2$ values are 76 and 68, respectively. Similarly to CF, the best combination of discovered paths is combination$_2$ since it has the lowest value of SdCF.

## 3.4 HTLC-MeDSR design

The HTLC-MeDSR implementation is based on MeDSR that is a reactive routing protocol. In MeDSR, a node issues RREQ packets only when it has data to send. RREQ packets are flooded through the network and each

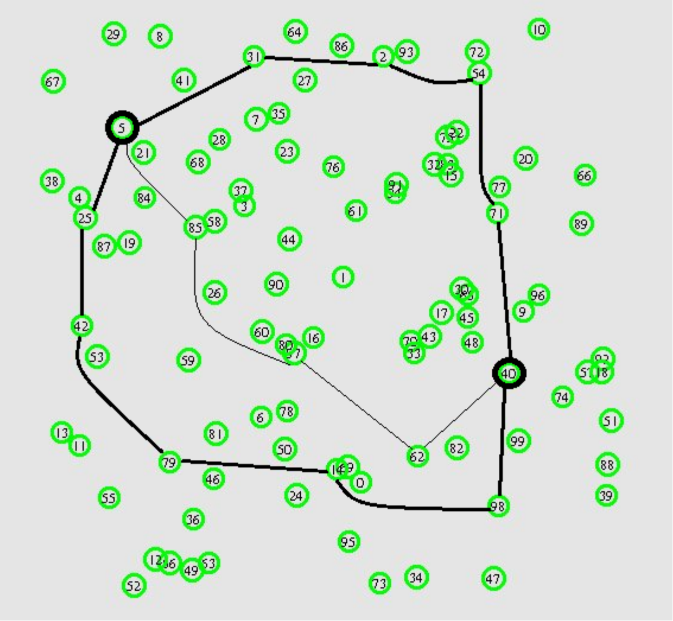


Figure 3.4: The correlation factor concept for a randomly distributed scenario with 100 nodes

node appends its own address to each request it receives, and then re-broadcasts it. The request originator issues a new RREQ packet for the same destination after an exponential back-off time if no RREP packet is received. The destination issues a RREP packet in response to the RREQ packets received. The RREP includes the path which was accumulated as the request was forwarded through the network. The RREP is source-routed back to the originator along the reverse path.

In MeDSR, FRFs due to collisions are misinterpreted as link failures. HTLC-MeDSR uses probe packets to detect link failures and the standard SRL was modified to 15 (as for both CBR and Pareto traffic, a higher throughput could be obtained, as explained in section 3.1). These two mechanisms allow for the reduction of packet loss and for the increase of throughput. The accuracy in determining FRFs could be increased: if each node overheard packet transmissions from other nodes, and if all transmitted packets from neighboring nodes were considered probe packets, which would mean that that neighbor is still alive.

### 3.4.1 Obtaining neighborhood information

Each node keeps its neighborhood information consisting of first and second degree neighbors in its neighbors cache. The direct neighbors (first degree neighbors) are those nodes from which HELLO packets are received. If no HELLO packet is received from a given node over the last $\omega$ seconds (e.g., 5 seconds), it is assumed that the node has failed and it is removed from the first degree neighbors cache. The information of the first degree neighbor is added to the HELLO packet and broadcast. Each node that receives the information of the first degree neighbor adds it to its second degree neighbors cache. The neighborhood information is used to calculate CF and SdCF.

### 3.4.2 Obtaining ETX information

To allow ETX calculation, every node maintains caches for *LQ* and *NLQ*. Each node sends periodically HELLO packets which are used to determine *LQ*. This *LQ* information is added to the HELLO packet. Each node receiving the LQ information, adds it to its *NLQ* cache, and uses it to calculate the ETX. This ETX information is also added to the HELLO packet. As HELLO packets are broadcast, each node has the ETX information of its first and second degree neighbors, which is therefore added to the RREQ packet during the route discovery process.

### 3.4.3 Building route sets

During the route discovery process, each RREQ packet collects neighborhood and ETX information of each node in the path to the destination. With this information, a graph can be built and well known path finding algorithms such as Dijkstra's [CLRS09], A* [HNR68], etc, can be used to find the shortest path between the source and destination nodes. It is assumed here that the destination node is powerful enough to run these algorithms.

Let $s$, $t$, $S_s$ and $S_t$ be the source and destination nodes, and the first degree neighbors of the source and destination nodes, respectively. Let $\mathcal{A}_l$ be the adjacency list based on the neighborhood information collected during the route discovery process. Let $S_{p_{st}}$ be the set of shortest paths between $s$ and $t$. Initially, $S_{p_{st}}$ is empty. Let $\delta_i$ be the usage counter of node $i$, which tells how many times node $i$ is used by the Dijkstra's algorithm. It is assumed that $\delta_s$ and $\delta_t$ are set to infinity. $\delta$ of any element of $S_s$ and $S_t$ is set to two because it increases the probability of finding more node disjoint paths. For higher values of $\delta$, the gains obtained are negligible. For all the other nodes, $\delta$ is set to one.

The algorithm used to find the list of available paths works as follows:

1. Run Dijkstra's algorithm at the destination node taking as inputs: $\mathcal{A}_l$ and $s$.

2. If the algorithm returns a path $p_{st}$, add it to $S_{p_{st}}$ and decrement $\delta$ of all the nodes in $p_{st}$.

3. Remove all nodes from $\mathcal{A}_l$ whose usage counter is less than one.

4. Repeat all previous steps until no more paths are returned.

The ETX information is added to all the discovered paths. These paths are grouped in pairs and the correlation factors of all possible combinations of pairs of paths are determined. The resulting pairs are sorted according to

$$(\text{ETX}_{p_1} + \text{ETX}_{p_2}) \times \left(\text{CF} + \frac{\text{SdCF}}{2}\right) \tag{3.2}$$

where $\text{ETX}_{p_i}$ is the ETX value of path $p_i$. The first term of equation 3.2, corresponds to the sum of the path's ETX values. It can be seen from the second term that the SdCF does not have the same weight as the CF. The SdCF is useful for situations where there are some pairs of paths with the same value of CF. Equation 3.2 takes into account both ETX and CF which enabled HTLC-MeDSR to find the best pairs of paths, both in terms of smaller expected number of transmissions to reach the destination and smaller cross-interference.

At most, four pairs of paths are selected to be included in the route set. ETX and neighborhood information of all nodes in these paths is also added to the route set. To improve the tolerance against transmission errors, the route set is inserted in the RREP packet that is sent back to the source node for every path in it.

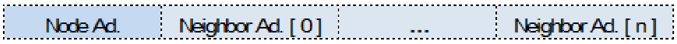


Figure 3.5: Neighborhood information structure

### 3.4.4 Data transmission

With equation 3.2, the source node selects the best pair of paths and transmits data packets through them in a round robin manner. In other words, the source node will send one data packet through the path with the smallest ETX value, and then send the next packet through the other path. In case of route failure, a new pair of paths is selected to transmit data packets. By distributing data packets among the selected paths, the energy usage is balanced.

In case of data prioritization, the most important data is sent through the path with the smallest ETX value, as it is the path with the highest throughput. Therefore, the least important data is sent through the second path.

### 3.4.5 Cross-layer approach

HTLC-MeDSR uses a cross layer approach to address the FRF problem involving the MAC and routing layers. At the routing layer, ETX and CF values are used to find high throughput paths with low route coupling between them, if they exist. At the MAC layer, a modified value of SRL is used. In addition, probe packets are used to identify routing failures, which reduces packet loss and unnecessary route maintenance operations.

### 3.4.6 Scalability

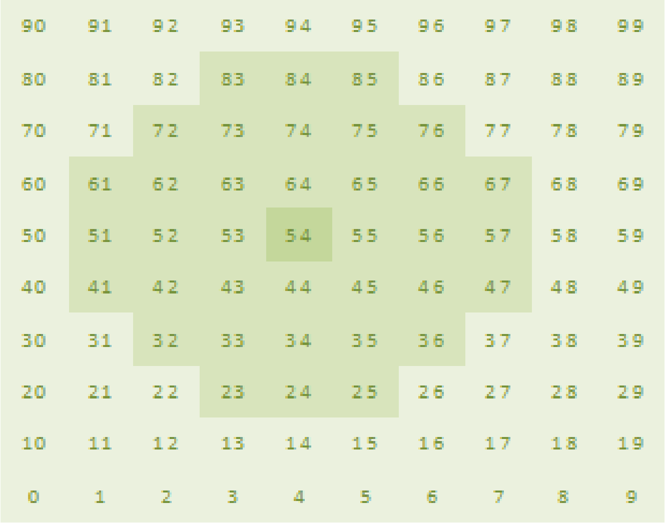
Since RREQ packets sent by the source node collect neighborhood and ETX information while traveling to the destination node, for small networks, this information is small, but for large networks, it may exceed the maximum IEEE802.11 [IEE12] frame body size of 2312 bytes. When a normal packet (i.e., neither broadcast nor multicast) exceeds the maximum frame body size, it is fragmented. As RREQ packets are sent in broadcast without retransmission, the ones exceeding the maximum frame body size would not be sent neither fragmented, therefore reducing the probability of finding paths at the destination node because of collisions and contention occurring during the Route Discovery process.

Two schemes are used in order to reduce the size of the collected information: (1) use of node identifiers (2) data compression. These techniques can be used individually or combined depending on the situation, and are explained in the following subsections.

#### Use of Node Identifiers

Figure 3.5 shows the neighborhood information structure and Figure 3.6 shows a 10x10 dense grid network. Consider that the distance between nodes and the transmission range allows, for example, node 54 to have the following 36 direct neighbor nodes: {23, 24, 25, 32, 33, 34, 35, 36, 41, 42, 43, 44, 45, 46, 47, 51, 52, 53, 55, 56, 57, 61, 62, 63, 64, 65, 66, 67, 72, 73, 74, 75, 76, 83, 84, 85}. The size of node 54 neighborhood information structure is 148 bytes[1]. By applying the same logic to all nodes in this particular network (see Figure 3.6), that is, from node 0 to node 99, the neighborhood information occupies 10880 bytes.

[1] An IPv4 node address has 32bits (4 bytes).



|  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 |
| 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 |
| 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 |
| 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 |
| 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 |
| 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Figure 3.6: 10x10 Grid network.

The proposed scheme assigns a node identifier of 8/16 bits to every node in the network. Consider, for example, node identifiers of 8 bits. Since all nodes in the network have 1 byte identifier, that accounts for 100 bytes. Now, the size of node 54 neighborhood information structure occupies 37 bytes. This allowed a reduction of 75%. Applying the same logic to the entire network, i.e., for nodes 0 to 99, now, the occupied size is of 2820 bytes. The attained reduction, for the latter case, was of 74.1%.

#### Data Compression

For larger networks, even with the scheme described in the previous subsection, additional compression mechanisms are necessary to allow a RREQ packet to collect neighborhood and ETX information while traveling through the network until it reaches the destination node.

The authors of [SKLA12] surveyed practical data compression algorithms for wireless sensor networks, i.e., algorithms that are based on real world requirements (e.g. minimizing power consumption). According to the authors, data compression approaches can be classified as: distributed data compression, which is usually applied in dense sensor networks, and local data compression, which performs data compression locally on each node without distributed collaboration among sensor nodes. The local data compression approach can be subdivided in two techniques: lossless compression algorithms, which ensure that the information is correct during the processes of compression and decompression, and lossy compression algorithms, in which some information may be lost.

Each node that deals with a RREQ packet must be able to compress neighborhood and ETX information, or decompress it in case of the destination node. In HTLC-MeDSR, it is assumed that the collected information always fits in a single packet, therefore being used a lossless compression algorithm. The use of additional compression mechanisms such as lossy compression, by using RSSI information to discard some low quality links, is left for future study.

| Parameter | Value |
|---|---|
| Network field | 700 m x 700 m |
| Number of Sinks/Number of Sources | 1/1 |
| Packet Size | 210 bytes |
| Idle power | 1.0 8mW |
| Receive power | 1.3 mW |
| Transmit power | 1.875 mW |
| Sleep power | 0.045 mW |
| Radio Propagation Model | Two Ray Ground |
| Traffic types | CBR, Pareto On/Off |
| MAC Layer | IEEE 802.11a |
| Physical Layer data rate | 6 Mbps |
| Simulation time | 210 seconds |
| Number of Sensor | 12, 100 |
| Node Energy | 1 MJ |
| Source Data rates | 128 – 1536 Kbps |

Table 3.1: Simulation parameters

## 3.5 Simulation model

The performance of the HTLC-MeDSR protocol in comparison with a single path and multipath routing protocols is analyzed. The simulation model was based on NS-2 [FV11]. The channel capacity of every node was set to 6Mbps. All transmitters had the same transmission range. The distributed coordination function (DCF) of IEEE 802.11 for wireless LANs was used. The 802.11Ext [CSEJ$^+$07a] was used as MAC protocol simulator for NS-2 LANs because it uses an additive interference model [MPG11].

Two different network scenarios were considered: scenario 1 (S1) consisting of 12 nodes (see Figure 3.3) and scenario 2 (S2) consisting of 100 nodes (see Figure 3.6). Nodes in both scenarios are distributed over an area of 700 m x 700 m. S1 consisted of two non-interfering paths. S2 consisted of a 10 x 10 grid network where nodes were 50 m apart from each other. Only static scenarios were considered because they are more common in WMSN.

The simulation duration was set to 210 s. Ten runs of simulations were performed for both scenario, and the results were averaged. Different seeds were used in order to get non-deterministic results across runs.

The types of User Datagram Protocol (UDP) traffic considered in the simulation were: Constant Bit Rate (CBR) and Pareto On/Off with transmission periods of 50% of the time on average. The size of the data packets was set to 210 bytes. There is only one source/destination traffic pair. For each traffic source, the following data rates were used: 128, 256, 512, 768, 1024, 1280 and 1536 Kbps. The average throughput, end-to-end delay, packet loss, control overhead, energy efficiency and path length were analyzed for both scenarios with their 95% confidence intervals.

Each node contains an initial energy of 1 MJ. Overall, it gives amounts of 12 MJ and 100 MJ for groups of 12 and 100 nodes, respectively. Since all nodes participate in the sensor network, they use their energy to transmit or to receive packets as well as while operating in standby mode. Nodes can only participate in the network if they have energy. The values of power consumption in four radio states, including idle, transmit, receive and sleep state were set according to [SBS02], i.e., $P_{idle}$ = 1.08 mW, $P_{tx}$ = 1.875 mW, $P_{rx}$ = 1.3 mW, and $P_{slp}$ = 0.045 mW.

Table 3.1 show the simulation parameters.

## 3.6 Simulation results

The performance is evaluated according to the following metrics:

- *Throughput*: This metric represents the ratio between the number of data packets that are sent by the source and received by the destination during the simulation over the simulation time.
- *Average End-to-End Delay*: The end-to-end delay is averaged over all surviving packets from the source to the destination.
- *Packet loss ratio*: This metric represents the ratio between the number of dropped packets over all packets.
- *Control overhead*: The control overhead represents the ratio between the total number of routing control packets over all packets.
- *Energy efficiency*: The energy efficiency measures the energy dissipated per transmitted bit by the nodes throughout the entire simulation.
- *Path length*: The path length is averaged over all surviving packets from the source to the destination.

The performance of the following protocols is evaluated and compared:

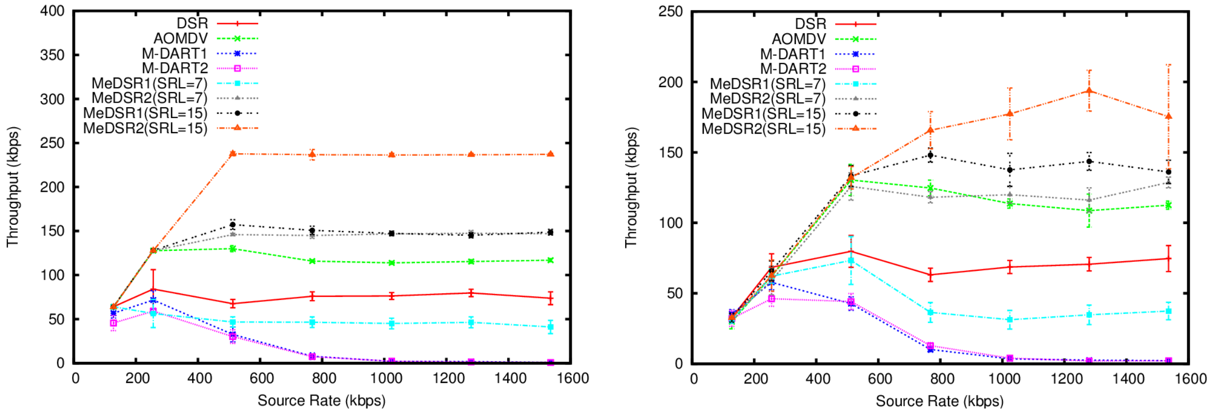
- DSR: Dynamic Source Routing protocol that is a single path routing protocol.
- AOMDV: Ad hoc On-demand Multipath Distance Vector that only uses one path to forward traffic. It keeps alternative paths as backups if the main path fails.
- MDART1: Multipath Dynamic Address Routing protocol [CP11], which uses a single path and keeps other paths as backup.
- MDART2: MDART using two simultaneous paths to forward traffic.
- MeDSR1: The Multipath Extension to Dynamic Source Routing (MeDSR) protocol.
- MeDSR2: High Throughput Low Coupling Multipath Extension to the Dynamic Source Routing (HTLC-MeDSR) protocol.

Please note that most of the routing protocols listed before were revised in Section 2.2.1.

### 3.6.1 Throughput

Multipath routing protocols in a fault tolerance configuration like M-DART1, only use one path to forward traffic and another path for backup. For small source rates, single path protocols perform better than those using multiple paths simultaneously (e.g., MDART2) because nodes suffer less contention and collisions while forwarding traffic through this single path. The same happens between DSR and MeDSR1 with normal SRL as they use SRL to detect link failures.

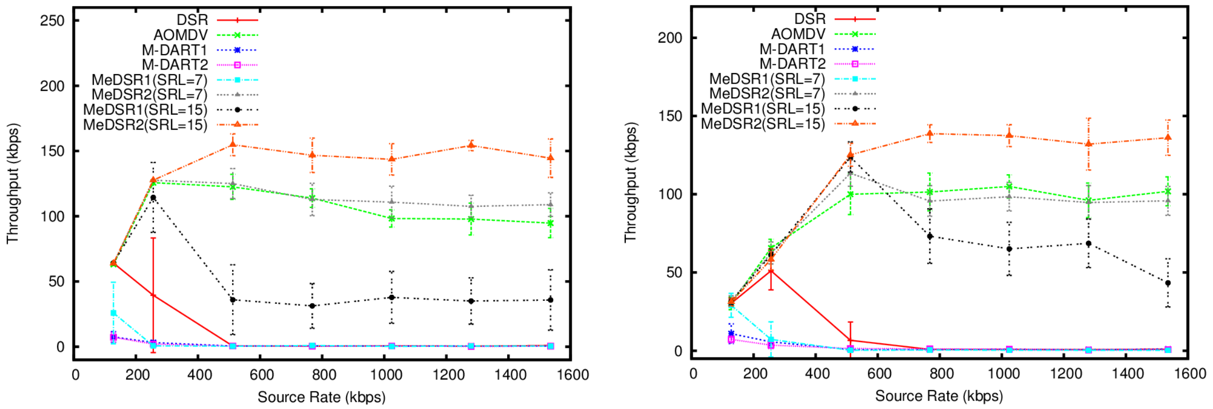
In S1, MeDSR1 with normal SRL uses two independent paths to forward traffic. Therefore, it has to deal with contention and collisions in these two paths, and with the increase of source rate, link failures in both paths cause



(a) CBR traffic

(b) Pareto traffic

Figure 3.7: Throughput as function of source rate on S1



(a) CBR traffic

(b) Pareto traffic

Figure 3.8: Throughput as function of source rate on S2

additional performance degradation. However, if SRL is increased more MAC level retransmissions are permitted per node. Figure 3.7 shows that there is a throughput increase since more packets are delivered to the destination. This validates the results previously obtained in Section 3.1. As AOMDV and MeDSR2 use probe packets to detect link failure, they perform better than all other routing protocols since this mechanism reduces FRFs. Figure 3.7 also shows that using multiple paths to forward traffic increases throughput if the paths have no cross-interference. Similarly to MeDSR1 with SRL = 15, with the increase of SRL in MeDSR2, there is also a throughput increase for both CBR and Pareto traffic. For Pareto traffic (see Figure 3.7(b)), the throughput magnitude is smaller than the one achieved with CBR traffic because Pareto traffic is less intense due to burst and pause times.

The results for S2 are shown in Figure 3.8. If only normal SRL is considered, MeDSR2 performs almost similarly to AOMDV, but performing better if compared with other routing protocols for both traffic patterns. In S2, the difference in performance is due to the fact that routing protocols are able to find node disjoint paths between the source-destination pair, but the selected paths are not completely independent as the ones of S1. Consequently, the selected paths in S2 suffer more route coupling even for MeDSR2.

Figure 3.8 also shows that MeDSR1 with normal SRL performs worse than DSR. This happens because of the route coupling among the selected pair of multipath paths, as the MeDSR1 has to deal with a lot of contention and collisions, causing unnecessary new route discovery processes that severely degrade performance. For S2 and resembling to S1, the increase of SRL in MeDSR1 also causes an increase in terms of throughput in comparison with other routing protocols. The same happens to MeDSR2.

MDART performs worst because it is not able to find usable paths with the increase of the source rate due to excessive collisions.

| | | DSR | AOMDV | M-DART1 | M-DART2 | MeDSR1 |
|---|---|---|---|---|---|---|
| S1 | CBR | 77.1 | 17.8 | 433 | 531 | 158 |
| | Pareto | 52.9 | 2.7 | 355 | 389 | 120 |
| S2 | CBR | 611 | 5.6 | 5737 | 6609 | 2513 |
| | Pareto | 536 | -1.3 | 2774 | 3734 | 1457 |

Table 3.2: MeDSR2 throughput gains for normal SRL (%)

Table 3.2 presents the average throughput gains comparison for SRL=7 between MeDSR2 and other routing protocols on S1 and S2. MeDSR2 is on average 1147% better than protocols that use one path to forward traffic (e.g. DSR, AOMDV and M-DART1) for CBR traffic and 620% better for Pareto traffic. It is also on average 2453% better than protocols that use two paths to forward traffic (e.g. M-DART2 and MeDSR1) for CBR traffic and 1425% for Pareto traffic.

| | | DSR | AOMDV | MeDSR1 | | MeDSR2 |
|---|---|---|---|---|---|---|
| | | SRL=7 | SRL=7 | SRL=7 | SRL=15 | SRL=7 |
| S1 | CBR | 164 | 75.6 | 284 | 46.1 | 49.0 |
| | Pareto | 105 | 37.7 | 195 | 17.8 | 34.1 |
| S2 | CBR | 779 | 30.6 | 3132 | 164 | 23.6 |
| | Pareto | 716 | 26.5 | 1895 | 65.1 | 28.2 |

Table 3.3: MeDSR2 throughput gains for SRL = 15 (%)

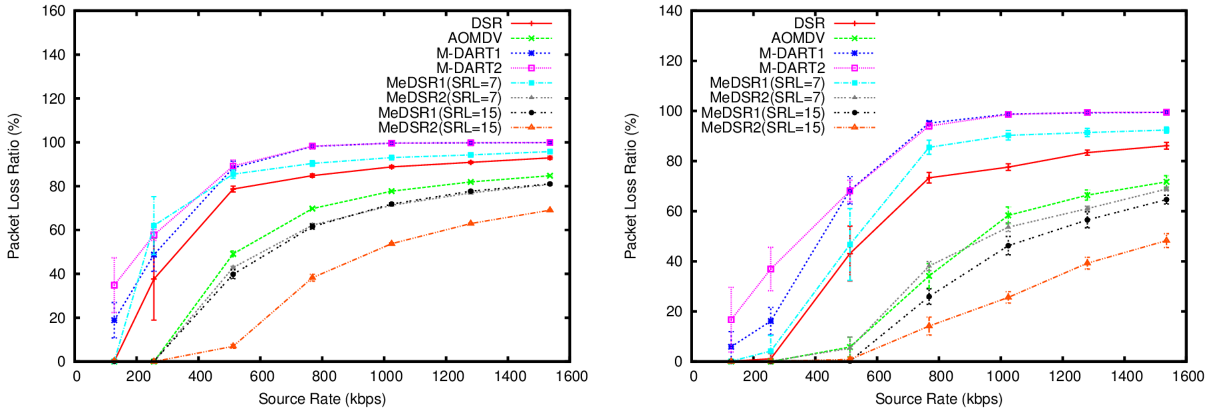
Table 3.3 presents the average throughput gains comparison between MeDSR2 with SRL = 15 and other MeDSR implementations with SRL = 7 and SRL = 15, DSR and AOMDV for both scenarios. MeDSR2 with SRL = 15 is on average 616% better than other MeDSR implementations for CBR traffic and 373% better for Pareto traffic.

In comparison with DSR, MeDSR2 is on average 472% better for CBR traffic and 411% better for Pareto traffic. If compared to AOMDV, MeDSR2 is on average 53.1% better for CBR traffic and 32.1% better for Pareto traffic.

### 3.6.2 Packet loss ratio

Figure 3.9 shows the packet loss ratio for S1. For normal SRL, MeDSR2 presents less packet loss than all other protocols because it uses node independent paths and probe packets to detect link failure. With the increase of SRL, MeDSR1 and MeDSR2 use the additional attempts to successfully deliver more packets to the next hop, at the expense of an additional delay.

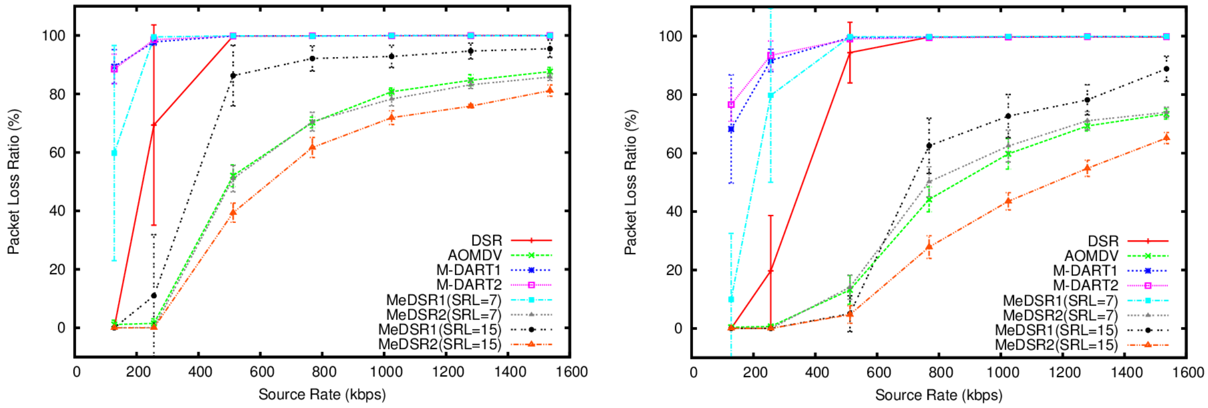
Figure 3.10 shows the packet loss ratio for S2. These results show that over some conditions single path routing protocols tend to perform better than multipath routing protocols regarding the packet loss ratio. As an example, consider DSR and MeDSR1 with normal SRL, and both version of MDART. For smaller source rates,



(a) CBR traffic

(b) Pareto traffic

Figure 3.9: Packets loss ratio gains as function of source rate on S1



(a) CBR traffic

(b) Pareto traffic

Figure 3.10: Packets loss ratio gains as function of source rate on S2

single path routing protocols, like DSR and MDART1, suffer less from concurrent transmissions than multipath routing protocols like MeDSR1 with normal SRL and MDART2.

Despite AOMDV and MeDSR2 with normal SRL being multipath routing protocols, they perform similarly in terms of packet loss as they use a similar mechanism to avoid FRFs. The increase of SRL enables protocols to reduce even more packet loss due to additional transmission attempts.

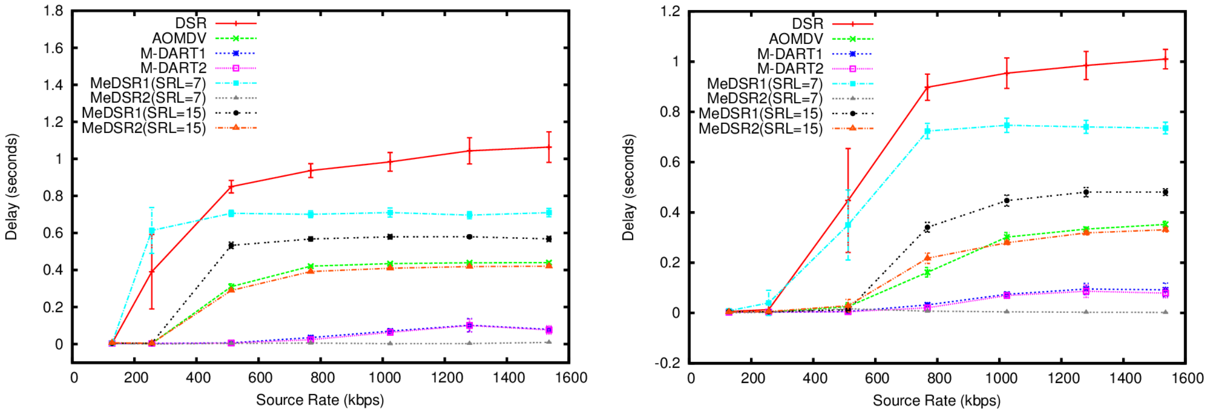
| | | DSR | AOMDV | M-DART1 | M-DART2 | MeDSR1 |
|---|---|---|---|---|---|---|
| S1 | CBR | 29.4 | 8.0 | 39.2 | 42.3 | 34.8 |
| | Pareto | 37.7 | 3.9 | 53.0 | 55.7 | 43.7 |
| S2 | CBR | 35.2 | 2.4 | 46.3 | 46.4 | 44.0 |
| | Pareto | 46.5 | 4.2 | 58.7 | 59.5 | 53.9 |

Table 3.4: MeDSR2 packet drop ratio gains for normal SRL (%)

Table 3.4 presents the average packet loss ratio gains comparison for normal SRL between MeDSR2 and other routing protocols for both scenarios. MeDSR2 is on average 26.7% better than protocols that use one path to forward traffic for CBR traffic and 34% better for Pareto traffic. It is also on average 27.9% better than protocols that use two paths to forward traffic for CBR traffic and 53.2% for Pareto traffic.

|  |  | DSR | AOMDV | MeDSR1 |  | MeDSR2 |
|---|---|---|---|---|---|---|
|  |  | SRL=7 | SRL=7 | SRL=7 | SRL=15 | SRL=7 |
| S1 | CBR | 51.2 | 36.4 | 54.9 | 30.4 | 30.8 |
|  | Pareto | 64.8 | 45.7 | 68.2 | 33.7 | 43.5 |
| S2 | CBR | 42.0 | 12.7 | 49.9 | 30.1 | 10.5 |
|  | Pareto | 61.4 | 24.8 | 66.7 | 35.4 | 27.8 |

Table 3.5: MeDSR2 packet drop ratio gains for SRL = 15 (%)



(a) CBR traffic

(b) Pareto traffic

Figure 3.11: Average end-to-end delay as function of source rate on S1

Table 3.5 presents the average packet loss ratio gains comparison between MeDSR2 with SRL = 15 and other MeDSR implementations with SRL = 7 and SRL = 15, DSR and AOMDV for both scenarios. MeDSR2 with SRL = 15 is on average 34.4% better than other MeDSR implementations for CBR traffic and 45.9% better for Pareto traffic.

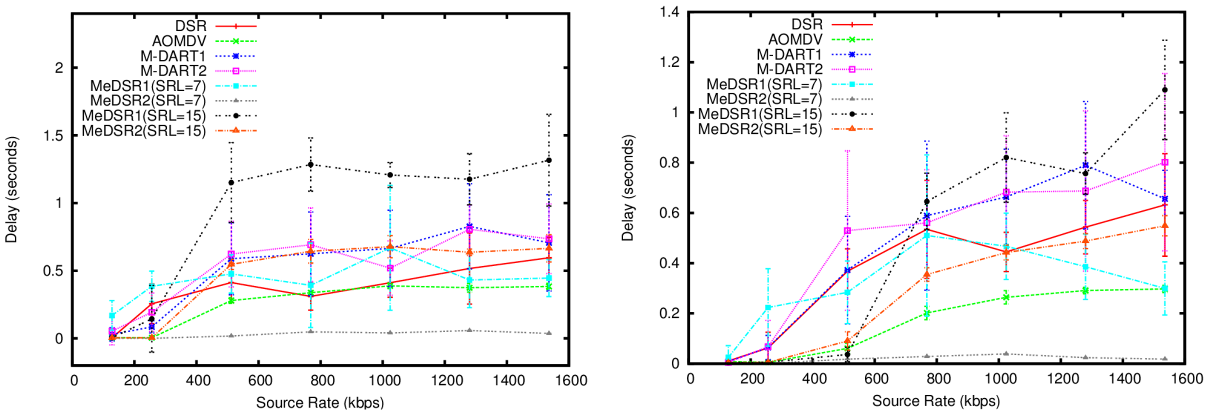
In comparison with DSR, MeDSR2 is on average 46.6% better for CBR traffic and 63.1% better for Pareto traffic. If compared to AOMDV, it is on average 24.6% better for CBR traffic and 35.3% better for Pareto traffic.

### 3.6.3 Average end-to-end delay

For S1, DSR presents a low end-to-end delay, for small values of source rates and for both CBR and Pareto traffic (see Figure 3.11). Above these values, DSR presents a considerable increase in the end-to-end delay. This happens because it uses only one path to forward traffic and if that path becomes broken, salvaged packets suffer additional delay since intermediate nodes might not have alternative paths to the destination node. In these situations, the route maintenance procedure initiates a route discovery process attempting to discover new paths to the destination node.

MeDSR1 performs better because it uses multiple paths to forward traffic. By using multiple paths alternatively, there is less load in each path, resulting in a smaller end-to-end delay. It can also be seen that MeDSR1 with SRL = 15 performs better than the normal SRL on S1 due to the use of more MAC retransmissions as previously explained.

MDART versions have less end-to-end delay than AOMDV and MeDSR2 with SRL = 15 because only a small subset of packets sent by the source node is received by the destination node, and these received packets have



(a) CBR traffic

(b) Pareto traffic

Figure 3.12: Average end-to-end delay as function of source rate on S2

small end-to-end delay. The increase of SRL despite its benefits such as the reduction of packet loss, has some disadvantages such as the increase of end-to-end delay. This is caused by the additional MAC level retransmissions over sucessive nodes in a path. Figures 3.11 and 3.12 show that MeDSR2 with normal SRL presents the smallest values of end-to-end delay. Since MeDSR2 with normal SRL and AOMDV have similar values of throughput (see Figure 3.8 or Table 3.2), MeDSR2 is a good choice for small end-to-end delay applications. If MeDSR2 with normal SRL on both scenarios are compared, the set of changes proposed were useful in selecting paths with smaller end-to-end delay.

Table 3.6 presents the average delay gains comparison for normal SRL between MeDSR2 and other routing protocols for both scenarios. MeDSR2 is on average 94% better than protocols that use one path to forward traffic for both CBR and Pareto traffic. It is also 94.3% better on average than protocols that use two paths to forward traffic for CBR traffic and 93.9% for Pareto traffic.

| | | DSR | AOMDV | M-DART1 | M-DART2 | MeDSR1 |
|---|---|---|---|---|---|---|
| S1 | CBR | 99.4 | 98.7 | 91.0 | 90.3 | 99.3 |
| | Pareto | 99.1 | 96.9 | 88.3 | 86.6 | 98.9 |
| S2 | CBR | 92.0 | 88.7 | 94.3 | 94.5 | 93.3 |
| | Pareto | 95.3 | 88.7 | 95.9 | 96.2 | 94.2 |

Table 3.6: MeDSR2 average end-to-end delay gains for normal SRL (%)

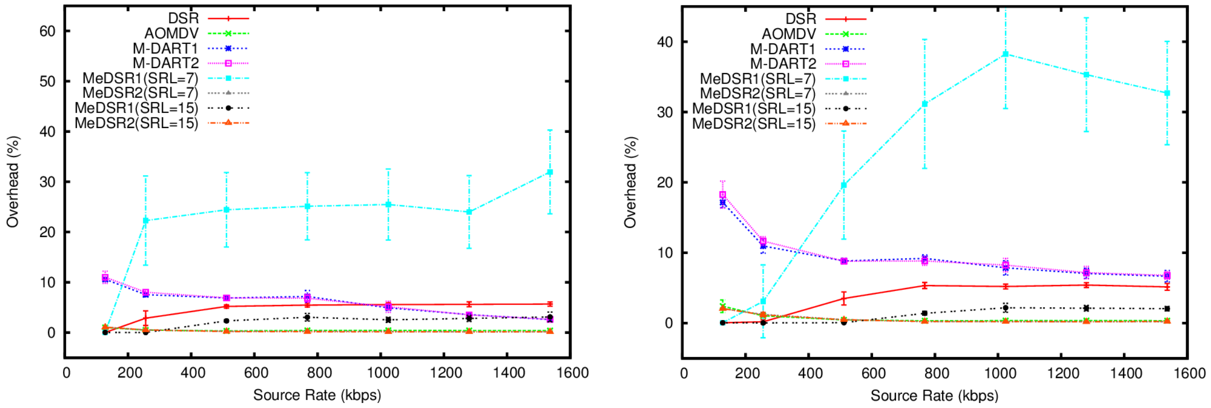
Table 3.7 presents the average delay gains comparison between MeDSR2 with SRL = 15 and other MeDSR implementations with SRL = 7 and SRL = 15, DSR and AOMDV for both scenarios. MeDSR2 with SRL = 15 is on average 31.5% better than other MeDSR1 versions for CBR traffic and 36.9% better for Pareto traffic.

MeDSR2 with SRL = 15, despite presenting considerable gains in comparison with both MeDSR1 implementations, does not perform as good in terms of delay if compared with the normal SRL version. MeDSR2 with normal SRL is on average 4304% better than SRL = 15 version for CBR traffic and 2314% better for Pareto traffic.

In comparison with DSR, MeDSR2 is on average 18.1% better for CBR traffic and 50.6% better for Pareto traffic. If compared to AOMDV, MeDSR2 is on average 37% worse for CBR traffic and 35.9% worse for Pareto traffic.

|  |  | DSR | AOMDV | MeDSR1 |  | MeDSR2 |
|---|---|---|---|---|---|---|
|  |  | SRL=7 | SRL=7 | SRL=7 | SRL=15 | SRL=7 |
| S1 | CBR | 63.1 | 5.3 | 52.1 | 31.6 | -7121 |
|  | Pareto | 72.5 | -0.4 | 63.1 | 33.2 | -3213 |
| S2 | CBR | -26.9 | -79.3 | -7.1 | 49.4 | -1487 |
|  | Pareto | 28.6 | -71.3 | 11.9 | 39.6 | -1415 |

Table 3.7: MeDSR2 average end-to-end delay gains for SRL = 15 (%)



(a) CBR traffic

(b) Pareto traffic

Figure 3.13: Control overhead as function of source rate on S1

### 3.6.4 Control overhead

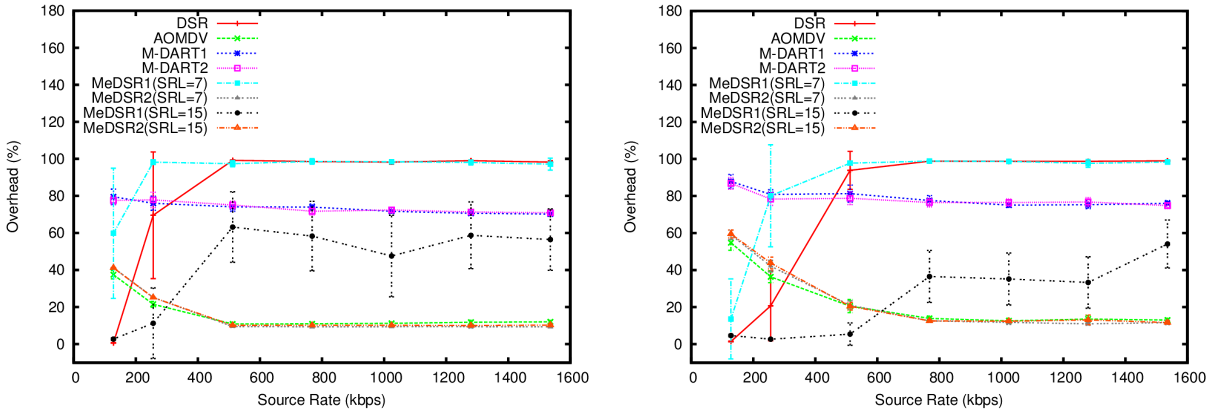
Figure 3.13 shows, for S1, that AOMDV and MeDSR2 have less control overhead because they use probe packets to detect link failures. For small source rates, DSR presents smaller control overhead. But as the source rate increases, collisions and contention also increase as does the control overhead. MeDSR1 with normal SRL presents a higher amount of control overhead since it uses multiple paths to forward traffic and it has to deal with contention and collisions in both paths.

For smaller source rates, MeDSR2 and AOMDV present more overhead because the number of control packets tends to be higher than data packets (see Figure 3.14). In addition, for small source rates, the number of generated data packets is quite low but with the increase of the source rate, the number of data packets also increases, thus considerably reducing the percentage of control overhead. MDART performs similarly but presents a higher control overhead as the number of delivered packets is very small.

By analyzing MeDSR1 with normal SRL, one can conclude that the number of FRFs is very high. FRFs are directly responsible for new route discovery processes which also increase the control overhead.

|  |  | DSR | AOMDV | M-DART1 | M-DART2 | MeDSR1 |
|---|---|---|---|---|---|---|
| S1 | CBR | 91.3 | 23.7 | 93.9 | 94.0 | 98.2 |
|  | Pareto | 81.4 | 12.9 | 93.2 | 93.4 | 96.9 |
| S2 | CBR | 79.9 | 2.1 | 78.1 | 78.1 | 82.5 |
|  | Pareto | 66.5 | 2.8 | 69.5 | 69.2 | 71.1 |

Table 3.8: MeDSR2 control overhead gains for normal SRL (%)



(a) CBR traffic

(b) Pareto traffic

Figure 3.14: Control overhead as function of source rate on S2

| | | DSR | AOMDV | MeDSR1 | | MeDSR2 |
|---|---|---|---|---|---|---|
| | | SRL=7 | SRL=7 | SRL=7 | SRL=15 | SRL=7 |
| S1 | CBR | 92.1 | 30.2 | 98.4 | 82.6 | 8.6 |
| | Pareto | 81.5 | 13.3 | 96.9 | 41.8 | 0.4 |
| S2 | CBR | 79.3 | -1.1 | 81.9 | 60.8 | -3.3 |
| | Pareto | 65.7 | -5.4 | 70.4 | -1.2 | -2.5 |

Table 3.9: MeDSR2 control overhead gains for SRL = 15 (%)

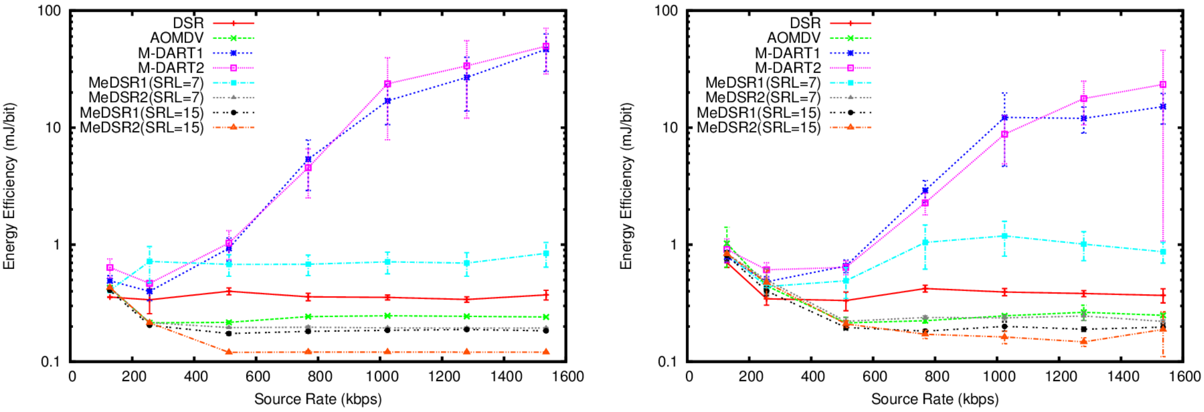
Table 3.8 presents the average control overhead gains comparison for normal SRL between MeDSR2 and other routing protocols for both scenarios. MeDSR2 is on average 62.1% better than protocols that use one path to forward traffic for CBR traffic and 54.4% better for Pareto traffic. It is also 88.2% better on average than protocols that use two paths to forward traffic for CBR traffic and 82.7% for Pareto traffic.

Table 3.9 presents the average control overhead gains comparison between MeDSR2 with SRL = 15 and other MeDSR implementations with SRL = 7 and SRL = 15, DSR and AOMDV for both scenarios. MeDSR2 with SRL = 15 is on average 54.8% and 34.3% better than other MeDSR implementations for CBR and Pareto traffic respectively.

In comparison with DSR, MeDSR2 is on average 85.7% better for CBR traffic and 73.6% better for Pareto traffic. If compared to AOMDV, MeDSR2 is on average 14.6% better for CBR traffic and 3.9% better for Pareto traffic.

### 3.6.5 Energy efficiency

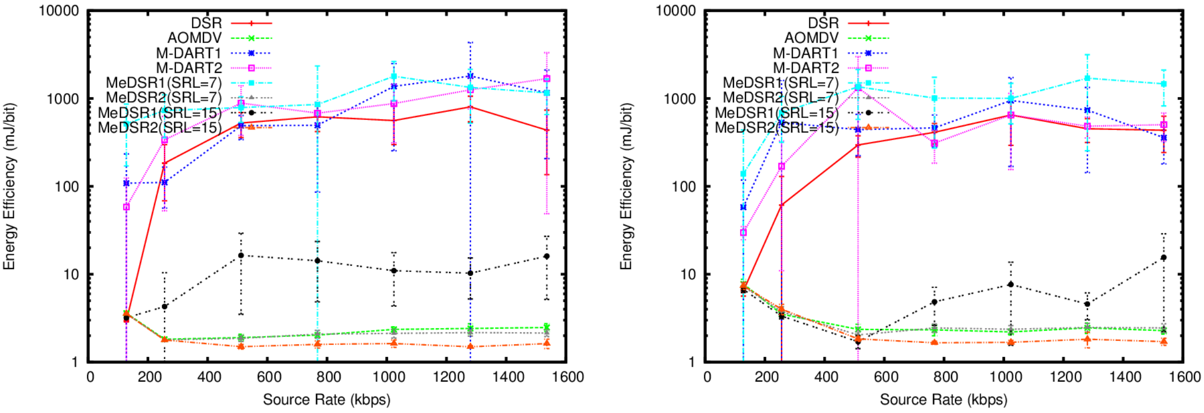
Figures 3.15 and 3.16 show that DSR presents the best values of energy efficiency, for a source rate equal to 128Kbps, since a single path routing protocol is sufficient for transmission at low data rates, and it does not use additional control packets like AOMDV and MeDSR2. But with the increase of the source rate, nodes suffer more packet loss which causes an increase in energy consumption and consequently a reduction of throughput. MeDSR1 presents a similar behavior to DSR, but the normal SRL version consumes more energy as it experiences more control overhead than other routing protocols. MeDSR2 is the most energy efficient protocol because it presents similar values of energy consumption as other routing protocols but achieves higher values of throughput. M-DART



(a) CBR traffic

(b) Pareto traffic

Figure 3.15: Energy Efficiency as function of source rate on S1



(a) CBR traffic

(b) Pareto traffic

Figure 3.16: Energy Efficiency as function of source rate on S2

is the routing protocol that performs worst in both scenarios since it presents the lowest values of throughput.

Table 3.10 presents the average energy efficiency gains comparison for normal SRL between MeDSR2 and other routing protocols for both scenarios. MeDSR2 is on average 58% better than protocols that use one path to forward traffic for CBR traffic and 51.4% better for Pareto traffic. It is also on average 89.2% better than protocols that use two paths to forward traffic for CBR traffic and 85.1% for Pareto traffic.

| | | DSR | AOMDV | M-DART1 | M-DART2 | MeDSR1 |
|---|---|---|---|---|---|---|
| S1 | CBR | 34.4 | 11.7 | 97.8 | 97.9 | 59.9 |
| | Pareto | 15.3 | 3.3 | 93.2 | 93.5 | 49.1 |
| S2 | CBR | 99.3 | 5.2 | 99.4 | 99.5 | 99.5 |
| | Pareto | 98.1 | 0.1 | 98.5 | 98.5 | 99.4 |

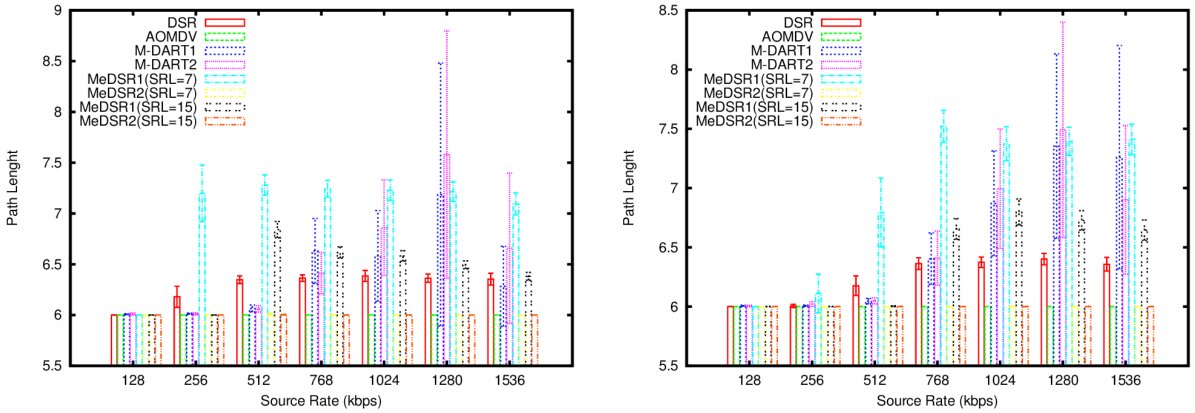
Table 3.10: MeDSR2 energy efficiency gains for normal SRL (%)

Table 3.11 presents the energy efficiency gains comparison between MeDSR2 with SRL = 15 and other MeDSR implementations with SRL = 7 and SRL = 15, DSR and AOMDV for both scenarios. MeDSR2 with SRL = 15 is on average 40.3% and 29.4% better than other MeDSR implementations for CBR and Pareto traffic respectively.

In comparison with DSR, MeDSR2 is on average 74.4% better for CBR traffic and 62.7% better for Pareto

| | | DSR | AOMDV | MeDSR1 | | MeDSR2 |
|---|---|---|---|---|---|---|
| | | SRL=7 | SRL=7 | SRL=7 | SRL=15 | SRL=7 |
| S1 | CBR | 49.4 | 31.9 | 69.1 | 18.1 | 22.9 |
| | Pareto | 26.9 | 16.6 | 56.1 | 1.6 | 13.7 |
| S2 | CBR | 99.4 | 20.3 | 99.6 | 15.9 | 15.9 |
| | Pareto | 98.4 | 11.6 | 99.4 | 11.6 | 11.6 |

Table 3.11: MeDSR2 energy efficiency gains for SRL = 15 (%)



(a) CBR traffic

(b) Pareto traffic

Figure 3.17: Path Length as function of source rate on S1

traffic. If compared to AOMDV, MeDSR2 is on average 26.1% better for CBR traffic and 14.1% better for Pareto traffic.

### 3.6.6 Path length

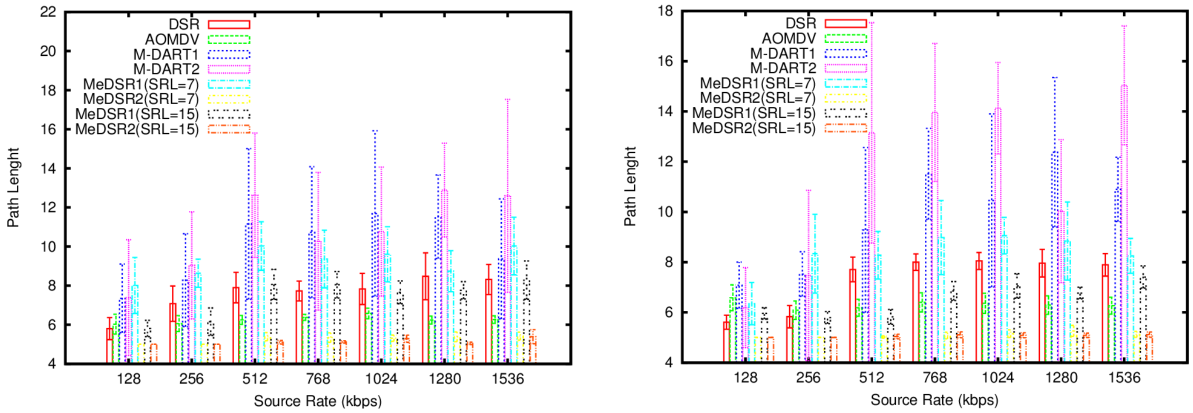
Routing protocols that use SRL to detect link failures tend to initiate a route discovery process more often, if intermediate nodes are not able to forward packets. On S1, the paths discovered by intermediate nodes do not have equal length, e.g., for node 5, one path has length 3 and another path has length 9 (which includes going back through the source node 0). As intermediate nodes attempt to salvage packets, they tend to use paths with longer path lengths. This happens even more with the increase of the source transmission rate, as more collisions occur, therefore resulting in an increase in the average path length.

MDART, in case of link failure, attempts to forward packets using other neighbor nodes that have paths with higher costs, resulting in higher average path lengths.

Figures 3.17 and 3.18 show that protocols with high control overhead tend to have also a high path length. MeDSR2 with SRL=15 achieves the best average path length.

| | | DSR | AOMDV | M-DART1 | M-DART2 | MeDSR1 |
|---|---|---|---|---|---|---|
| S1 | CBR | 4.5 | 0.0 | 6.2 | 7.9 | 14.4 |
| | Pareto | 3.8 | 0.0 | 8.6 | 8.4 | 13.6 |
| S2 | CBR | 30.1 | 15.9 | 47.4 | 51.2 | 42.8 |
| | Pareto | 29.9 | 19.3 | 48.4 | 55.3 | 38.6 |

Table 3.12: MeDSR2 path length Gains for normal SRL (%)



(a) CBR traffic

(b) Pareto traffic

Figure 3.18: Path Length as function of source rate on S2

Table 3.12 presents the average path length gains comparison for normal SRL between MeDSR2 and other routing protocols for both scenarios. MeDSR2 path length is on average 17.4% shorter than that of other routing protocols that use one path to forward traffic for CBR traffic and 18.3% shorter for Pareto traffic. It is also on average 29.1% better than that of the routing protocols that use two paths to forward traffic for CBR traffic and 28.9% for Pareto traffic.

| | | DSR | AOMDV | MeDSR1 | | MeDSR2 |
|---|---|---|---|---|---|---|
| | | SRL=7 | SRL=7 | SRL=7 | SRL=15 | SRL=7 |
| S1 | CBR | 4.5 | 0.0 | 14.4 | 6.5 | 0.0 |
| | Pareto | 3.8 | 0.0 | 13.6 | 6.3 | 0.0 |
| S2 | CBR | 32.4 | 17.9 | 44.2 | 30.6 | 2.5 |
| | Pareto | 30.3 | 19.8 | 39.0 | 21.8 | 0.6 |

Table 3.13: MeDSR2 path length gains for SRL = 15 (%)

Table 3.13 presents the average path length gains comparison between MeDSR2 with SRL = 15 and other MeDSR implementations with SRL = 7 and SRL = 15, DSR and AOMDV in both scenarios. MeDSR2 with SRL = 15 has an average path length 16.4% shorter than that of other MeDSR implementations for CBR traffic and 13.6% shorter for Pareto traffic.

In comparison with DSR, MeDSR2 is on average 18.5% better for CBR traffic and 17.1% better for Pareto traffic. If compared to AOMDV, MeDSR2 is on average 8.9% better for CBR traffic and 9.9% better for Pareto traffic.

## 3.7 Summary

A high throughput low coupling multipath routing protocol for wireless multimedia sensor networks has been proposed and its performance evaluated through simulation. In this approach, at the routing layer, nodes use ETX together with the first and second degree correlation factors between paths to find multiple high throughput paths with minimal interference between them. At the MAC layer, a modified value of SRL is used. In addition, probe packets are used to identify routing failures, which reduce packet loss and unnecessary route maintenance operations.

Simulation results show that HTLC-MeDSR presents considerable gains in comparison to other routing protocols in terms of the routing metrics' considered (throughput, delay, packet loss, control overhead, energy efficiency and path length).

By increasing SRL from 7 to 15 together with the mechanisms implemented at the routing layer, HTLC-MeDSR is 40%, 590% and 30% better in terms of throughput, in comparison with DSR, AOMDV and normal MeDSR, respectively. This shows the advantage of increasing the SRL value, despite the fact that certain caution should be taken because high SRL values can cause an overall performance degradation.

It was also noticed that in terms of delay, HTLC-MeDSR with normal SRL is 180%, 160% and 180% better in comparison with DSR, AOMDV and normal MeDSR, respectively. The increase of SRL, despite its throughput improvements, increases the end-to-end delay as more packets with additional latency are successfully delivered.

By increasing SRL from 7 to 15, HTLC-MeDSR is 50%, 550% and 10% better in terms of packet loss, and 10%, 300% better and 40% worst in terms of energy efficiency in comparison with DSR, AOMDV and normal MeDSR, respectively.

# Chapter 4

# A multi-objective routing algorithm for WMSNs

This chapter presents a new multi-objective optimization approach for addressing the routing problem in WMSNs. The proposed algorithm takes into account QoS requirements such as delay and ETX. Classical approximations optimize a single objective or QoS parameter, not taking into account the conflicting nature of these parameters which leads to sub-optimal solutions. The case studies applying the proposed multi-objective optimization routing algorithm show clear improvements on the QoS routing solutions even when compared to HTLC-MeDSR.

## 4.1 Description of the problem

The notation and terminology used, in this chapter, is borrowed from [YL01]. A WMSN can be represented by a connected graph $G(V, E)$ where $V$ is the set of vertices representing nodes and $E$ is the set of edges representing links between the nodes. Each edge $e = u \to v$ is associated with $k$ weights where $\omega_l(e) > 0, \forall e \in E$ and $1 \leq l \leq k$. Similarly to [YL01], it is assumed that all constraints are path constrains, and that the weight of a path is equal to the sum of the weights of all edges on the path. Thus, for each path $p = v_0 \to v_1 \to \cdots \to v_n, \omega_l(p) = \sum_{i=1}^{n} \omega_l(v_{i-1} \to v_i)$. A path constraint, e.g., delay, represents the end-to-end QoS requirement for the complete path.

**Definition 4.1.1.** Multi-constrained QoS routing problem. Given an undirected graph $G(V, E)$ with each edge $e$ associated with $k$ weight functions, where $\omega_l(e) > 0, \forall e \in E$ and $1 \leq l \leq k$, and a constant's vector $c = (c_1, c_2, \ldots, c_k)$. A multi-constrained QoS routing problem consists in finding a path $p$ between a source $s$ and destination $t$, so that, $\omega_l(p) \leq c_l$, where $1 \leq l \leq k$.

**Definition 4.1.2.** Multi-constrained Optimal QoS routing problem. Given an undirected graph $G(V, E)$ with each edge $e$ associated with $k$ weight functions, where $\omega_l(e) > 0, \forall e \in E$ and $1 \leq l \leq k$. A path $p = s \to v_1 \to \cdots \to t$ is considered an optimal QoS path from $s$ to $t$, if $\nexists\, q = s \to \cdots \to t$ such that $\omega(q) < \omega(p)$.

Each optimal path can possibly satisfy a particular QoS constraint not yet satisfied by any other path. QoS routing guarantees finding a path that satisfies the QoS constraints if it exists, by considering all QoS optimal paths. The number of optimal paths can grow exponentially with respect to the network size [YL01].

The WMSN QoS routing problem can be addressed meta-heuristically using multi-objective optimization algorithms, as explained in the following section.

## 4.2 Multi-objective optimization

### 4.2.1 Basic definitions

Multi-objective Optimization Problems (MOP) deals with more than one objective function which are to be minimized or maximized, subject to a number of constraints:

$$\left.\begin{array}{ll} \text{Minimize/Maximize } f_m(x), & m = 1, 2, ..., M; \\ \text{subject to } g_j(x) \geq 0, & j = 1, 2, \ldots, J; \\ h_k(x) = 0, & k = 1, 2, \ldots, K; \\ x_i^{(L)} \leq x_i \leq x_i^{(U)} & i = 1, 2, \ldots, n. \end{array}\right\} \tag{4.1}$$

where $M$ is the number of objective functions subject to $J$ inequalities and $K$ equality constraints. A solution $x$ is a vector of $n$ decision variables. The lower bound $x_i^{(L)}$ and upper bound $x_i^{(U)}$, restricting each decision variable $x_i$, constitute a decision variable space $D$. $L$ and $U$ are the Lower and Upper bounds restricting each decision variable.

A vector of $n$ decision variables $(x_1, x_2, ..., x_n)^T$ constitutes a solution $x$. A *feasible solution* $x$ is one that satisfies all constraints and decision variable bounds. The set of all feasible solutions is called the *feasible region* (or *search space* $S$).

**Definition 4.2.1.** Domination. A solution $x^{(1)}$ is said to dominate another solution $x^{(2)}$, if the following conditions are verified:

1. $x^{(1)}$ is not worse than $x^{(2)}$ in all objectives, or $f_m(x)^{(1)}$ is not worse than $f_m(x)^{(2)}$ for all $m = 1, 2, 3..., M$.
2. $x^{(1)}$ is strictly better than $x^{(2)}$ in at least one objective, or $f_{\overline{m}}(x^{(1)})$ is better than $f_{\overline{m}}(x^{(2)})$ for at least one $\overline{m} = 1, 2, 3, ..., M$.

**Definition 4.2.2.** Non-dominated set. For a set of solutions $P$, a non-dominated set of solutions $\overline{P}$ is a set of solutions that are not dominated by any member of the set $P$.

**Definition 4.2.3.** Globally Pareto-Optimal set. A Globally Pareto-Optimal set is a non-dominated set of the entire feasible search space $S$. Since solutions of the globally Pareto-optimal set are not dominated by any other within the search space, they are optimal solutions of the MOP, and are simply referred as the Pareto-optimal set (POS).

### 4.2.2 MOP formulation

The multi-objective optimization algorithm's goal is to produce a diverse set of optimal solutions that can be used by the user while evaluating trade-offs between different objectives.

### Objective functions

The goal is to minimize the following metrics: delay and ETX.

**Delay.** In computer networks [KR12], a packet originated in a source node passes through a set of intermediate nodes, until it reaches its destination node. During its travel from one node to a subsequent one along a path, the packet suffers, at each node, from several types of delays, namely nodal processing delay, queuing delay, transmission delay, and propagation delay. Processing delay $(d_{proc})$ can be seen as the time required (1) to examine the packet's header in order to decide where to direct the packet, and/or (2) to check the packet for bit-level errors that possible occurred while transmitting it to a subsequent node. Queueing delay $(d_{queue})$ corresponds to the delay packets suffer in nodes' queues while waiting to be transmitted onto the link. Transmission delay $(d_{trans})$ is the amount of time necessary to transmit all of the packets' bits into the link. Propagation delay $(d_{prop})$ is the time necessary for all packet's bits to propagate from the beginning of a link of a given node to the subsequent one. The packet's bits propagate at the propagation speed of the link, which depends on type of physical medium of the link. The propagation speed is in the range from $2 \cdot 10^8$ to $3 \cdot 10^8$ $meters/sec$. So, the nodal delay is defined as

$$delay = d_{proc}{+}d_{queue} + d_{trans} + d_{prop} \tag{4.2}$$

$$d_{trans} = \frac{L}{R} \tag{4.3}$$

$$d_{prop} = \frac{d}{s} \tag{4.4}$$

where $L$ is the packet length in $bits$, $R$ is the link transmission rate in $bits/sec$, $d$ is the distance among nodes in $meters$, $s$ is the link propagation speed in $meters/sec$.

Since wireless networks are composed of many nodes, the total delay $(d_{total})$, also called end-to-end delay, considering that there are $K - 1$ intermediate nodes between a source node and a destination node, is given by

$$d_{total} = \sum_{i=1}^{K} delay_i \tag{4.5}$$

where $K$ is the number of hops in the end-to-end path.

**Expected transmission count.** As mentioned in Section 3.2, ETX is given by

$$\text{ETX} = \frac{1}{\mathit{LQ} \times \mathit{NLQ}} \tag{4.6}$$

Considering that there are $K - 1$ intermediate nodes between a source-destination pair, the total ETX is the sum of the ETX metrics along the path, i.e.,

$$\text{ETX}_{total} = \sum_{i=1}^{K} \text{ETX}_i \tag{4.7}$$

where $K$ is the number of hops in the end-to-end path.

ETX is collected using HTLC-MeDSR assuming that there are no flows between the source and destination nodes.

**Constraints**

**Path constraint.** WMSN applications have different QoS requirements $(Q)$ such as bounded latency (or delay) $\mathfrak{L}$, bandwidth (or throughput) $B$, jitter $J$, packet loss $P$ and energy consumption $E$, so $\{\mathfrak{L}, B, J, P, E\} \subset Q$. Depending on the constraints, and in order to impose them it is necessary to check whether $min\{\omega_Q(p)\} \geq Q_{min}$ or $max\{\omega_Q(p)\} \leq Q_{max}$, where $Q_{min}$ or $Q_{max}$ are the minimum or maximum values allowed. For example, the bandwidth path constraint ensures that $min\{\omega_B(p)\} \geq B_{min}$, for all valid paths.

### 4.2.3 Strength pareto evolutionary algorithm

The Strength Pareto Evolutionary Algorithm (SPEA) [Deb08] is an elitist multi-objective evolutionary algorithm (MOEA) because it ensures that good solutions found in early runs are only replaced if better solutions are discovered. The algorithm achieves this by maintaining an external population $\overline{P}_l$ consisting of a fixed number of non-dominated solutions found before the beginning of the simulation. If new non-dominated solutions are found during the simulation, they are compared with the existing external population and the resulting non-dominated solutions are stored. The external population participates in the genetic operators with the current population expecting to influence the population towards good regions of the search space.

Below, one iteration of the algorithm is described step-by-step [Deb08]. Initially, a population $P_0$ of size $N$ is randomly created, and the external population $\overline{P}_0$ with maximum capacity of $\overline{N}$ is empty. In generation $t$,

1. Find the best non-dominated set of $P_t$ and copy these solutions to $\overline{P}_t$.

2. Find the best non-dominated solutions of the modified population $\overline{P}_t$ and delete all dominated solutions.

3. If $|\overline{P}_t| \geq \overline{N}$, the clustering technique must be used to reduce the external population size to $\overline{N}$. If not, do nothing to $\overline{P}_t$. The outcome is the external population $\overline{P}_{t+1}$ used in the next generation.

4. Assign fitness to each elite solution $i \in \overline{P}_{t+1}$ and to each population member $j \in P_t$.

5. In a minimization sense and with the previously assigned fitness values, apply genetic operators (described in 4.2.4) in order to create the new population $P_{t+1}$ of size $N$ using the combined population $(\overline{P}_{t+1} \cup P_t)$ of size $(\overline{N} + N)$.

The new external and current population that are used in the next generation, are obtained through steps 3 and 5 respectively. When the stopping criteria is satisfied, the algorithm stops.

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| *SN* | 1 | 2 | 3 | 4 | ... | *DN* |

Figure 4.1: Solution representation

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| 0 | 8 | 22 | 30 | 38 | 40 | 48 |

Figure 4.2: A candidate solution for the scenario of Figure 4.4(a)

### 4.2.4 SPEA implementation

A MOEA consists of the individual, the set of individuals also called population, the fitness of each individual and of the genetic operators applied to the population. Below, a description is given on how each of the components is obtained:

#### Individual

Each candidate solution (individual or chromosome) represents a path between a source node SN and a destination node DN. A link connects two consecutive nodes in a path (see Figure 4.1). For example, Figure 4.4(a) (Section 4.3.1) shows a simulation scenario with 49 nodes. The source node is node 0 and the destination node is node 48. Figure 4.2 shows a candidate solution marked on Figure 4.4(a)).

#### Population initialization

The initial population is created using the BFS graph search algorithm. It was previously mentioned that HTLC-MeDSR uses probe packets to determine ETX. A simulation with the HTLC-MeDSR routing protocol was performed on the NS-2 network simulator without any traffic source and all link's ETX and delay values were collected. The probe packet size was set to 2312 bytes, in order to measure the maximum transmission time. BFS algorithm was fed with all links so that it could compute a set of individuals. Since BFS explores all possible nodes and links among nodes, its time complexity is $O(|V|+|E|)$. Consequently, the BFS execution time depends on the network's size [CLRS09], i.e., the number of nodes and links, which is significant for large networks. To speed up the search process, each branch was probabilistically explored with a probability of $20\%$ and $1\%$ for scenarios $1$ and $2$, respectively (see Section 4.3.1). Thus, the search algorithm was not exhaustive and the feasible individuals were found using MOEA's genetic operators, i.e., crossover and mutation, to solve the problem as explained in the following subsections. The initial population is randomly selected from the sets of individuals obtained for both scenarios.

#### Fitness assignment

Fitness values are assigned to each individual of the initial population using the ETX and delay values collected in the previous step. An individual's ETX is the sum of its link's ETX values. An individual's end-to-end delay is the

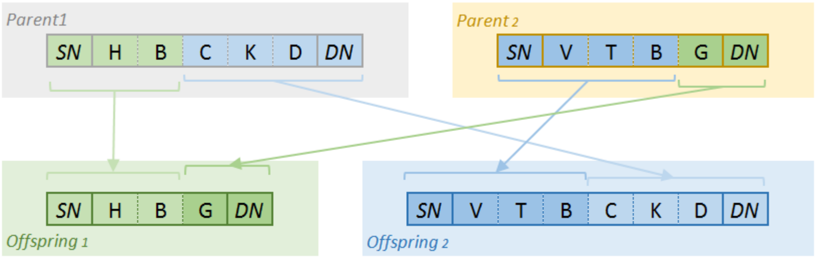


Figure 4.3: During the crossover operation, *offspring*$_1$ and *offspring*$_2$ are created by concatenating parts from $parent_1$ and $parent_2$. For example, *offspring*$_1$ results from the concatenation of nodes starting from *SN* to common node B from *parent*$_1$ and with the nodes following B to *DN* in *parent*$_2$

sum of its link's delay values. The set of non-dominated individuals is selected taking into account the individual's fitness values.

### Genetic operators

The goal of applying genetic operators is to produce better solutions than the current ones.

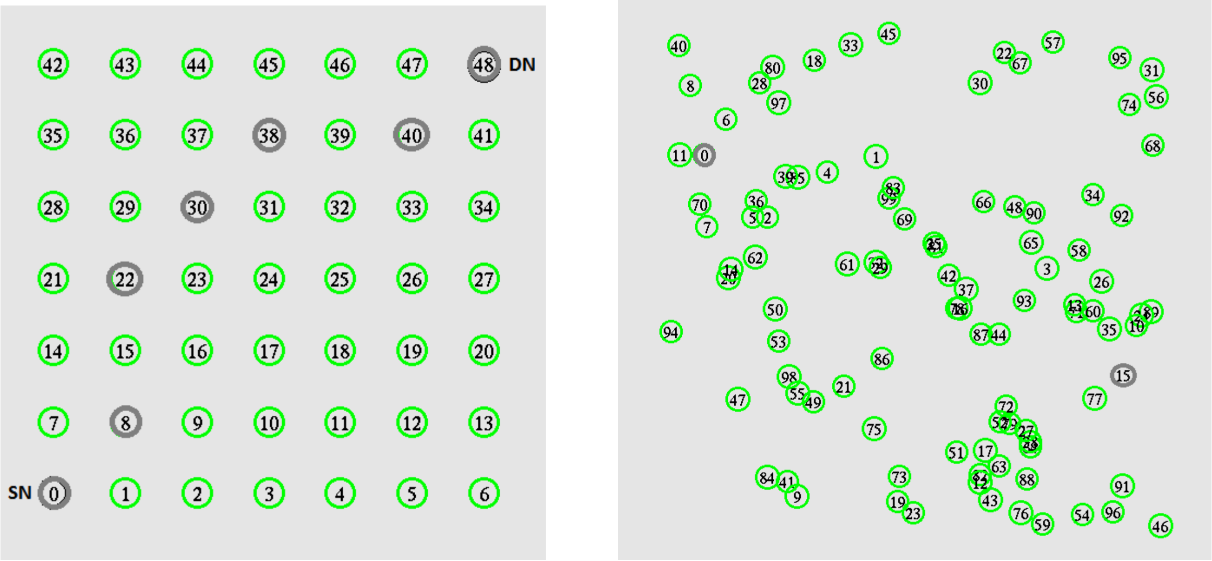
**Selection operator.** The best individual found in the binary tournament selection is placed in the mating pool. The binary tournament selection is applied to all population in two rounds, so that each individual participates at most twice in the two tournaments. Therefore, the offspring is created from the best individuals (parents) which are chosen 'only' from the mating pool.

**Crossover operator.** A single-point crossover with a crossover probability $P_C$ was considered. The algorithm first checks the crossover probability to determine if crossover will take place. Then, the algorithm tries to obtain a crossover node by randomly selecting a position after the source and prior to the destination node, taking into account the size of the shortest path (see Figure 4.3). If no common node exists, a crossover point is randomly selected. If the crossover node is common between the parents, there is a high probability of producing a valid offspring.

**Mutation operator.** The algorithm checks the mutation probability $P_m$ to determine if a mutation shall take place. Then, the algorithm randomly obtains a mutation position after the source and prior to the destination node. The segment between the source node *SN* and the node at the mutation position is kept, and the BFS algorithm is applied to find valid nodes until the destination node. So, the algorithm enables population diversity.

## 4.3 Simulation model

The simulation model evaluation objective is twofold. On the one hand, a simulation with NS-2 was performed aiming at obtaining delay and ETX values for each edge (or link) in the network. Besides that, simulations with DSR and HTLC-MeDSR were also performed, and the total delay and ETX values of the paths used by the algorithms were obtained, as will be explained in section 4.4. On the other hand, the proposed SPEA was evaluated.



(a) Scenario 1

(b) Scenario 2

Figure 4.4: The simulation scenarios

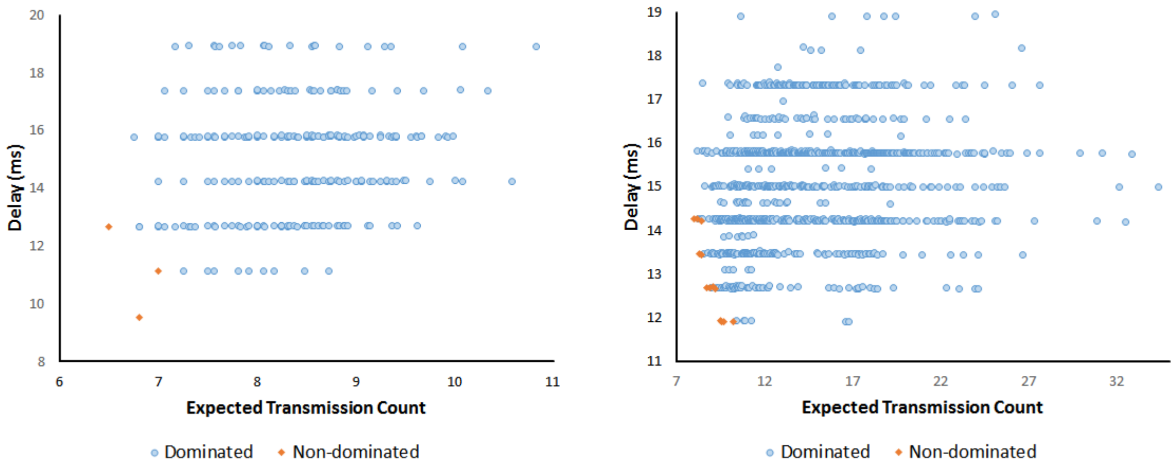
### 4.3.1 Network simulator setup

A simulation model based on NS-2 was used together with two different network scenarios (see Figure 4.4). Scenario 1 (S1) was a grid network with 49 nodes regularly distributed over an area of 500 m x 500 m. In scenario 2 (S2), 100 nodes were randomly distributed over an area of 700 m x 700 m. Here, only static scenarios were considered since they are more common in WMSNs [MPG15]. There was only one source-destination pair which was placed at opposite corners. The simulation duration was set to 200 s. Ten simulation runs were executed and the results statistically analyzed. In order to get non-deterministic results across runs, different seeds were used. One simulation traffic source was used: Constant Bit Rate (CBR) at 128 Kbps. The packet size was set to 2312 bytes. The mobile host's channel capacity varied between 6, 12, 24 and 54 Mbps depending on the distance between the nodes, as for any Wi-Fi link. Higher transmission rates correspond to shorter distances. The transmission range was the same for all transmitters. The IEEE 802.11 DCF was used for Wireless LANs, and finally, the 802.11Ext [CSEJ$^+$07b] was considered as NS-2's MAC protocol.

Table 4.1 shows the simulation parameters.

| | Scenario 1 | Scenario 2 |
|---|---|---|
| Network field | 550 m x 550 m | 700 m x 700 m |
| Number of Sensor | 49 | 100 |
| Number of Sinks/Number of Sources | 1/1 | |
| Source Node (SN) | 0 | 0 |
| Destination/Sink Node (DN) | 48 | 15 |
| Packet Size | 2312 bytes | |
| Radio Propagation Model | Two Ray Ground | |
| Source Data rate | 128 Kbps | |
| Traffic type | CBR | |
| MAC Layer IEEE | 802.11a | |
| Physical Layer data rate | 6, 12 Mbps | 6, 12, 24, 54 Mbps |
| Simulation time | 200 seconds | |

Table 4.1: Simulation parameters



(a) Scenario 1

(b) Scenario 2

Figure 4.5: Non-dominated Sets

### 4.3.2 SPEA setup

Table 4.2 shows the SPEA parameters. Fifteen runs of the proposed SPEA were performed.

| | Scenario 1 | Scenario 2 |
|---|---|---|
| Population Size | 100 | 200 |
| External Population Size | 3 | |
| Number of Generations | 20 | 50 |
| Crossover Probability ($P_C$) | 0.75 | |
| Mutation Probability ($P_m$) | 0.2 | |

Table 4.2: SPEA parameters

## 4.4 Simulation results

The proposed algorithm was compared with two routing protocols, respectively:

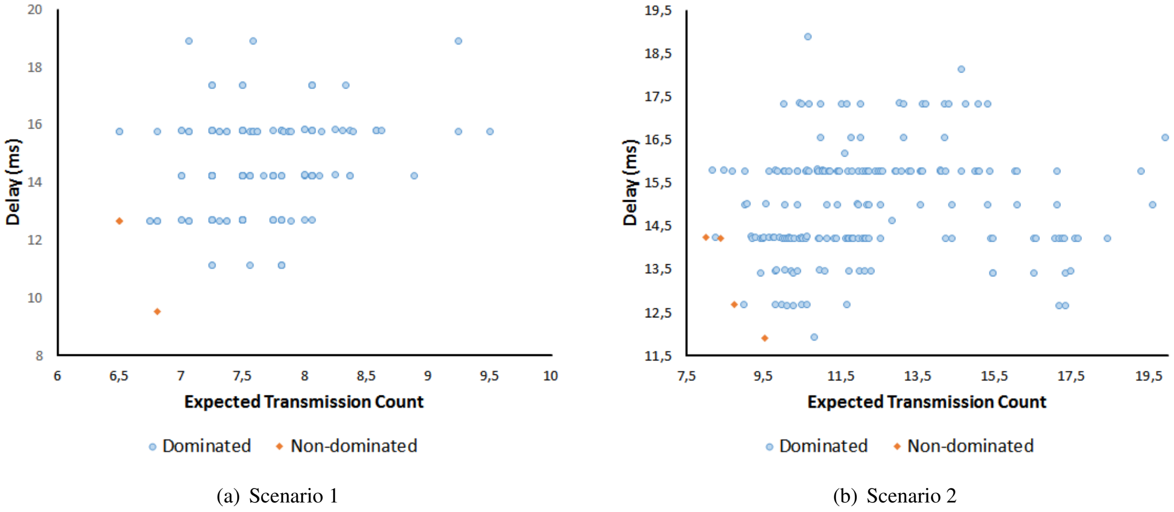
- *DSR*: Dynamic Source Routing that is a single path routing protocol and suffices for low data rates applications.
- *HTLC-MeDSR*: High Throughput Low Coupling Multipath Extension to the Dynamic Source Routing (HTLC-MeDSR) protocol which is an energy efficient multipath routing protocol that uses ETX and CF.

### 4.4.1 Non-dominated sets

Figure 4.5 presents all solutions (dominated and non-dominated) obtained over fifteen runs of the proposed algorithm on both scenarios. In S1, 80% of the non-dominated solutions (see Figure 4.5(a)) presented the following pairs of values: (6.5, 12.66 ms) and (6.81, 9.54 ms) for (ETX, Delay), respectively. In S2, the non-dominated sets were close but not overlapping. This was due to the increase in network dimension and the use of a random network topology in comparison to the grid topology of S1.

| Data rates (Mbps) | Txtime (ms) |
|---|---|
| 6 | 3.15 |
| 12 | 1.59 |
| 24 | 0.80 |
| 54 | 0.41 |

Table 4.3: Maximum transmission times



(a) Scenario 1

(b) Scenario 2

Figure 4.6: Pareto Optimal Sets

### 4.4.2 Pareto-optimal sets

Table 4.3 shows the maximum amount of time necessary to transmit a data packet among nodes, for each one of the physical layer data rates presented on Table 4.1. The end-to-end delay values presented on Figure 4.6 and Table 4.4 are discrete because of all possible combinations of transmission time values presented on Table 4.4.

| Scenario | Solutions | Objective Functions | |
|---|---|---|---|
| | | ETX | Delay (ms) |
| 1 | [0,2,10,18,26,34,48] | 6.5 | 12.66 |
| | [0,8,16,18,26,34,48] | 6.5 | 12.66 |
| | [0,8,16,18,32,40,48] | 6.5 | 12.66 |
| | [0,2,10,18,32,40,48] | 6.5 | 12.66 |
| | [0,8,16,24,32,40,48] | 6.81 | 9.54 |
| 2 | [0,5,4,99,81,37,71,15] | 8.0 | 14.25 |
| | [0,5,61,81,37,71,15] | 8.39 | 14.22 |
| | [0,36,4,99,81,37,71,15] | 8.73 | 12.69 |
| | [0,5,61,32,42,44,71,15] | 9.54 | 11.9 |

Table 4.4: Optimal solutions for the proposed SPEA

The Pareto-optimal Sets (POSs) for S1 and S2 are composed of 5 and 4 optimal solutions, respectively (see Figure 4.6). The corresponding ETX and end-to-end delay values of all optimal solutions are listed in Table 4.4. In terms of the multi-objective optimization, two distinct goals should be considered: (1) solutions should be as close as possible to the POS, and (2) solutions should be as diverse as possible in the obtained non-dominated set. In terms of the former goal, the set of non-dominated solutions presented were the best among all runs of the SPEA algorithm whose goal was to minimize both objectives (delay and ETX). Figure 4.6 shows that solutions from S2 are better distributed across the POS in comparison to those of S1. Despite this, both sets of solutions satisfy the latter goal.

### 4.4.3 QoS routing

A WMSN routing protocol must take into account a set of end-to-end QoS metrics like bandwidth, delay, packet loss, etc. The QoS requirements can be met if one or multiple paths are used.

For a single path routing protocol, like DSR, whose objective is to discover paths on-demand from a source node to a destination node, hop-count is the only metric taken into account when a path is to be chosen among those discovered. Since no additional information is available to DSR, the path with the least number of nodes is selected and used to forward data. Hop-count is a simple metric because it does not require additional information to be collected and/or maintained by the nodes.

HTLC-MeDSR attempts to use multiple paths between a given source-destination pair, if they exist. If the paths are carefully selected, the use of multiple path has advantages such as fault tolerance, load balancing and data aggregation. During the route selection process, HTLC-MeDSR takes into account ETX and CF among paths.

Table 4.5 shows average objective function values over 10 simulation runs with 95% confidence interval values. The objective function values for the proposed algorithm were averaged from the 4 non-dominated solutions obtained. It can be seen that the proposed SPEA outperforms DSR and HTLC-MeDSR in both metrics for the considered scenarios.

| | | Objective Functions | | Confidence Intervals | |
|---|---|---|---|---|---|
| | Protocol | ETX | Delay (ms) | ETX | Delay |
| Scenario 1 | DSR | N/A | 313.92 | N/A | 0.359 |
| | HTLC-MeDSR | 12.24 | 52.82 | 0.76 | 0.020 |
| | Proposed SPEA | 5.56 | 12.04 | 0.12 | 0.001 |
| Scenario 2 | DSR | N/A | 689.43 | N/A | 0.571 |
| | HTLC-MeDSR | 19.08 | 43.77 | 5.94 | 0.013 |
| | Proposed SPEA | 8.88 | 13.27 | 1.47 | 0.001 |

Table 4.5: Average objective function values with 95% confidence interval

DSR uses only one path between the source and destination nodes in both scenarios. If a node fails to send a packet a certain number of times, it discards the packet and considers that the next hop node is no longer available. Since DSR does not have any mechanism to distinguish collision losses from mobility-induced errors, and a route maintenance procedure is initiated which penalizes the packet's end-to-end delay, as can be seen on Table 4.5.

On the other hand, HTLC-MeDSR uses two independent paths to forward traffic. Differently from DSR, HTLC-MeDSR uses probe packets to identify routing failures, which reduces packet losses and consequently unnecessary route maintenance operations. As only one source rate was considered, contention and collisions are the main reason for the end-to-end delay values presented on Table 4.5.

Table 4.6 shows ETX and end-to-end delay improvement ratios between the proposed SPEA and the other routing protocols (DSR and HTLC-MeDSR) for S1 and S2. As can be seen, the proposed SPEA is 2.2 and 2.15 times better in terms of ETX than HTLC-MeDSR for S1 and S2, respectively. In terms of end-to-end delay, the proposed SPEA is 26.07 and 51.95 times better than DSR, and 4.39 and 3.3 times better than HTLC-MeDSR, for

S1 and S2, respectively.

| | Protocol | ETX | Delay |
|---|---|---|---|
| Scenario 1 | DSR | N/A | 26.07 |
| | HTLC-MeDSR | 2.2 | 4.39 |
| Scenario 2 | DSR | N/A | 51.95 |
| | HTLC-MeDSR | 2.15 | 3.30 |

Table 4.6: Proposed SPEA improvement ratio for both scenarios

### 4.4.4 Discussion

It is important to mention that, the improvement ratios presented, in the previous sections, were obtained comparing simulation results of DSR and HTLC-MeDSR with the result of an optimization algorithm (i.e., an ideal situation), for both scenarios. Even if any of the optimal solutions found by the MOEA algorithm was used, due to contentions and collisions, a regular routing protocol such as DSR would discard these optimal routes and look for new ones during the route maintenance operation.

As mentioned before, HTLC-MeDSR incorporates a set of mechanisms, such as the use of probe packets and different short retry limits at the MAC layer, and uses of ETX and CF to find high throughput paths with low route coupling among them at routing layer. During route discovery, the protocol collects network information and uses it to build a graph which is later on used by a path finding algorithm, like Dijkstra. It is assumed in Chapter 3 that the destination node is powerful enough to run this type of algorithms. Thus, the proposed MOEA could be incorporated on HTLC-MeDSR route set's building procedure to find the optimal routes from the network topology graph.

Steps 2 (population initialization) and 4 (genetic operators) in our MOEA algorithm (in Section 4.2.3) use also a path finding algorithm, e.g., BFS. In Step 2, BFS is used to create a set of individuals that are used later to create the initial population. The path finding algorithms ensure that the created individuals are feasible, by creating them using edges that were provided by the neighborhood information that was collected during the route discovery process. Otherwise, if nodes were randomly selected to create the initial population, more generations would be necessary for the algorithm to converge, since many non-existing edges could have been created. Another reason is that a fully connected network is not considered in this paper.

It was mentioned that due to the BFS's time complexity, the graph's branches were probabilistically explored to reduce the time necessary to create the set of individuals. It was noticed that the MOEA algorithm would converge to a local solution instead of a global one, if the population set was not composed of most of the source nodes direct neighbors. Thus, it is desirable that the algorithm that creates the set of individuals, explores branches composed by the source node's direct neighbors, even if a probabilistic approach is used.

It was also previously stated that WMSN applications have different QoS requirements such as, bounded latency or delay, throughput, jitter, availability and energy consumption. The proposed approach uses two objective functions: delay and ETX. ETX allows finding high throughput paths taking into account the effects of link loss ratios, asymmetry in the loss ratios between the two directions of each link, and the interference among the successive links of a path. If the number of objectives were increased, what would happen in this case would be the increase in the problem's complexity, as the search space changes from bi-dimensional to tridimensional, increasing also the number of candidate solutions to be analyzed. But for the problem in hand, delay and ETX are the most relevant objectives. However, other pairs of objectives could have been selected which would have kept the algorithm's complexity similar.

In this chapter, only static simulation scenarios were considered. Nodes' mobility still presents a considerable challenge for most WMSN routing protocols, since nodes get more often unreachable, rendering paths useless. A way of addressing the nodes' mobility issue is to use a store, carry and forward approach more common in DTNs. The use of MOEA algorithms in DTNs is still an open research area.

## 4.5 Summary

A multi-objective optimization algorithm aims at producing (1) solutions as close as possible to the POS, and (2) solutions as diverse as possible in the obtained non-dominated set. This diverse set of optimal solutions expresses trade-offs between different objectives. Routing protocols for WMSNs must take into account a set of QoS requirements like bandwidth, delay and energy consumption. The ETX metric allows finding high throughput paths on a multi-hop wireless network, and incorporates the effects of link loss ratios, asymmetry in the loss ratios between the two directions of each link, and the interference among the successive links of a path.

In the route selection process, DSR only considers hop count, not meeting WMSN QoS requirements. Hop count does not consider the path's links conditions. HTLC-MeDSR uses ETX during the path selection process to evaluate link conditions along the path. Since HTLC-MeDSR is a multipath routing protocol, it also considers the Correlation Factor during the path selection process, to find multiple paths with minimum cross interference among a source-destination pair. The proposed MOEA algorithm allows finding paths that minimize both objectives, and paths that express trade-offs among objectives.

The paths found by the proposed SPEA present 2.2 and 2.15 times less ETX than those found and used by HTLC-MeDSR for S1 and S2, respectively. In terms of end-to-end delay, the paths found by the proposed SPEA present 26.07 and 51.95 times less delay than the ones found and used by DSR, and 4.39 and 3.30 times less delay than the ones found and used by HTLC-MeDSR, for S1 and S2, respectively.

# Chapter 5

# Nodes' misbehavior in DTNs

Up until now, only static networks were considered. The mobility of nodes still presents a considerable challenge for most WMSN routing protocols, since nodes get more often unreachable rendering paths useless. A way of addressing the nodes' mobility issue is to use a store-carry-and-forward approach, which is more common in dynamic networks such as DTNs.

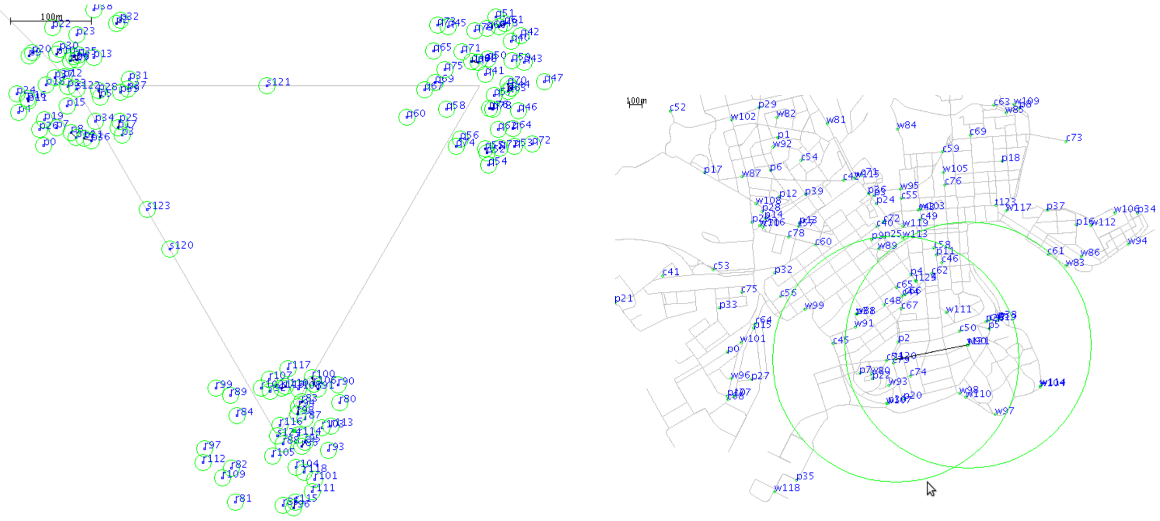
Since many previous work already addressed security aspects in static WANETs [Pat10], this chapter focuses on the problem of some nodes making limited or no contribution to the network. However, the store-carry-and-forward paradigm assumes that nodes are willing to cooperate forwarding messages of other nodes, despite connectivity issues. Misbehaving nodes consume network resources, reducing its performance and availability, therefore they constitute an important problem that should be considered. The impact of node misbehavior on seven DTN routing protocols using a large set of simulations is studied. The results show that different protocols are more resilient to different types of node misbehavior.

## 5.1 Description of the problem

A DTN relies on the fact that nodes cooperate with each other to forward messages [PVS07, KPVO11]. A node can misbehave in two ways [ALRL04]: (1) by doing content failures, i.e., delivering modified messages; and (2) by doing timing failures, i.e., delivering messages out of time (or not at all) or repeatedly. The first class of misbehavior is not considered here because it is simple to translate such faults into timing failures. The mechanism to make this translation consists simply in the sender concatenating a digital signature or a Message Authentication Code (MAC) [MOV10] to every message sent; the receiver verifies the signature/MAC and discards the message if the verification fails. This mechanism guarantees that if a node corrupts a message, it is equivalent to discarding the message (except for the resources consumed).

Three types of misbehavior were considered:

- *Type I.* Upon message reception, the node drops it.
- *Type II.* Upon message reception, the node drops it, creates ten new messages and sends them.
- *Type III.* Upon message reception, the node delays it for a quarter (1/4) or three quarters (3/4) of its remaining TTL, and for a minimum of thirty minutes.



(a) Scenario 1

(b) Scenario 2

Figure 5.1: The simulation scenarios

## 5.2 Simulation model

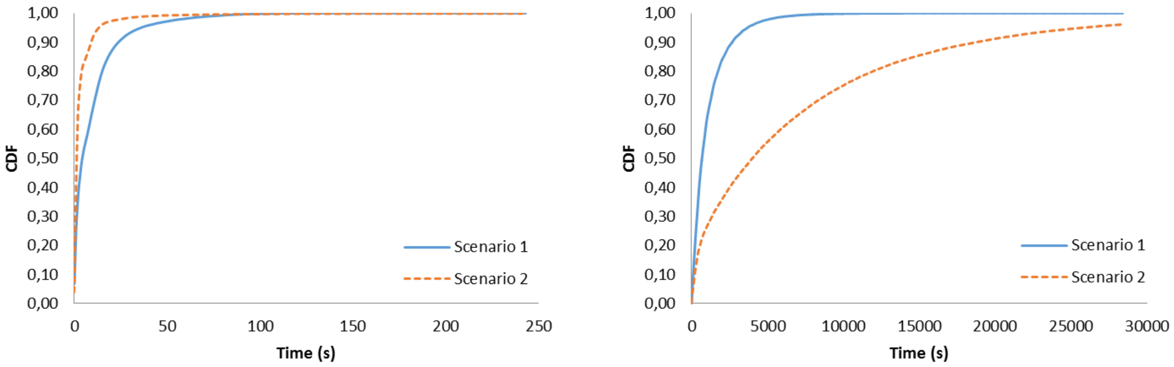
Two simulation scenarios were used in the ONE Simulator [KOK09]. Both scenarios consisted of a network with 125 nodes distributed as follows: 120 pedestrians and 5 trams for Scenario 1 (S1) (see Figure 5.1(a)), and 80 pedestrians, 40 cars and 5 trams for Scenario 2 (S2) (see Figure 5.1(b)). S1 resembles [BDD$^+$05, JFP04]. The simulation time was set to 24h with an update interval of 0.1 s. The node misbehavior was modeled as described in Section 5.1, and their effect was examined on the 8 routing protocols of Table 2.3. For Spray and Wait (SnW), $n$ was set to 6. The percentage of misbehaving nodes was varied from 20% to 80% over the total number of nodes, in steps of 20%. Misbehaving trams were not consider.

*Mobility.* For S1, the cluster based mobility model was considered with three clusters (each cluster can be a remote village). Trams, which can be message ferries [KAF12], were used to connect the clusters. Inside of each cluster, pedestrians were moving in a speed varying between 0.5 to 1.5 m/s and between the clusters, trams were moving at a speed varying from 3 to 5 m/s. Each time a tram reaches its destination, it pauses for a time varying from ten to thirty seconds. For S2, the map-based mobility model of the Helsinki city was used over an area of 4.5 $\times$ 3.4 Km. Trams were following predefined routes, moving at a speed varying from 7 to 10 m/s with pause times varying from ten to thirty seconds. Pedestrians were moving at the same speed of S1, and cars were moving at a speed varying from 2.7 to 13.9 m/s.

*Connectivity and transmission.* Only two nodes within range can communicate with each other at a time. The communication range between the nodes is 10 m, and the communication was bi-directional at a constant transmission rate of 2 Mbits/s. Trams of S2, were also equipped with WLAN radios with 100 m radio range and a constant transmission rate of 10 Mbits/s.

*Traffic model.* Every 5 to 10 min, a source node randomly chosen generated one message to a randomly chosen destination. Trams did not generate messages, being only used to carry them. Nodes did not change their behavior (malicious or not) over time. The Time-to-live (TTL) attribute of each message is 5 h, and the message size varies from 100 KB to 2 MB.

*Buffer management.* In both scenarios, pedestrian and tram nodes had a buffer size for DTN traffic of 20 MB



(a) Contacts Times CDF

(b) Inter-Contacts Times CDF

Figure 5.2: Contact characteristics as function of the scenarios

and 100 MB respectively. Cars on S2 had also a buffer of 20 MB.

## 5.3 Simulation results

The following metrics were considered to evaluate the routing performance in the presence of nodes' misbehavior: delivery ratio, buffer time, hop count, latency and overhead ratio. The delivery probability is a key performance indicator of the simulation, as it tells the percentage of successfully received packets of all sent. Buffer time indicates for how long messages were queued in the node's buffers. Hop count indicates the number of nodes the packet traversed with the exception of the source node. Latency is the time for a successful message delivery. Overhead is the number of message transmissions for each delivered message.

The routing protocols considered were: Direct Delivery (DD), Epidemic, First Contact (FC), MaxProp, Rapid, Prophet, Spray and Wait Binary (SnWBinary) and Spray and Wait Normal (SnWNormal). Please, note that the routing protocols used in the following section were revised in Section 2.2.3.

Thirty independent simulations runs using different seeds for each protocol-percentage pairs were performed, and the results were averaged. Simulations usually run much faster than in real-time. It was observed that the mean simulation speeds ranged from 90:1 to 300:1 (i.e., ranges from 3 to 15 min per simulation), depending on the routing protocol and the percentage of misbehaving nodes; only Rapid was slower (as low as 30:1 and less), taking almost 9 h per simulation. The values obtained are presented in graphs of the following subsections with 95% confidence intervals. For cases where there are large differences in values, a logarithmic scale is used in the ordinate axis.

### 5.3.1 Contact Characteristics

To characterize the impact of the mobility models in both scenarios, the contact times and the inter-contact durations between nodes were analyzed. Contact times correspond to how long the nodes were in communication range of each other. Inter-contact times correspond to the time between the end of the previous contact and the beginning of a new contact between two nodes.

Figure 5.2 shows the Cumulative Distribution Function (CDF) of the contact times and inter-contact durations

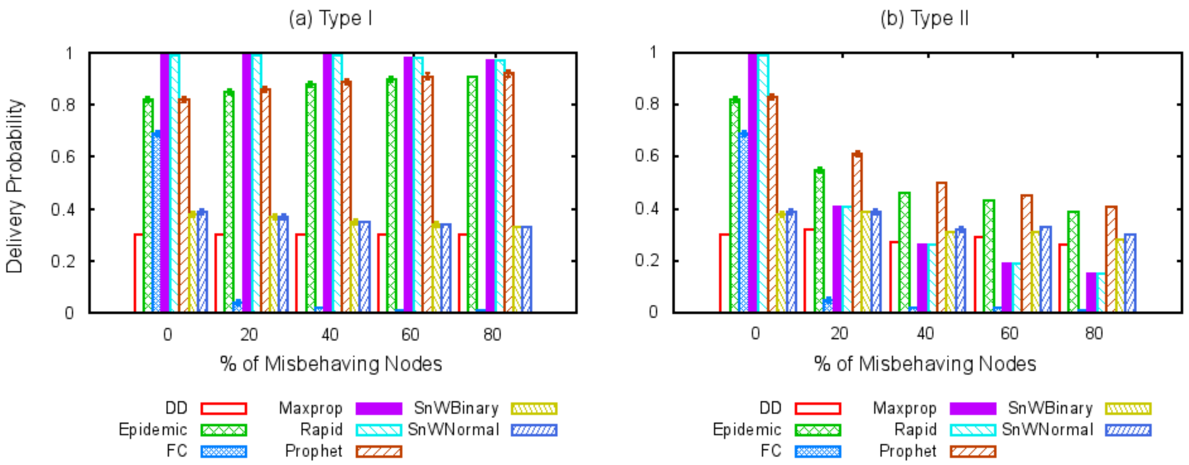


Figure 5.3: Average delivery probability as function of the percentage of Types I and II misbehaving nodes for S1

between nodes in both scenarios.

S2 is characterized by short contacts as 90% of the contacts last less than 8 seconds. For S1, 90% of the contacts last less than 22 seconds, lasting 3 times more than those of S2. If 90% of the inter-contact durations were considered, the ones from S2 last 7 times more than those from S1. This happens due to different node density and mobility patterns between the scenarios.

## 5.3.2 Misbehavior of Types I and II for S1

### Average message delivery probability

Figure 5.3 shows the average message delivery probability for all protocols for two types of misbehavior. DD is not affected by Type I misbehavior as nodes only deliver messages when they meet the message recipient. An issue that affects DD's delivery probability is the cluster scenario, since communication between clusters is made through trams. Consequently, DD's delivery probability must be below 1/3, as nodes in different clusters never meet directly.

SnW is similar to DD in that during the wait phase it performs direct transmissions. But, because of the spray phase, some messages generated to nodes in another cluster are delivered, which allows the protocol to have a delivery probability above 1/3.

First Contact forwards only one copy of a message to the first node met. Because of this, it is the most affected DTN routing protocol by misbehaving nodes as even if the network only contains 20% of misbehaving nodes, the probability of meeting a misbehaving node that drops the message is high resulting in a reduction of 94% of the delivery probability.

Misbehaving nodes cause a reduction of the number of message copies circulating in the network (i.e., they reduce network congestion) as they drop them. Protocols like Epidemic and Prophet take advantage of small percentages (below 80%) of Type I misbehaving nodes as their delivery probabilities increases with the reduction of the network traffic. Maxprop and Rapid have the best delivery probabilities for Type I misbehavior, since the replication and discarding mechanisms used by both ensure selection of the best sets of messages to transmit and remove, respectively. For Type II misbehavior, all protocols experienced reductions in the delivery probability

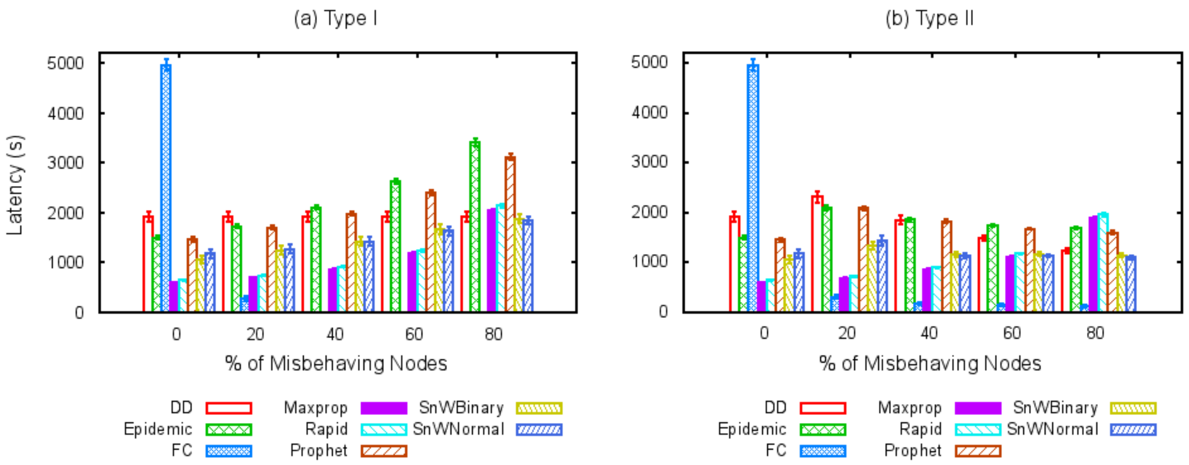


Figure 5.4: Average latency as function of the percentage of Types I and II misbehaving nodes for S1

because misbehaving nodes degraded the network conditions.

Protocols like MaxProp and Rapid, are barely affected by Type I misbehavior. But due to the additional overhead caused by Type II misbehaving nodes, their performance degrades with the increase of the percentage of misbehaving nodes.

### Average message latency

For scenarios where nodes do not misbehave, FC also presents high values of latency because of the high number of hops, as can be seen in Figure 5.4. FC is also the only protocol in which there is a reduction of latency with the increase of the percentage of misbehaving nodes. This happens because undelivered messages do not contribute to the latency statistics and fewer messages are delivered as more misbehaving nodes drop them.

For Type I misbehavior, DD's latency does not change. The average DD latency decreased in the presence of Type II misbehaving nodes, as the protocol is slightly affected by the radio usage to transfer additional messages generated by misbehaving nodes.

For Type I and Type II misbehaviors, latencies Maxprop and Rapid increased with the increase of the percentage of misbehaving nodes. Since messages travel on average a similar number of hops, the increase of misbehaving nodes caused an increase in network latency. This happens because messages with small latency values were dropped more and more by the increasing number of misbehaving nodes.

### Average message overhead

Figure 5.5 shows that Epidemic and Prophet have the highest values of overhead ratio. This happens because of the similarities between these two routing protocols. Prophet only has smaller overhead because it uses a probabilistic metric that decides if it is worth replicating a message to a contacted node.

Due to the similarities between DD and both versions of SnW, DD has zero overhead and SnW has smaller values of overhead ratio in comparison with other DTN routing protocols.

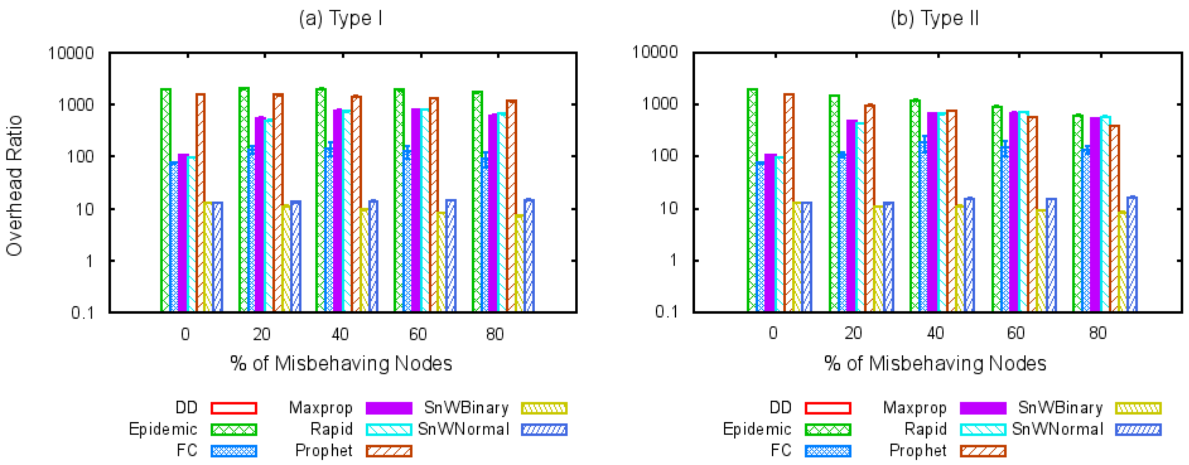


Figure 5.5: Average overhead ratio as function of the percentage of Types I and II misbehaving nodes for S1

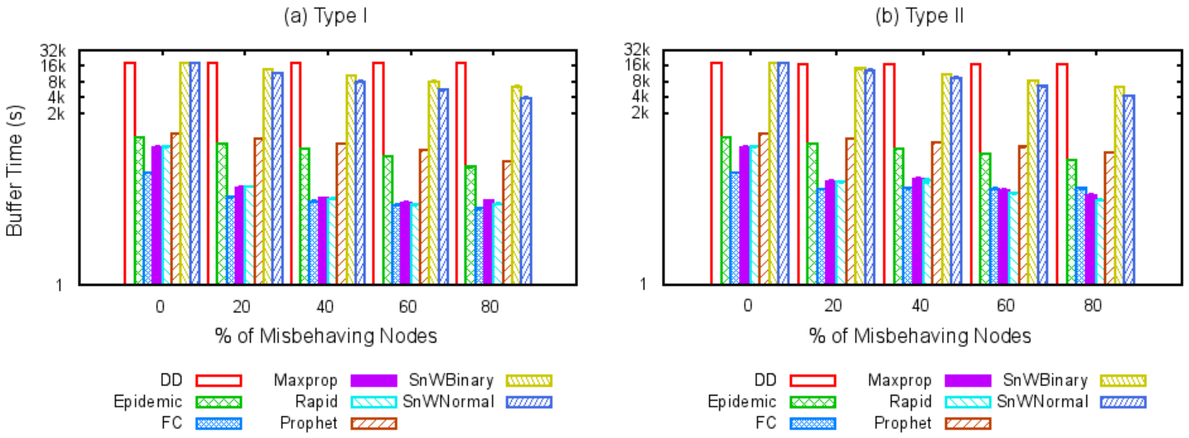


Figure 5.6: Average buffer time as function of the percentage of Types I and II misbehaving nodes for S1

### Average message buffer time

Due to the direct transmissions' approach used by DD and SnW, they present the highest values of buffer time in comparison with other protocols. For both types of misbehaving nodes, DD presents buffer time values very close to the maximum TTL (see Figure 5.6) because the protocol buffers messages until it finds the destination nodes or drops them if they expire. Nevertheless, Type II misbehavior presents a slight reduction on the buffer time with the increase of the percentage of misbehaving nodes. This happens because of the high number of messages generated by Type II misbehaving nodes caused an increase in radio usage therefore preventing other nodes in communication range from delivering their messages.

Both versions of SnW suffer more with misbehaving nodes due to the spray phase, as the source node can replicate messages to misbehaving nodes that silently drop them. As a consequence, with the increase of the percentage of misbehaving nodes, the protocol tends to reduce buffer time.

Other DTN routing protocols presented in this chapter, despite their routing algorithms, have smaller values of average buffer time as they replicate messages to encountered nodes more often, which reduces buffer time.

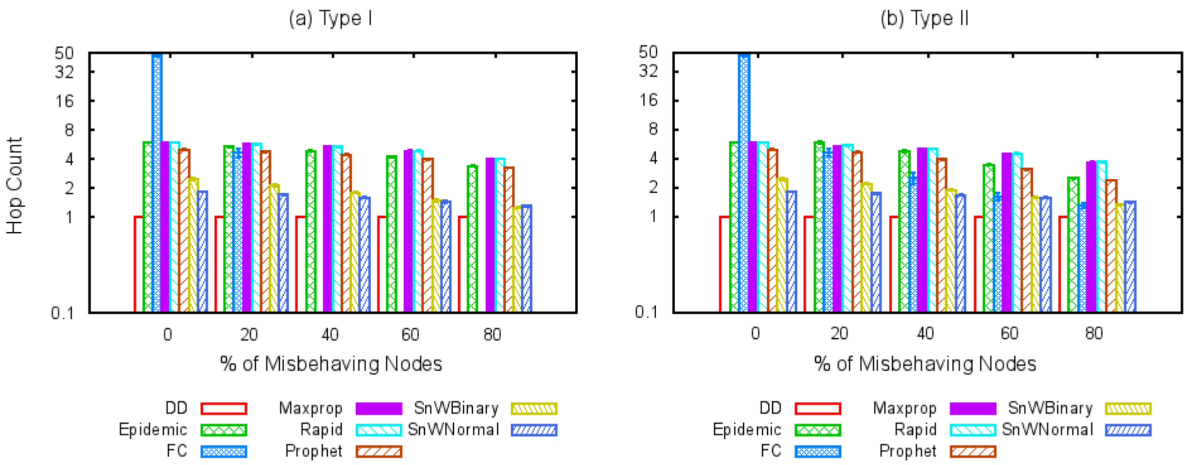


Figure 5.7: Average hop count as function of the percentage of Types I and II misbehaving nodes for S1

**Average message hop count**

Figure 5.7 shows that when nodes do not misbehave FC presents the highest hop count. This is due to the fact that FC forwards messages to the first encountered nodes and these messages are continuously forwarded until they reach the intended destination node. In general, the increase of misbehaving nodes causes a reduction in the hop count as nodes can only rely on the remaining amount of well-behaved nodes to store, carry and forward messages. As the percentage of well-behaved nodes reduces, only messages that travel fewer hops (closer to one) are delivered, unless the protocol has a mechanism of only selecting a well-behaved node as forwarder.

### 5.3.3 Misbehavior of Type I and II for S1 and S2

Figure 5.8 shows the average delivery probability as function of the percentage of Types I and II in both scenarios. DD behaves better on S2 in comparison to S1, because it has a higher possibility of delivering messages to the destination node.

SnW behaves similarly to DD in both scenarios, but it delivers more messages on S2, due to the absence of clusters (topology). Even for S1, because of the existence of two phases in the protocol, it delivers more messages (a little bit above 1/3) than DD, as it sprays multiples copies to intermediate nodes which may come in contact with the destination node. FC forwards only one copy of the message to the first node met. Because of this, it is the most affected DTN routing protocol by misbehaving nodes as even if the network only contains 20% of misbehaving nodes, the probability of meeting a misbehaving node that drops the message is high, resulting in a reduction of 93% and 81% of the delivery probability for S1 and S2, respectively.

As previosly stated, misbehaving nodes cause a reduction of the number of message copies circulating in the network as they drop them. Because nodes meet less on S2, Epidemic and Prophet deliver fewer messages in comparison with S1, but are less affected in comparison with other routing protocols by misbehaving nodes because of their unlimited copy algorithms. MaxProp and Rapid suffer more from Type I misbehavior because they had shorter contact durations, and for longer messages they were not enough to transfer the entire message.

For Type II misbehavior, all protocols experienced strong reductions in the delivery probability as misbehaving nodes degrade the network conditions by consuming additional resources like buffers, transmission time, etc.

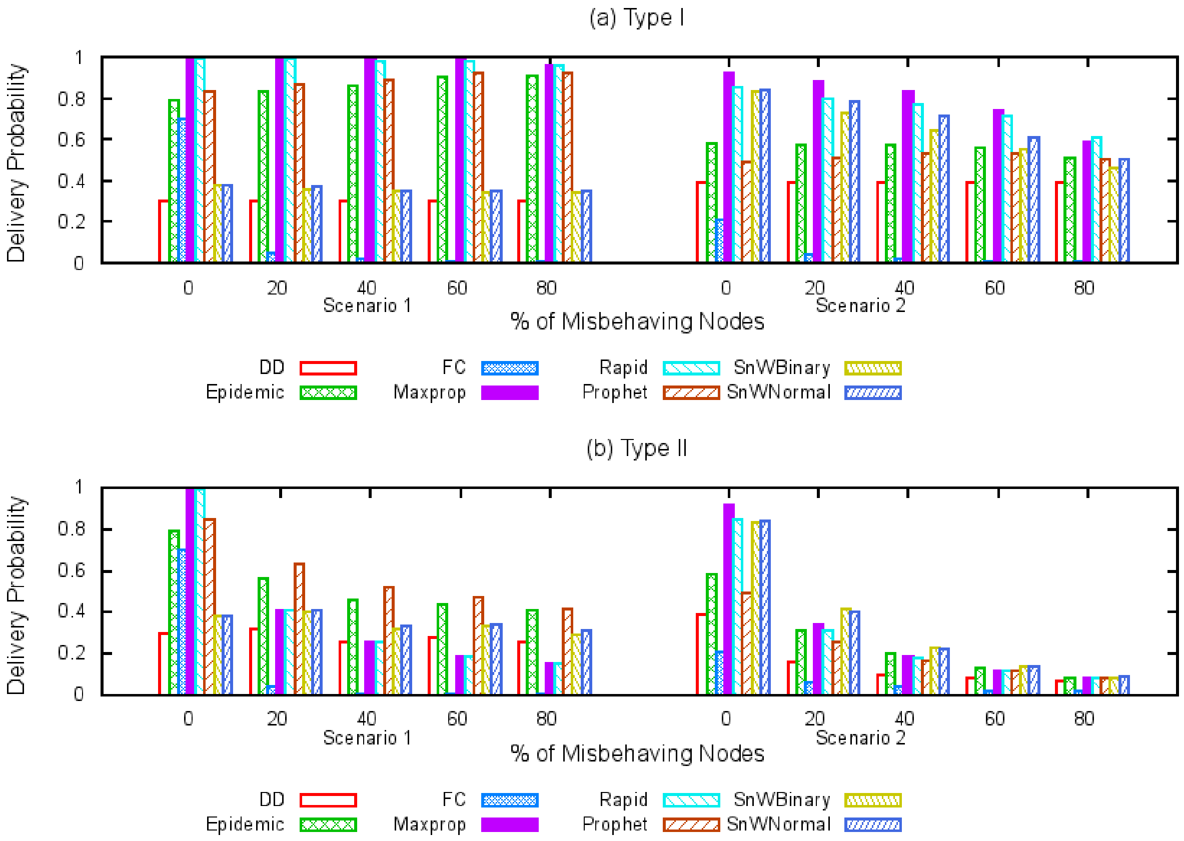


Figure 5.8: Average delivery probability as function of the percentage of Types I and II misbehaving nodes in both scenarios

### 5.3.4 Misbehavior of Type III

**Average Message Delivery Probability**

Figure 5.9 shows the average delivery probability as function of the percentage of Type III misbehaving nodes in both scenarios. Regardless of the scenario, DD is not affected by Type III misbehaving nodes due to the direct transmission's approach.

Epidemic, Prophet and FC take advantage of some malicious delay because their average delivery probability increases with the increase of the percentage of misbehaving nodes. For the cases of Epidemic and Prophet, they use the reduction of network traffic caused by the increase of the number of misbehaving nodes, as fewer packets circulate in the network. Whereas, FC takes advantage of the additional delay since messages can be forwarded elsewhere in the network (that is usually closer to the destination node). SnW's performance degrades with the increase of Type III misbehaving nodes. Between the versions, SnWBinary is the one that suffers more as it sprays half of its message copies to the following node that can delay them all.

In both scenarios, Maxprop and Rapid present the highest values of average delivery probability, for the same reasons explained in Section 5.3.2. They are followed by Epidemic and Prophet on S1, and SnW on S2.

By increasing the malicious delay time to a maximum of 90% of TTL, simulation results have shown that most protocols average delivery probability did not change with the increase of the percentage of misbehaving nodes. The only exception was FC on S1 that experienced a reduction on its average delivery probability with the increase

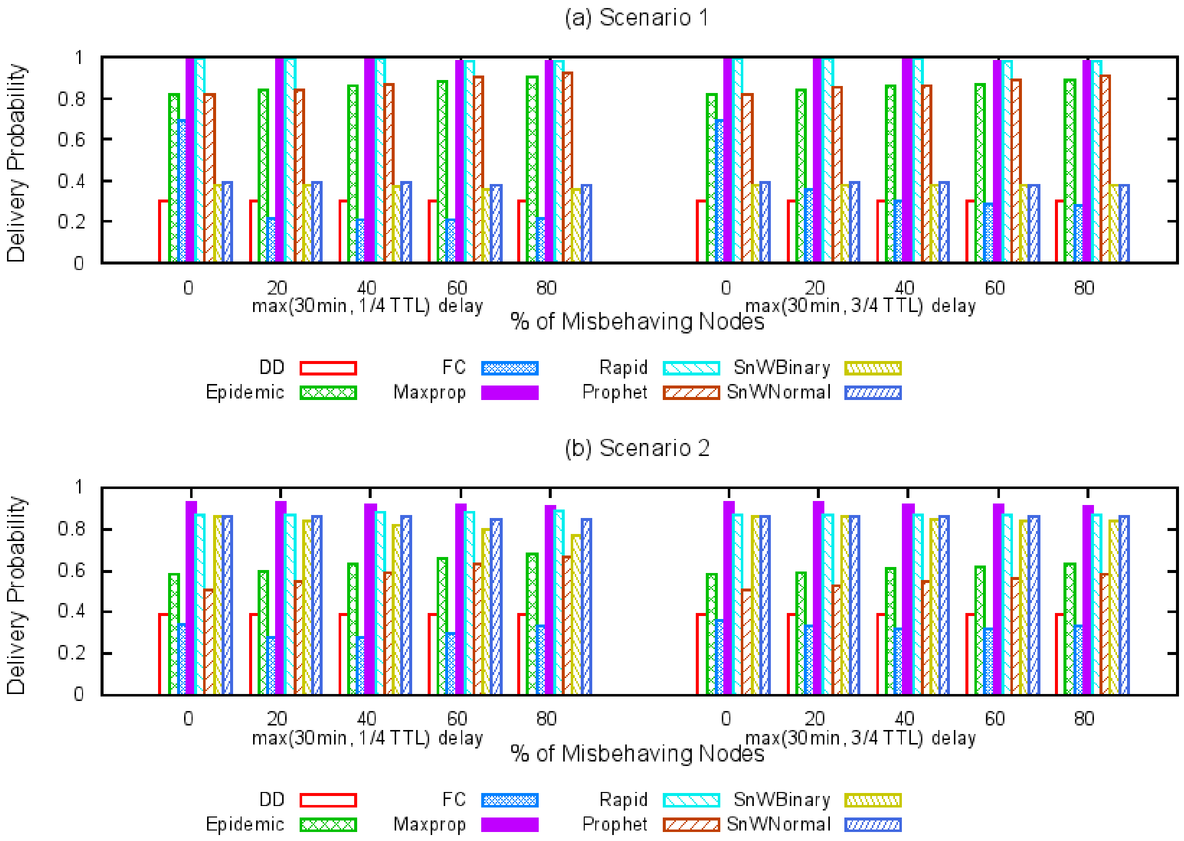


Figure 5.9: Average delivery probability as function of the percentage of Type III misbehaving nodes in both scenarios

of the percentage of Type III misbehaving nodes because most packets were discarded since their TTL expired.

### Average Message Latency

Figure 5.10 shows the average latency as function of the percentage of Type III misbehaving nodes in both scenarios. The idea is to analyze the influence of malicious message delay on the latency of delivered messages.

For S1, all the DTN routing protocols' average latencies increase with the percentage of misbehaving nodes, except DD. This happens because of the increase of the probability of the following contact be with a misbehaving node, whose sole propose is to delay messages. FC is the protocol with the second highest values of latency. This was due to the high number of intermediate nodes used to reach the destination node.

For S2, DTN routing protocols have higher average latency values because nodes on this scenario have shorter contact opportunities in comparison with those of S1. Another aspect to consider is that, with the increase of the percentage of misbehaving nodes, most DTN routing protocols experience an increase of the average latency values as messages were delayed more times before being delivered. FC presents very high values of average latency since it presented a similar behavior to DD.

### Average message overhead

Figure 5.11 shows the average overhead as function of the percentage of Type III misbehaving nodes in both Scenarios.

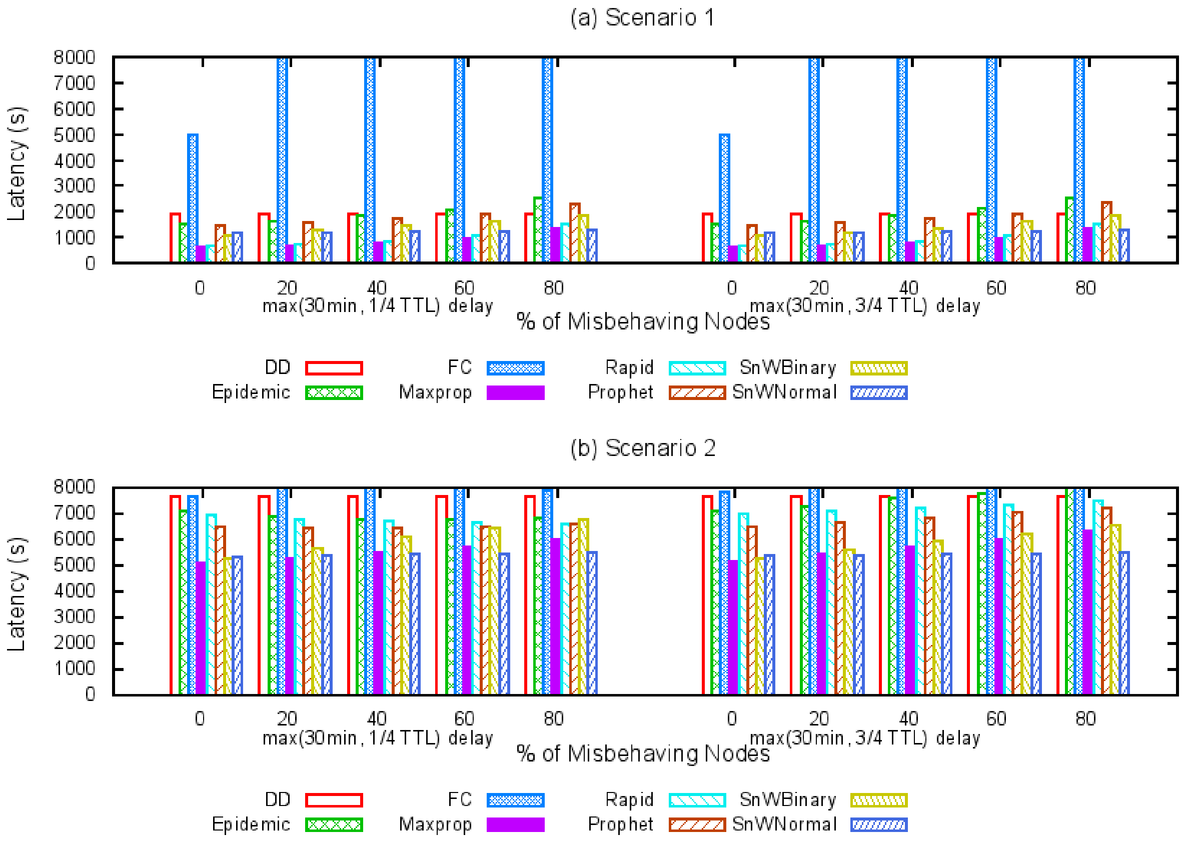


Figure 5.10: Average latency as function of the percentage of Type III misbehaving nodes in both scenarios

In both scenarios, Epidemic and Prophet presented the highest values of overhead ratio. This was also because of their similarities.

All DTN routing protocols, with exception of DD, experience reductions on the network traffic and consequently a reduction in the average overhead with the increase of the percentage of misbehaving nodes because more messages are retained at misbehaving nodes' buffers. But this reduction becomes smaller if the malicious time delay of a message is increased since messages with higher delay are dropped.

As mentioned before, routing protocols that use direct transmissions present low overhead. For instances, DD, in both scenarios, presented zero overhead meanwhile the overhead magnitude of SnW depended on the scenario used. For S1, it presented smaller values of overhead ratio, because of the topology, whereas on S2, it presented values considerable higher because it delivered more messages.

### Average message buffer time

Figure 5.12 shows the average buffer time as function of the percentage of Type III misbehaving nodes in both scenarios.

In both scenarios, DD and SnW have the highest values of buffer time for the same reasons explained in previous sections. The remaining protocols experience an increase on buffer time with the increase of the percentage of misbehaving nodes since more nodes delayed more messages. But, it was observed an increase of the average buffer time resulting from the increase of malicious message delay. However, this increase was not of the same

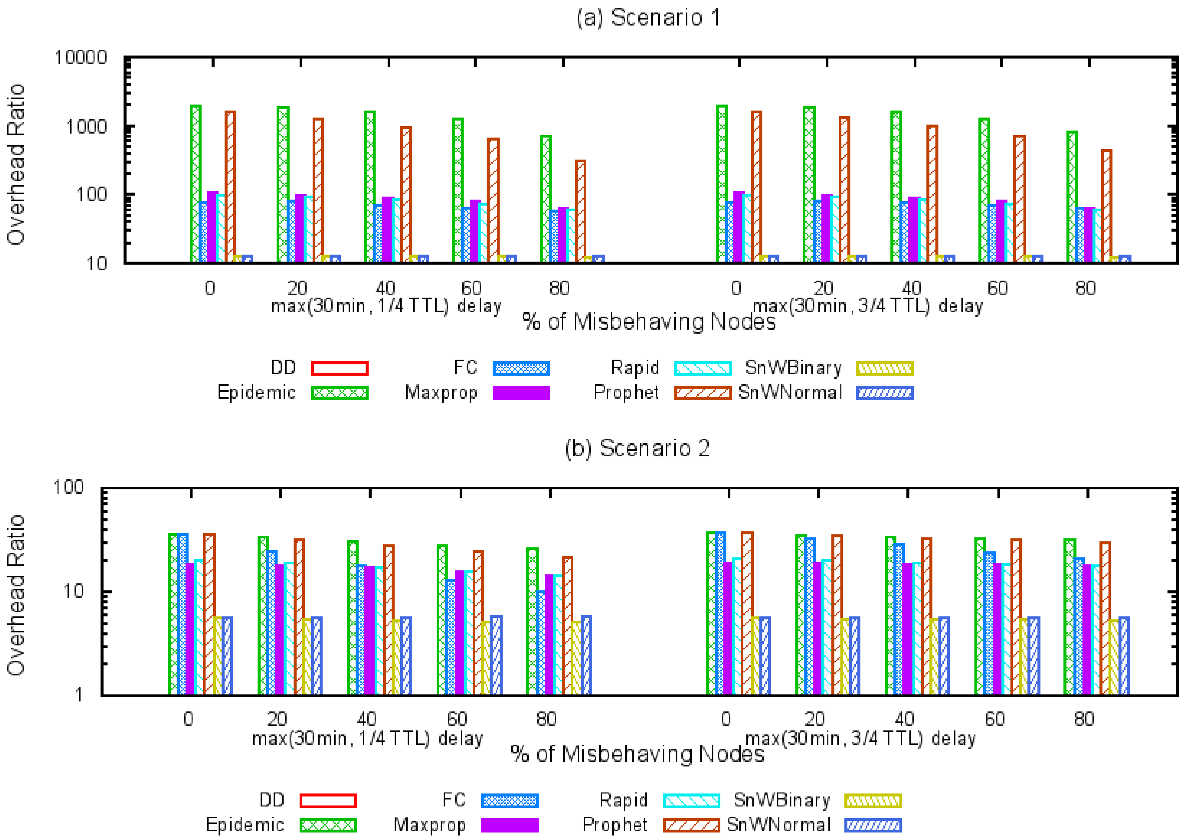


Figure 5.11: Average overhead as function of the percentage of Type III misbehaving nodes in both scenarios

order of magnitude as less messages were delivered.

## 5.4 Summary

This chapter presented a study of nodes' misbehavior in DTNs. For S1, simulations results have shown that in the presence of misbehaving nodes, Epidemic and Prophet are quite robust as they have no limits to message replication and may even benefit from the presence of misbehaving nodes since the latter may reduce network congestion. For scenarios without misbehaving nodes, Maxprop and Rapid presented the highest values of delivery probability. But, in the presence of misbehaving nodes, these protocols were strongly affected. Both versions of SnW presented low values of delivery probability because of the scenarios considered. If the source and destination nodes were located in different clusters, and unless the message was sprayed to a tram and then to a node inside the destination's cluster, the message would never reach its destination. Therefore, the delivery probabilities of DD was around 1/3. FC was the most affected routing protocol in the presence of misbehaving nodes because there was a single message copy that being relayed to a malicious node was lost forever.

An aspect to consider in both scenarios is the contact's characteristics, due to the mobility model, and the topology. S1 allowed longer contact's durations between nodes because of the mobility patterns and the cluster node density used. Nodes inside the clusters were able to deliver messages to the final destination. If a message had to be forwarded to a node in another cluster, the node in question had to rely on trams because they were the ones moving between clusters. The use of trams represents a drawback for DTN routing protocols that use

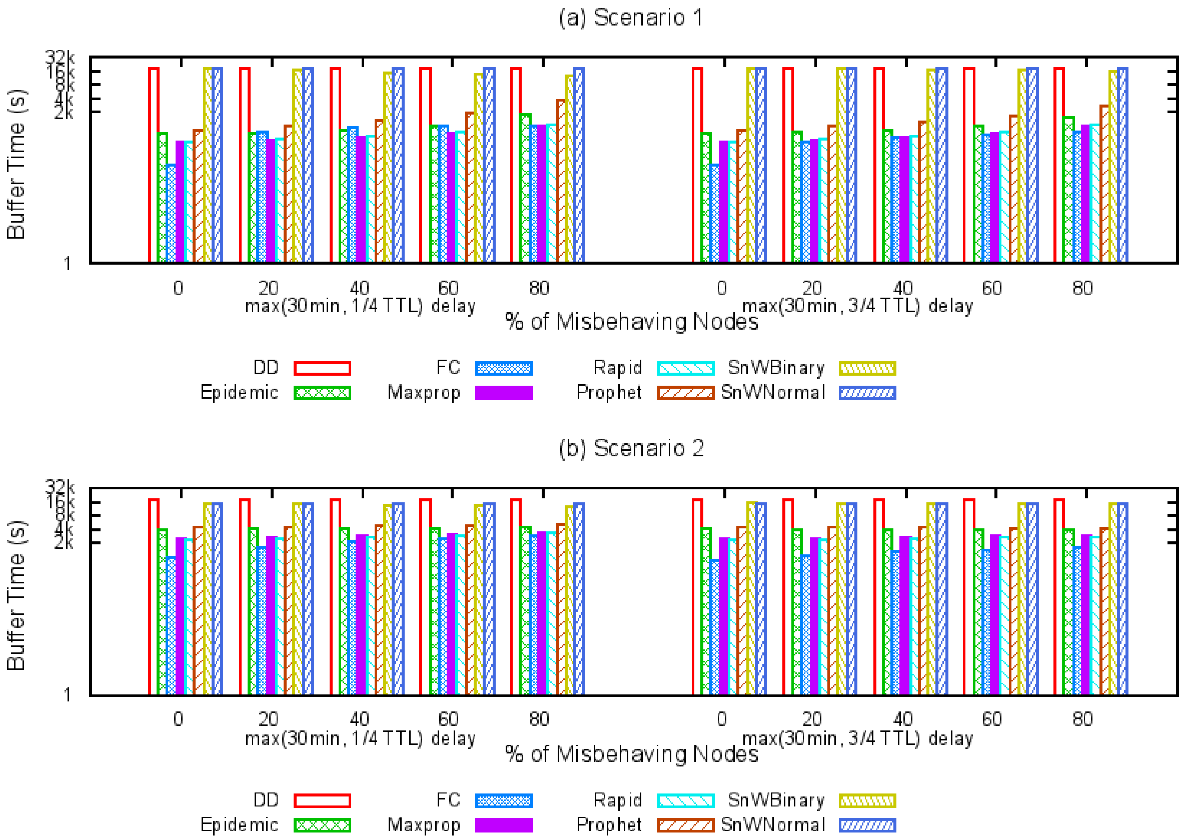


Figure 5.12: Average buffer time as function of the percentage of Type III misbehaving nodes in both scenarios

direct transmissions. S2 allowed smaller contact durations and the nodes were following predefined routes. If the destination was far away, nodes not only had to rely on trams to deliver messages but on other nodes as well.

Simulation results have shown that Type III misbehavior did not affect strongly the delivery probability of the routing protocols for the scenarios considered. The result was, in some cases, even the opposite as messages were only forwarded when the message carrier was closer to their destination, thus presenting a low communication overhead. On the contrary, the average latency increased with this kind of misbehavior. The delivery probability was not surprising since a TTL value that was reasonable but not too demanding was used (i.e., 5 hours). Figure 5.10 shows that the maximum average delay was around 2 hours (8000s). Simulation results would have been worse with a TTL near or lower than that value.

Additionally, simulations results have shown that Type I and II misbehaviors affected more the performance of DTN routing protocols than Type III misbehavior (see Figure 5.8 and Figure 5.9). Type I and II misbehaviors influenced the ability of DTN routing protocols to forward messages, as misbehaving nodes dropped them, or even for the case of Type II, by not allowing other nodes to transmit useful information, as misbehaving nodes excessively used the wireless medium. The influence of Type III misbehaving nodes was greatly affected by the buffer sizes at each node, and by the amount of time that messages were delayed. For example, if nodes with small sized buffers are considered, messages from other nodes would be dropped due to buffer overload or while creating space for new messages.

The main conclusion of this chapter is that the delivery probability of DTN routing protocols depends on two factors: (1) the contact characteristics provided by the mobility model and the topology; (2) the type of misbehavior. With Type I misbehavior, Maxprop and Rapid presented the best results in both scenarios followed by Epidemic and Prophet. With Type II misbehavior, Prophet and Epidemic were the best for S1, and both versions of SnW were the best for S2. With Type III misbehavior, Maxprop and Rapid were the best followed by Epidemic and Prophet. Moreover, FC was the routing protocol most affected by misbehavior because it is single-copy.

An interesting question is: “What characterizes the resilience of the routing protocol to nodes’ misbehavior?”. The routing protocols considered were classified in terms of two metrics, “*-copy” and “estimation-based” (cf. Table 2.3). In terms of the first, clearly the best protocols were the unlimited-copy ones’ and the worst was FC that is a single-copy, with SnW ($n$-copy) in the middle. In relation to the second metric, estimation-based protocols are apparently better. Epidemic is not estimation-based and fares well, but it is also a brute-force protocol that does flooding, therefore having the highest overhead.

# Chapter 6

# A Robust and distributed reputation system for Delay-Tolerant Networks

Distributed reputation systems can be used to foster nodes cooperation in decentralized and self-managed systems due to the nonexistence of a central entity. In this chapter, a robust and distributed reputation system for Delay-Tolerant Networks is presented. The system is robust because despite taking into account first- and second-hand information, it is resilient against false accusations and praise, and distributed, as the decision to interact with another node depends entirely on each node. The proposed system is composed of reputation, trust and routing decision modules. The reputation module collects first-hand information (i.e., evidence through direct communication between two nodes, that is, through experience) and evaluates each node's reputation. The trust module integrates second-hand information (i.e., evidences through other nodes recommendations) and evaluates each node's trust. The routing decision module uses maximum likelihood or optimal Bayesian decision criteria to classify nodes.

This chapter also shows that there are trade-offs in such systems, for example, to reduce detection time, one may sacrifice robustness, or even, by penalizing nodes that do not forward messages, one may temporarily isolate nodes that could contribute to message forwarding.

## 6.1 Assumptions and notations

This section introduces assumptions and notations that will be used throughout this chapter.

**Scenario.** There is a network with several nodes, i.e., wireless devices held by people or in vehicles that may be moving. Messages are forwarded in a store-carry-and-forward manner that requires the existence of a suitable carrier to forward them until the destination is found or they are discarded due to, for example, time-to-live (TTL) expiration.

**Node capability.** Each node has a Unique IDentifier (UID) that cannot be spoofed. Each node can only monitor its one-hop neighbors, i.e., can only monitor nodes that are directly connected to him. Upon an encounter between two nodes, they have a way of deciding if it was satisfactory.

**Attack model.** Akin to benign nodes, malicious nodes are also wireless devices that deviate from the protocol in the following ways: (*i*) *lying attacks (liars),* nodes that not having received a message return wrong confirmation that they have it in their buffer. In addition, these nodes may disseminate false first-hand information; (*ii*) *black-hole attacks*, nodes that do not forward others' messages; (*iii*) *sleeper attacks*, where a node behaves accurately for some time to create a good reputation for itself and then starts misbehaving; and (*iv*) *collusion attacks*, where nodes may forward data to each other to earn reputation. The intensity of individual attacks can be augmented by collusion. In addition, active attacks, which are characterized by an unauthorized party modifying the contents of the message, are not considered because wireless network's cryptographic techniques perform well under active attacks.

In order to facilitate future references, frequently used notations in this chapter are listed below in Table 6.1.

| Notation | Meaning |
|---|---|
| $\theta$ | The state of nature |
| $x$ | An observation |
| $\alpha, \beta$ | The parameters of the Beta probability density function |
| $y_\eta^\tau$ | The accumulated rating with aging of a given node at time period $\tau$ |
| $\eta$ | The longevity factor |
| $\mathcal{F}_{x_{\text{ij}}}$ | The first-hand information |
| $\mathcal{S}_{x_{ij,k}}$ | The second-hand information |
| $\mathcal{R}_{\text{ij}}$ | The reputation rating |
| $\phi$ | The smothing factor |
| $\varsigma$ | The individuality factor |
| $d$ | The deviation threshold |
| $\mathcal{T}_{\text{ij}}$ | Trust rating |
| $\pi(\theta)$ | The Prior information |
| $l(x)$ | The likelihood ratio |
| $t$ | The decision threshold |

Table 6.1: The most used notations in this chapter

## 6.2 Bayesian decision theory

Statistical decision theory [Ber13] aims at providing a rational framework for dealing with situations where decisions have to be made under uncertainty. The Bayesian approach, which is a particular way of formulating and dealing with statistical decision problems, offers a method of formalizing *prior* beliefs and of combining them with available observations, with the goal of allowing a formal derivation of optimal decision criteria.

Some fundamental concepts of the theory of Bayesian decision making [Mur12] are:

- All that is unknown but relevant for making a decision (also called *state of nature*) is represented by $\theta$ and takes values on a state space $\Theta$. The available knowledge about $\theta$, *prior* or *a priori*, is characterized by its probability function $\pi(\theta)$. Hereafter, it is considered that $\Theta$ is discrete.

- The *observations*, $x$, which are used to make decisions, are most likely random depending on $\theta$. According to probability theory, this dependence is expressed assuming that $x$ is a sample of a random variable $X \in \mathcal{X}$

whose probability function is conditioned on the true $\theta$, i.e., $f\left(x|\theta\right)$ that is also known as *observation model* or *likelihood function*.

- A *decision rule* $\delta(x)$ has to choose an action amongst a set $\mathcal{A}$ of allowed decisions or actions. $\delta(x)$ is a function from $\mathcal{X} \rightarrow \mathcal{A}$ thus specifying how actions or decisions are chosen given $x$. $\mathcal{D}$ is the set of allowed decision rules.
- A *loss function* $L\left(\theta, a\right) : \Theta \times \mathcal{A} \rightarrow \mathbb{R}$, specifies the cost incurred if the true state of nature is $\theta$ and the chosen decision is $a$, thus quantifying the consequences of the decisions.

A Bayesian decision problem can be formalized by the set of elements $\{\Theta,\ \pi\left(\theta\right),\ \mathcal{A},\ \mathcal{X}, L\left(\theta, a\right), \mathcal{D},\ f\left(x|\theta\right)\}$ and is considered solved if a decision rule $\delta(x)$ is chosen in a way to obtain some kind of optimality criterion that is associated with the loss function.

In Bayesian decision theory, the *posterior* or *a posteriori* expected loss, conditioned on observation $x$, is defined as

$$\rho\left(\pi\left(\theta\right), a|x\right) = \mathbb{E}\left[L\left(\theta, a\right)|x\right] = \sum_{\theta \in \Theta} \pi\left(\theta|x\right) L\left(\theta, a\right) \tag{6.1}$$

whereas via Bayes law,

$$\pi\left(\theta|x\right) = \frac{f\left(x|\theta\right)\pi\left(\theta\right)}{\sum_{\theta' \in \Theta} f\left(x|\theta'\right)\pi\left(\theta'\right)} \tag{6.2}$$

where $\pi\left(\theta\right)$ is the prior density of $\theta$, $\pi\left(\theta|x\right)$ is posterior density for $\theta$ given $x$, $f\left(x|\theta\right)$ is the likelihood for $\theta$ based on $x$ so that in terms of $\theta$, $posterior \propto likelihood \times prior$.

## 6.3 The robust and distributed reputation system

### 6.3.1 The Bayesian approach

Each node considers that there is a given parameter, $\theta$, such that another node misbehaves with probability $\theta$, and that the outcome is drawn independently at each observation $x$. Furthermore, each node considers that there is a different $\theta$ for every other node. These parameters are unknown, hence modelled assuming that they are drawn according to $\pi(\theta)$, which is updated as new observations become available.

The beta probability density function, $Beta(\alpha, \beta)$, is used as the prior as it represents probability distributions of binary events (e.g., good or bad) and the conjugate is also a Beta distribution [Dav03]. The Beta density can be expressed as

$$f\left(\theta|\alpha, \beta\right) = Beta\left(\theta|\alpha, \beta\right) = \frac{\Gamma\left(\alpha + \beta\right)}{\Gamma\left(\alpha\right)\Gamma\left(\beta\right)}\theta^{\alpha - 1}(1 - \theta)^{\beta - 1} \tag{6.3}$$

where $0 \leq \theta \leq 1$ and $\alpha, \beta > 0$, and $\Gamma$ is the Euler gamma function defined as $\Gamma\left(z\right) = \int_0^\infty u^{z-1}e^{-u}du$ and is valid for any complex number $z$. The expectation of the Beta density is

$$\mathbb{E}\left[\text{Beta}\left(\theta|\alpha, \beta\right)\right] = \frac{\alpha}{\alpha + \beta} \tag{6.4}$$

The Bayesian process works as follows. Initially each node has the prior $Beta(1,1)$, that is, the uniform distribution on $[0,1]$, for all its neighbors. The $Beta(1,1)$ prior represents absence of information as there are no observations. When a new observation is made, if a correct behavior is observed then $x = 1$; otherwise $x = 0$. The prior is updated according to $\alpha_{\text{new}} = \alpha_{\text{old}} + x$ and $\beta_{\text{new}} = \beta_{\text{old}} + (1 - x)$.

Due to the network dynamics, a node may change its behavior over time in contrast to the standard Bayesian framework that gives the same weight regardless of time of occurrence of the observation. Therefore, old observations may not always be relevant for the most recent ones. A model in which old observations are given less weight than more recent ones is desirable. An idea could be to forget gradually old observations. The forgetting factor can be adjusted according to the expected speed of change in the observations. The aging scheme [JQ09] is given by

$$y_{\eta}^{\tau} = y_{\eta}^{\tau-1}\eta + y^{\tau} \tag{6.5}$$

where $y_{\eta}^{\tau}$ is the accumulated rating with aging of a given node at time period $\tau$, $y^{\tau}$ is the new rating at time $\tau$ and $\eta$ is the longevity factor and $0 \leq \eta \leq 1$.

Now, the prior is updated according to

$$\begin{aligned} \alpha_{t_{\text{ij}},\eta}^{\tau} &= \alpha_{t_{\text{ij}},\eta}^{\tau-1}\eta + \alpha_{t_{\text{ij}}}^{\tau} \\ \beta_{t_{\text{ij}},\eta}^{\tau} &= \beta_{t_{\text{ij}},\eta}^{\tau-1}\eta + \beta_{t_{\text{ij}}}^{\tau} \end{aligned} \tag{6.6}$$

### 6.3.2 Information gathering

Each node is equipped with a pseudo watchdog component that allows it to monitor the behavior of the neighbors with whom it interacts. Specifically, if node $i$ forwards a message to node $j$, the behavior of $j$ is evaluated in terms of two types of evidence, namely: (*i*) if $j$ accepts messages of $i$ and, (*ii*) if $j$ forwards $i$'s messages of another node, say $k$. The former evidence is collected through direct communication between two nodes (i.e., through experience), meanwhile the latter, is through special feedback messages referred to as *proof of relay* [MS12] (POR) message. Therefore, $i$ waits for a POR message. However, and differently from [MS12], two types of POR messages were considered: (*i*) *type-1* that is created by $k$, which is 2 hops away from node $i$ (which can be the source or forwarder of the message) and (*ii*) *type-2* that is created by the destination of the message. Each POR contains the message identifier, the list of nodes the message traversed and the message digest. Their use depends on various factors such as the network density as, for instance, on sparse networks *type-2* can be more adequate because nodes have less contacts and less opportunities to monitor and report evidence of the behavior of other nodes. On the other hand, *type-1* shall suffice for dense networks.

The first-hand information represents the parameters of the Beta distribution assumed by node $i$ in its Bayesian opinion of node $j$'s behavior in the network. Each node keeps two data structures (records), namely, *accept first-hand information* ($\mathcal{F}_{a_{\text{ij}}}$) for accepted messages and *forward first-hand information* ($\mathcal{F}_{f_{\text{ij}}}$) for forwarded messages. Additionally, for each record there are two counters namely $\alpha$ and $\beta$. Accept and forward first-hand information are given by

$$\mathcal{F}_{x_{\text{ij}}} = (\alpha_x, \beta_x) = (\alpha, \beta)_x \tag{6.7}$$

where $x \in \{a, f\}$, and they are updated as follows:

- $\alpha$ is incremented if a good behavior is observed when:
  - node $j$ accepts messages of other nodes, e.g., node $i$. However, nodes that only accept messages may be performing black-hole attacks. Therefore, it is also necessary to ensure that node $j$ forwards messages that it receives if the message is not destined to him; or
  - node $i$ receives a POR message from $k$ because of a message $i$ forwarded to $j$. It is assumed that among all neighbors of $j$, node $k$'s delivery likelihood to the destination is the highest one.
- $\beta$ is incremented if a misbehavior is observed when:
  - node $j$ not being the destination of a message sent by node $i$, does not forward the message (no POR message was received neither did the message expire); or
  - node $j$ does not accept messages of other nodes, e.g., node $i$. Node $j$ can only refuse to accept messages forwarded to him if he already has them in buffer. Moreover, node $j$ must prove to node $i$ that he has the message in buffer as follows: node $i$ sends a message containing the message identifier (MID) and a nonce (N) to node $j$. If $j$ has the message, it must reply with a digital signature containing the digest of the message with identifier MID and N.

Since only using first-hand information may not be cost-effective, reputation systems that exclusively rely on it might have higher detection times in comparison with other approaches that also use second-hand information. A faster convergence of a reputation system is more likely as more information is considered by each node.

Second-hand information corresponds to first-hand information published by other nodes. For instance, node $i$ can gather node $k$'s first-hand information towards node $j$. Similarly to first-hand information, each node keeps two records: *accept second-hand information* ($\mathcal{S}_{a_{ij,k}}$) and*forward second-hand information* ($\mathcal{S}_{f_{ij,k}}$). Second- and first-hand information are related according to $\mathcal{S}_{x_{ij,k}} = \mathcal{F}_{x_{\mathrm{kj}}}$.

### 6.3.3 Reputation rating

A reputation rating $\mathcal{R}_{\mathrm{ij}}$ is updated (*i*) when first-hand information is updated, and (*ii*) when received second-hand information is considered valid to be incorporated.

If accept and forward first-hand information that are kept by each node are available, they are combined to form a unique first-hand information, hereafter called first-hand information $\mathcal{F}_{\mathrm{ij}} = (\alpha, \beta)_{\mathcal{F}}$, as follows

- If $\alpha_f > \alpha_a$ then $\alpha_{\mathcal{F}} = \alpha_f$. Otherwise, $\alpha_{\mathcal{F}} = \min(\alpha_a, \alpha_f)$.
- $\beta_{\mathcal{F}} = \max(\beta_a, \beta_f)$. For replication-based [SRT+10] approaches, an optimizations is proposed to penalize nodes that accept more messages than they forward: if $\alpha_a > \alpha_f$ (e.g., $\alpha_a = \chi$ and $\alpha_f = 1$) then increase $\beta_{\mathcal{F}}$ and decrease $\alpha_a$. $\chi$ represents the number of evidences of accepted messages a node has while not having any evidence of messages that the node forwarded of another node.

The first-hand information record, which contains two counters, is never published since it is considered private. What is published is the first-hand information rating that is computed using equation 6.4.

When first-hand information is updated, an exponential weighted moving average (EWMA) is used to allow for reputation fading as follows

$$\mathcal{R}_{\text{ij}}^{\tau} = (1-\phi)\mathcal{R}_{\text{ij}}^{\tau-1} + \phi\mathcal{F}_{\text{ij}}^{\tau} \tag{6.8}$$

where $\phi$ is the smoothing factor and $0 < \phi < 1$. Please note that first-hand information is equal to the accept first-hand information on the absence of forward first-hand information.

Let $\phi(1-\phi)^{n-1}$ and $(1-\phi)^{n}$ be the constants of the last but one and last terms of equation 6.8 with $n$ terms and for $\tau > 0$. The conditions $\phi(1-\phi)^{n-1} \geq (1-\phi)^{n}$ must be assured to guarantee that the last term weights less than the last but one term. For that, $\phi \geq \frac{1}{2}$.

When received second-hand information is considered valid to be incorporated, linear opinion pooling [DL16] is used for its integration. Assume two nodes $i$ and $k$ where $i$ has its opinion on how honest node $k$ is as an actor in the reputation system (i.e., the trust rating node $i$ has on $k$, $\mathcal{T}_{\text{ik}}$), and $k$ collects first-hand information about node $j$. A *recommendation* then consists in combining $i$'s opinion about $k$ with $k$'s opinion about $j$ in order for $i$ to get its opinion about $j$. It resembles trust transitivity [Jos99]. If $i$ considers $k$ trustworthy based on $\mathcal{T}_{\text{ik}}$, $\mathcal{F}_{x_{\text{kj}}}$ is used by node $i$ for updating $\mathcal{R}_{\text{ij}}$ after performing the deviation test (equation 6.10) according to

$$\begin{gathered}\mathcal{R}_{\text{ij}}^{\tau} = w_1\mathcal{R}_{\text{ij}}^{\tau-1} + w_2\mathcal{F}_{\text{kj}}^{\tau} \\ \omega_2 = \varsigma\mathcal{T}_{\text{ik}},\ 0 < \varsigma < 1\end{gathered} \tag{6.9}$$

where $w_1$ and $w_2$ are fixed non-negative weights with sum-total 1 and $\varsigma$ is the node's individuality factor. If $\varsigma$ is less than $\frac{1}{2}$, it means that a node trusts more its own experience hence guaranteeing that second-hand information carries less weight than first-hand information.

Moreover, first-hand information received from highly trusted nodes should carry more weight that the one received from nodes with low trust ratings. $\mathcal{T}_{\text{ik}}\mathcal{F}_{\text{kj}}$ allows to discount the first-hand information received as a function of the trust rating of the node that provided the information.

If $i$ considers $k$ untrustworthy, the accept and forward deviation tests are performed. The *deviation test* is computed as follows

$$\left|\mathcal{S}_{x_{ij,k}} - \mathcal{F}_{x_{\text{kj}}}\right| \geq d \tag{6.10}$$

where $d$ is the deviation threshold. The deviation test allows comparing if nodes $i$ and $k$ have similar opinions about $j$.

If the results of accept and forward deviation tests are both negative, $\mathcal{F}_{\text{kj}}$ is incorporated using equation 6.9. Otherwise, (*i*) if both are positive, $\mathcal{F}_{\text{kj}}$ is not incorporated; (*ii*) if either one or the other is positive, $\mathcal{F}_{\text{kj}}$ is incorporated at most twice since one of the deviation tests mostly probably failed because of stale recommendations.

The deviation threshold can be selected as follows. Initially, there is no information so Beta$(\alpha, \beta) = (1, 1)$. If good behaviors are continuously observed, then $\alpha$ will be continuously incremented. If $\mathbb{E}\left[\text{Beta}(\theta|\alpha, \beta)\right]$ per each increment is computed using equation 6.4 then the following monotonically increasing series is obtained

$\{1/2,\ 2/3,\ 3/4,\ 4/5,\ \ldots\}$. Making the difference of consecutive elements of the previous series, the following series is obtained $\{$1/6, 1/12, 1/20, $\ldots\}$. The deviation threshold can be set to any element or a combination of elements of the last series.

Similarly to [LW07], any node $k$'s recommendations towards $j$ are synthetized using the same moving average process as in equation 6.8, thus making the system resilient against false praise and accusation. Is it assumed that there is an acceptable number of misbehaving nodes. Second-hand information is integrated using

$$\mathcal{S}^{\tau}_{x_{ij,k}} = (1-\phi)\,\mathcal{S}^{\tau-1}_{x_{ij,k}} + \phi\mathcal{F}^{\tau}_{x_{\text{kj}}} \tag{6.11}$$

For example, if the deviation threshold is set to 1/6, only second-hand information rating whose difference to the first-hand information rating stored by the node is less of or equal to 1/6 will be incorporated. If the node is trustworthy but its recommendation differs highly from other nodes, the node's individuality factor is adjusted to reflect this difference.

### 6.3.4 Trust rating

The trust record also has the form $\mathcal{T}_{\text{ij}} = (\alpha,\beta)_{\mathcal{T}}$. As it was previously mentioned, $(\alpha,\beta)_{\mathcal{T}}$ represents the parameters of the Beta distribution assumed by node $i$ in its opinion about how honest node $j$ is as an actor in the reputation system. When node $i$ receives first-hand information from some node $k$ about node $j$, an update is performed.

Prior to incorporating the second-hand information, a deviation test is executed. On the one hand, it is used to update the trust rating node $i$ has of $k$, and on the other hand, in addition to the latter, it is also used to decide whether to update the reputation rating node $i$ has on $j$. $\alpha_{\mathcal{T}}$ is incremented if both deviation tests are positive. If both deviations tests are negative or if at least one is positive and $\mathcal{F}_{\text{kj}}$ was incorporated at most twice, then $\beta_{\mathcal{T}}$ is incremented.

The trust record $\mathcal{T}_{\text{ij}}$ is updated by

$$\begin{aligned}\alpha^{\tau}_{\mathcal{T}_{\text{ij}},\eta} &= \alpha^{\tau-1}_{\mathcal{T}_{\text{ij}},\eta}\eta + \alpha^{\tau}_{\mathcal{T}_{\text{ij}}}\\ \beta^{\tau}_{\mathcal{T}_{\text{ij}},\eta} &= \beta^{\tau-1}_{\mathcal{T}_{\text{ij}},\eta}\eta + \beta^{\tau}_{\mathcal{T}_{\text{ij}}}\end{aligned} \tag{6.12}$$

Similarly to first-hand information rating, the trust rating is computed using equation 6.4.

### 6.3.5 Bayesian classification

In classification problems, $\Theta$ is discrete and the goal is to estimate $\theta$ given an observation $x$. To address the task of finding suitable nodes to forward messages in DTNs, two binary classification problems are considered: the node's behavior ($\mathfrak{P}_1$) and trustworthiness ($\mathfrak{P}_2$) classification problems.

Let

- $\theta \in \Theta = \{\theta_0, \theta_1\}$ be the unknown state of nature.
  - For $\mathfrak{P}_1$: $\theta = \{\theta_0 = \mathrm{good/normal}, \theta_1 = \mathrm{bad/misbehaving}\}$.
  - For $\mathfrak{P}_2$: $\theta = \{\theta_0 = \text{trustworthy}, \theta_1 = \text{untrusworthy}\}$

- $X \in \mathcal{X}$ be a random variable with $\{f(x|\theta), \ x \in X\}$
- $\pi(\theta) > 0$ and $\sum_{\theta\in\Theta} \pi(\theta) = 1$ be the prior probability mass function
- $a \in \mathcal{A} = \{a_0, a_1\}$ be the allowed decision or action.
  - For $\mathfrak{P}_1$: $a = \{a_0 = \text{FORWARD}, a_1 = \text{DO\_NOT\_FORWARD}\}$.
  - For $\mathfrak{P}_2$: $a = \{a_0 = \text{TRUST}, \ \mathrm{a}_1 = \text{DO\_NOT\_TRUST}\}$.
- The "0/1" loss function is used for classification. It assigns zero cost to any correct decision and unit cost to any wrong decision, that is,

$$L(\theta, a) = \begin{cases} 1 \leftarrow \text{if, e.g., } \theta = \theta_0 \text{ decide } a = a_1 \\ 0 \leftarrow \text{if, e.g., } \theta = \theta_1 \text{ decide } a = a_1 \end{cases} \tag{6.13}$$

The optimal Bayesian decision is given by

$$\begin{aligned} \delta_{\text{Bayes}}(x) &= \underset{a\in\mathcal{A}}{\operatorname{argmin}}\, \rho(\pi(\theta), a|x) \\ &= \arg \ \ (L(\theta_0, a) f(x|\theta_0) \pi(\theta_0) + L(\theta_1, a) f(x|\theta_1) \pi(\theta_1)) \\ \delta_{\text{Bayes}}(x) &= \begin{cases} \theta_0 \leftarrow l(x) \geq t \\ \theta_1 \leftarrow l(x) < t \end{cases} \end{aligned} \tag{6.14}$$

where $l(x) = \frac{f(x|\theta_0)}{f(x|\theta_1)}$ is the likelihood ratio and $t = \frac{\pi(\theta_0)}{\pi(\theta_1)}$ is the decision threshold.

The likelihood function is given by the Bernoulli distribution $f(x|\theta) = \theta^r (1-\theta)^{n-r}$, where $r = \sum_{i=0}^{n} x_i$, and $r$ denotes the number of outcomes representing correct behavior.

In the beginning, if the only information available is the conditional probability density function of the observation given the true $\theta$, the maximum likelihood decision criterion ($\delta_{\text{ML}}$) [MC78] is used. $\delta_{\text{ML}}$ is defined as

$$\delta_{\text{ML}} = \begin{cases} \theta_0 \leftarrow l(x) \geq 1 \\ \theta_1 \leftarrow l(x) < 1 \end{cases} \tag{6.15}$$

### Node's behavior classification problem

After each interaction between two nodes, the sender updates the reputation rating of the other node based on the result of this interaction. Each node clusters the other nodes to whom it interacted in two groups: normal nodes, if $\mathcal{R}_{\text{ij}} \geq 1/2$, and misbehaving nodes, if $\mathcal{R}_{\text{ij}} < 1/2$. The prior probabilities $\pi(\cdot)$ of these clusters, which allow determining the decision threshold, are coefficients of the convex combination of the number of nodes in these clusters. The optimal Bayesian decision is computed using equation 6.14 given the prior probabilities. However, if a correct behavior is observed and $\pi(\theta_1) > \pi(\theta_0)$, one may incur in false positives, i.e., a misclassification, while using the optimal Bayesian decision criterion, because of the higher weight of the decision threshold in comparison to the likelihood ratio. The workaround consists in finding attenuation parameters $\alpha$ and $\beta$ of the *posterior mean Bayesian estimator* $\left(\hat{\theta}_{\text{PM}}\right)$ [Fig04] and computing an attenuated decision threshold. $\hat{\theta}_{\text{PM}}$ is given by

$$\hat{\theta}_{\text{PM}} = \frac{\alpha + r}{\alpha + \beta + n} \tag{6.16}$$

For the minimum possible case, i.e., one correct behavior being observed and two clusters, one with 2 misbehaving nodes and the other with 1 normal node, $l(x)$ is $4/3$. The Bayesian attenuation parameters $\alpha$ and $\beta$ can be determined from

$$t = \frac{1 - \hat{\theta}_{\text{PM}}}{\hat{\theta}_{\text{PM}}} \leq \frac{4}{3} \Longrightarrow \hat{\theta}_{\text{PM}} \geq 3/7 \tag{6.17}$$

If $\alpha = \beta$,

$$\frac{\alpha + r}{2\alpha + n} \geq \frac{3}{7} \tag{6.18}$$

$$\alpha \geq 3n - 7r \tag{6.19}$$

For the case above, $\alpha = 2$ and the decision threshold is equal to the likelihood ratio. If instead the *maximum a posteriori Bayesian estimator* [Fig04] was used, the decision threshold would be greater than the likelihood ratio which would lead to misclassification.

**Trustworthiness classification problem**

Each node also clusters nodes that sent first-hand information to him in two groups: trustworthy, if $\mathcal{T}_{\text{ij}} > 1/2$, and untrustworthy, if $\mathcal{T}_{\text{ij}} < 1/2$, based on the result of the deviation test. The deviation test is performed after the bootstrapping of the Trust Engine. During bootstrapping, nodes' recommendations are synthesized (equation 6.11). Ideally, the bootstrapping period should not be inferior to the time necessary for the distributed reputation system to converge, i.e., for each node's reputation engine to be able to classify correctly all the nodes with which it interacted.

In the same way to the node's behavior classification problem, the optimal Bayesian decision is computed using an attenuated decision threshold (equations 6.16 and 6.19) to avoid misclassifications.

## 6.4 Simulation model

The robust and distributed reputation system was implemented on the Opportunistic Network Environment (ONE) simulator [KOK09]. The simulation model consisted of a network with 150 pedestrians. The simulation time was 7 days with an update interval of 1.0 s. As explained in Section 6.3.3, the selected deviation threshold value is $1/6$. The individuality factor needs, according to its definition to be greater than $1/2$. By setting it to $2/3$, it means that first-hand information weights $2/3$, that is, the double of the second-hand information weight, i.e., 1/3. The same goes to the smoothing factor. The nodes misbehavior considered for evaluation were liars and black-hole attacks. It was considered that misbehaving nodes were also colluding, i.e., they increased $\alpha$ of misbehaving nodes and $\beta$ of normal nodes. Despite not being implemented, sleeper attacks were addressed by the proposed reputation system by means of the aging scheme as nodes' reputation and trust ratings were decayed as time passed allowing for redemption. The effects of nodes' misbehavior was examined on 3 routing protocols, namely Epidemic, MaxProp and Prophet [MPC13a]. The percentage of liars and nodes that performed black-hole attacks varied from 20% to

80% with increments of 20%.

*Mobility.* A map-based mobility model of the Helsinki City over an area of 4.5 × 3.4 Km was used. The pedestrians were moving in a speed varying between 0.8 to 1.4 m/s. Each time a pedestrian reaches its destination, it pauses for 100 seconds.

*Connectivity and transmission.* Only two nodes within range can communicate with each other at a time. The communication range between nodes is 10 m, and the communication is bidirectional at a constant transmission rate of 2 Mbit/s.

*Traffic model.* Every one to two minutes, a source node randomly chosen can generate one message to a randomly chosen destination. Nodes do not change their behavior (malicious or not) over time. The TTL attribute of each message is 12 h, and the message size varies from 500 kB to 1 MB.

*Buffer management.* The pedestrians' devices have a buffer size of 20 MB for DTN traffic.

## 6.5 Simulation results

The evaluation of the performance of the reputation system consisted in evaluating the reputation and trust modules, similarly to previous work [BL04]. Additionally, Bayesian classification at the routing decision module is also evaluated. For each setting, i.e., protocol-percentage pair, thirty independent simulations using different seeds were conducted, and the results averaged, for statistical confidence.

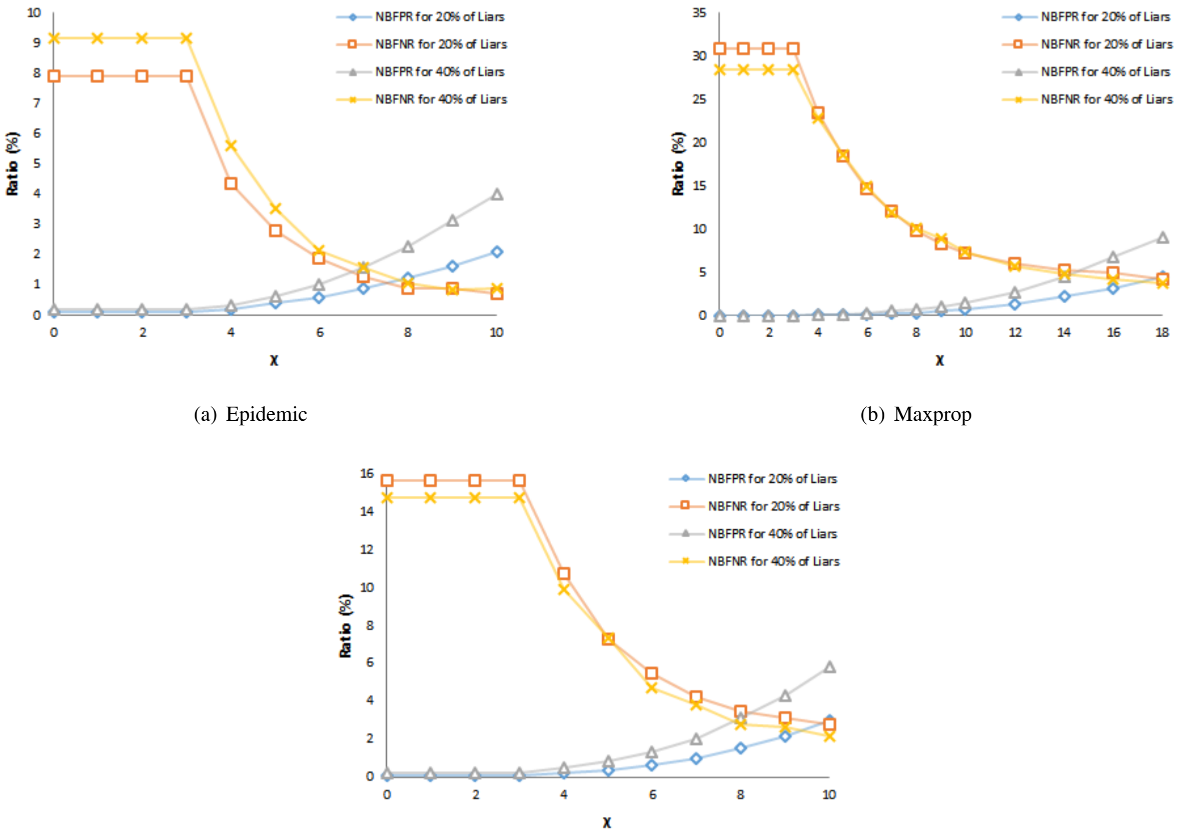
The following metrics were considered for the evaluation of the proposed reputation system:

- *Detection time of misbehaving nodes*, which is measured as the simulation time taken for all normal nodes to correctly classify all misbehaving nodes they came in contact with, starting at the detection instant of the first misclassification.

- *Robustness* against false accusations (false negatives) and false praise (false positives). The following metrics were defined:

  - Node's Behavior False Positives Ratio (NBFPR) is the number of misbehaving nodes with good node's behavior classification, i.e., classified as FORWARD, over all nodes classified.
  - Node's Behavior False Negatives Ratio (NBFNR) is the number of good nodes with bad node's behavior classification, i.e., classified as DO_NOT_FORWARD, over all nodes classified.
  - Node's Trustworthiness False Positives Ratio (NTFPR) is the number of misbehaving nodes with good node's trustworthiness classification, i.e., classified as TRUST, over all nodes classified.
  - Node's Trustworthiness False Negatives Ratio (NTFNR) is the number of good nodes with bad node's trustworthiness classification, i.e., classified as DO_NOT_TRUST over all nodes classified.

- *Reputation system overhead ratio (RSOR)*, which is the ratio between the number of control messages of the reputation system (i.e., first-hand information messages published and POR messages disseminated) over all messages.



(a) Epidemic

(b) Maxprop

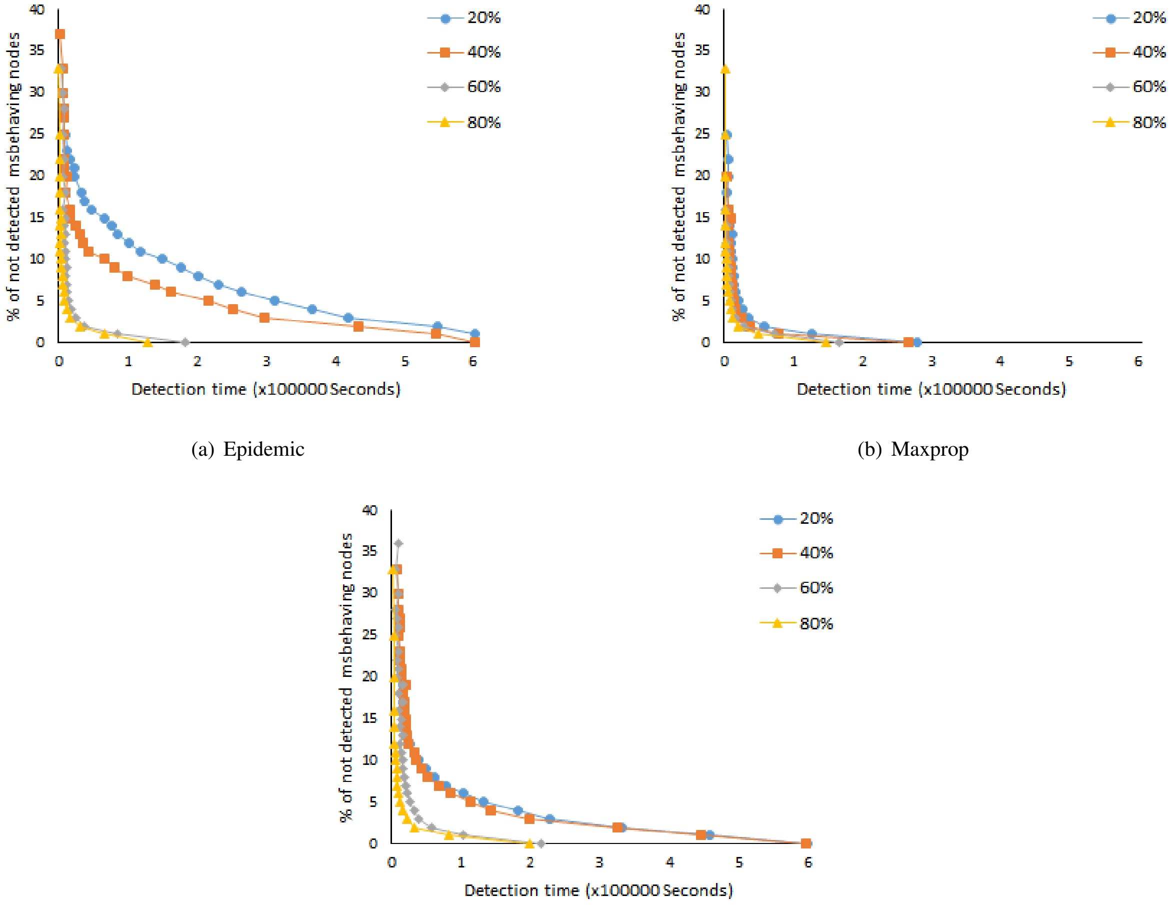
(c) Prophet

Figure 6.1: The variation of NBFPR and NBFNR as function of $\chi$ for the Epidemic, MaxProp and Prophet routing protocols

### 6.5.1 The selection of the parameter $\chi$

Figure 6.1 shows the variation of NBFPR and NBFNR as function of $\chi$ for the Epidemic, MaxProp and Prophet routing protocols. Recall that $\chi$ is the number of evidences of accepted messages node $A$ has of node $B$ while not having any evidence of forwarded messages of $B$. It allows identifying nodes performing black-hole attacks even while colluding.

For example, consider Figure 6.1(a). For small values of $\chi$ (0-4), the reputation system isolates misbehaving nodes and penalizes good nodes that do not forward messages (most probably only accepted messages of other nodes) or even whose PORs arrived after the evidence that the message was forwarded has expired. In other words, for small values of $\chi$, the system is too rigid. On the other hand, for high values (8-10) of $\chi$, the system is too tolerant as the number of false positives becomes greater than the number of false negatives meaning that misbehaving nodes are not being adequately penalized. A good compromise is achieved with middle values (5-7) since the number of false positives is smaller than the number of false negatives, i.e., the system adequately penalizes misbehaving nodes and good nodes that do not participate in the network by forwarding other nodes' messages. In addition, the middle values have an average of 0.86% and 2.20% for NBFPR and NBFNR, respectively, which are acceptable values.



(a) Epidemic

(b) Maxprop

(c) Prophet

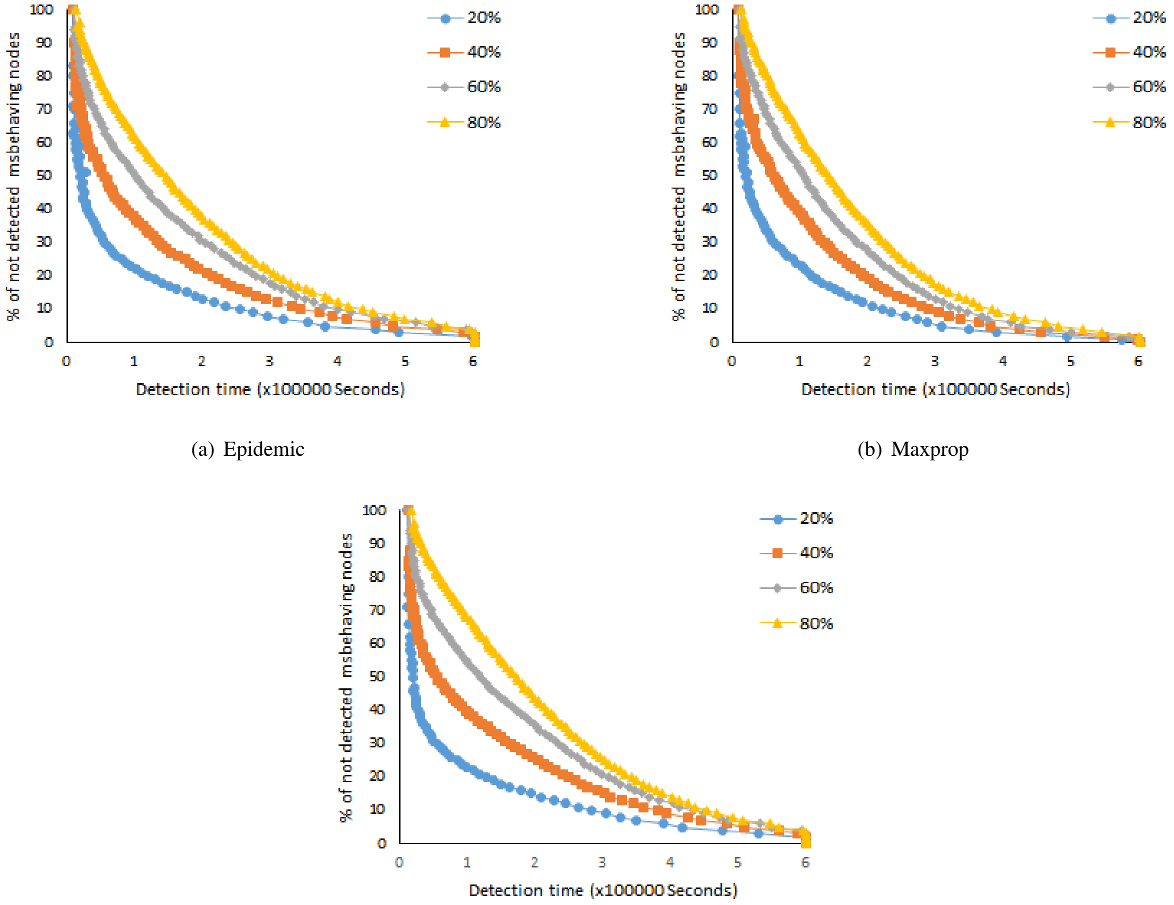
Figure 6.2: The time necessary to correctly classify misbehaving nodes as DO_NOT_FORWARD for Epidemic, MaxProp and Prophet routing protocols and for 20, 40, 60 and 80% of liars

### 6.5.2 Detection time of misbehaving nodes

Figures 6.2 and 6.3 present the time necessary for each good node to classify correctly all misbehaving nodes it met as DO_NOT_FORWARD for Epidemic, MaxProp and Prophet and for four percentages of misbehaving nodes, i.e., 20, 40, 60 and 80%.

The maximum likelihood and optimal Bayesian decision criteria were considered for the lying attack. No plots were presented for the lying attack with the maximum likelihood decision criterion because the proposed reputation system was able on the fly to detect all misbehaving nodes, i.e., the detection time in all protocols considered was zero. The optimal Bayesian decision criterion needs prior information to calculate the threshold, but prior information is not available at the beginning of the simulations. Consequently, the maximum likelihood decision criterion is used. As time passes, nodes learn by interacting with others and are able to create clusters using local information (i.e., based on their opinion about reputation ratings of other nodes) of normal and misbehaving nodes, thus obtaining prior information. Please note that the prior information is also updated at each interaction, what enables the classifier to learn as new observations are made.

MaxProp presented on average 58.2% and 70% lower detection time than Epidemic and Prophet, respectively, for the lying attack. The detection time was directly influenced by the routing layer, i.e., the algorithm used to



(a) Epidemic

(b) Maxprop

(c) Prophet

Figure 6.3: The time necessary to correctly classify misbehaving nodes as DO_NOT_FORWARD for Epidemic, MaxProp and Prophet routing protocols and for 20, 40, 60 and 80% of black-hole nodes

disseminate messages across the network. Ideally, the goal of any reputation system is to correctly classify all the other nodes with whom a given node interacts (e.g., for the simplest case, if the node accepted and forwarded the message it received) with the least possible number of contacts. However, overhead causes nodes to interact many times with the same node or group of nodes.

MaxProp's replication strategy played its role because controlled message replication and the protocol's reduced overhead contributed to the reduction of the detection time. Epidemic performance was most of the times affected by the protocol's excessive overhead therefore increasing the detection time of liars. Nonetheless, the presence of liars improved the Epidemic's performance since less message copies were created, as liars do not accept them and by not accepting they were detected thus reducing the detection time.

Moreover, by increasing the percentage of liars, all routing protocols reduced the detection time since good nodes got more often in contact with an increasing number of misbehaving nodes, which allowed them to faster and correctly classify misbehaving nodes. In summary, different routing protocols present different detection times because of the different number of message replicas each routing protocol generates at each contact. Excessive overhead has a negative impact as it increases detection time.

For simplicity, only the evaluation of the maximum likelihood decision criterion for the black-hole attack is presented. The black-hole attack requires both accept and forward first-hand information conversely to the lying attack that only required accept first-hand information. The reputation system used all the available information to infer the behavior of each node it interacted with. Still, in some cases, even though there were many evidence that a given node accepted many messages and did not forward any, one could not say for sure that this node was misbehaving since some good nodes also presented a similar behavior.

For the black-hole attack, the reputation system must penalize nodes that only accepted but did not forward messages given that evidence these messages were not forwarded expired. Even if a small penalization was given, misbehaving nodes performing black-hole attacks would be detected. However, good nodes that behaved similarly to misbehaving nodes would be also isolated from the network, although temporarily, because of the aging mechanism or if they started forwarding messages.

For POR *type-2*, since an evidence has, by default, the same TTL of a message that originated it, there is a tradeoff between the TTL and the detection time. If the goal is for the reputation system to converge sooner (i.e., to have a small detection time) then the TTL should not be too high. Otherwise, PORs might not have enough time to be effectively disseminated over the network, which will increase the number of misclassifications as a consequence of a too small TTL. Nevertheless, the reputation system took more time to detect an increasing percentage of nodes performing black-hole attacks in contrast to liars where an increased number of liars took less time to detect, mainly because of forward first-hand information.

It was noticed that the use of second-hand information did not have influence on the detection times. Its use did influence the ratios of false positives and negatives as will be explained on the next section.

### 6.5.3 Robustness

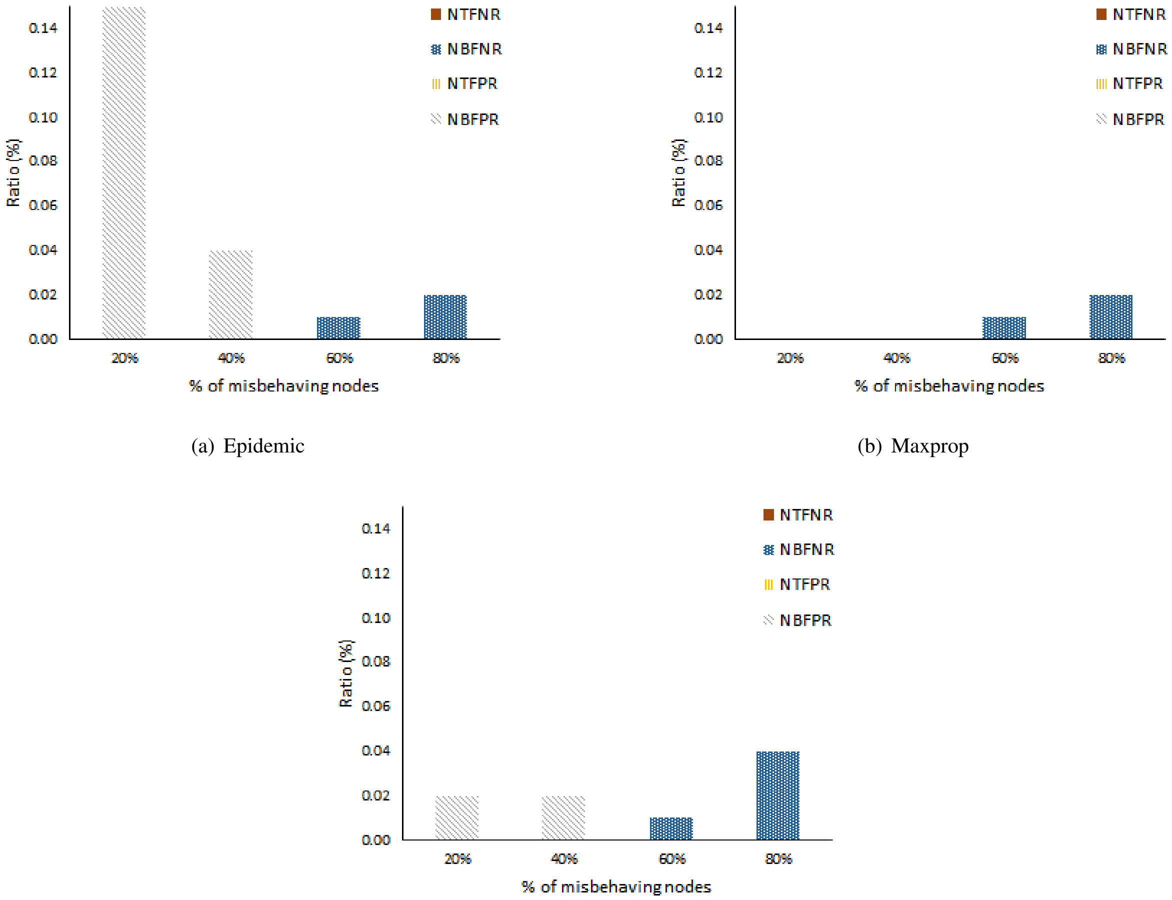
In Figures 6.4 and 6.5, four metrics were considered to measure the robustness of the reputation system against false accusations and praise for the lying and black-hole attacks. These metrics were obtained at the end of the simulations. One can conclude, by analyzing both Figures, that there are no misclassified good nodes.

Similarly to the previous section, no plots of the maximum likelihood decision criterion were presented.

The use of second-hand information by the optimal Bayesian decision criterion did not have any influence on the metrics considered. There are two reasons for that: (*i*) the bootstrapping of the trust engine and (*ii*) the tolerance to nodes that failed the deviation test (equation 6.10). Recall that the former allows each node to synthesize first-hand information received from other nodes (i.e., collect and accumulate using EWMA, see equation 6.11). By comparing the received information with the accumulated one on each node, the probabilities of false praise and accusations were small. But then again, as on DTNs many nodes get isolated, it was noticed that some nodes failed the deviation test because of stale accumulated information. Consequently, each node tolerates failures to the deviation test up to a given number of times. The combination of these two techniques allowed the trust engine to presents zero false positives and negatives in most of the cases.

Additionally, there is also a tradeoff between false positives and negatives. By attempting to isolate misbehaving nodes (i.e., to reduce the false positives ratio), good nodes that up to a given instant only accepted messages will be misclassified as DO_NOT_FORWARD, therefore increasing the ratio of false negatives.

Another aspect that affects robustness for the black-hole attack was the selection of the parameter $\chi$, as explained on Section 6.5.1.



(a) Epidemic

(b) Maxprop

(c) Prophet

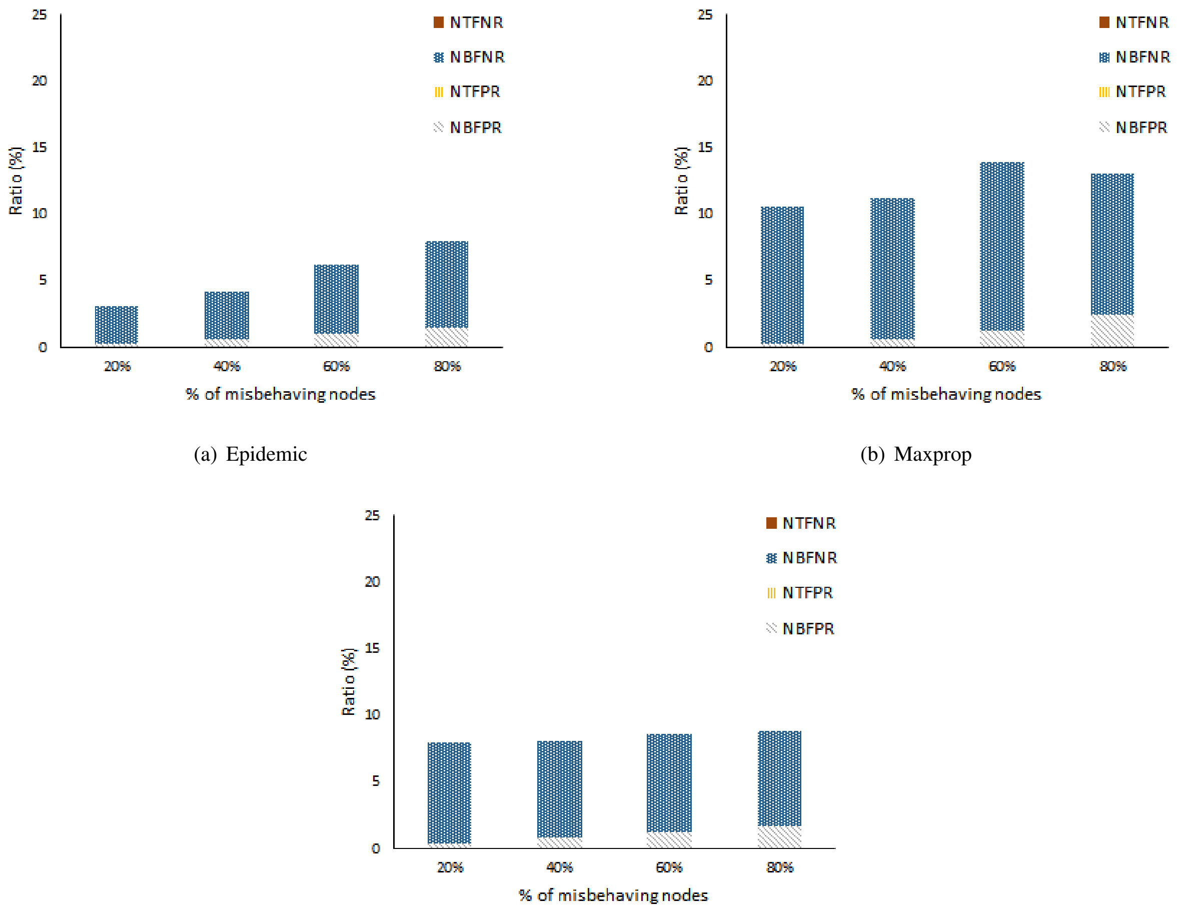
Figure 6.4: Reputation, Trust, Node's behavior and Trustworthiness false positives and negatives ratios for Epidemic, MaxProp and Prophet routing protocols and for 20, 40, 60 and 80% of liars

### 6.5.4 Reputation system overhead

Figure 6.6 presents the reputation system overhead ratio (RSOR) of three DTN routing protocols (Epidemic, MaxProp and Prophet) for 20, 40, 60 and 80% of nodes performing the lying and black-hole attacks. The RSOR, which was obtained at the end of the simulation, consists of first-hand information messages published and POR messages disseminated (for the black-hole case).

Similarly to the previous sections, no plot of the maximum likelihood decision criterion were presented. Therefore, RSOR was zero for all routing protocols.

As the percentage of misbehaving nodes increases, the number of data and reputation system's control messages reduces. However, the proportion at which the former and latter reduce differs. For instance, RSOR increases if the reduction of the former is greater than the reduction of the latter; otherwise, RSOR reduces. The reduction of the number of good nodes reduces the number of disseminated *type-1* POR and published first-hand information messages, which are the main reasons for the reduction of the reputation system's control messages.



(a) Epidemic

(b) Maxprop

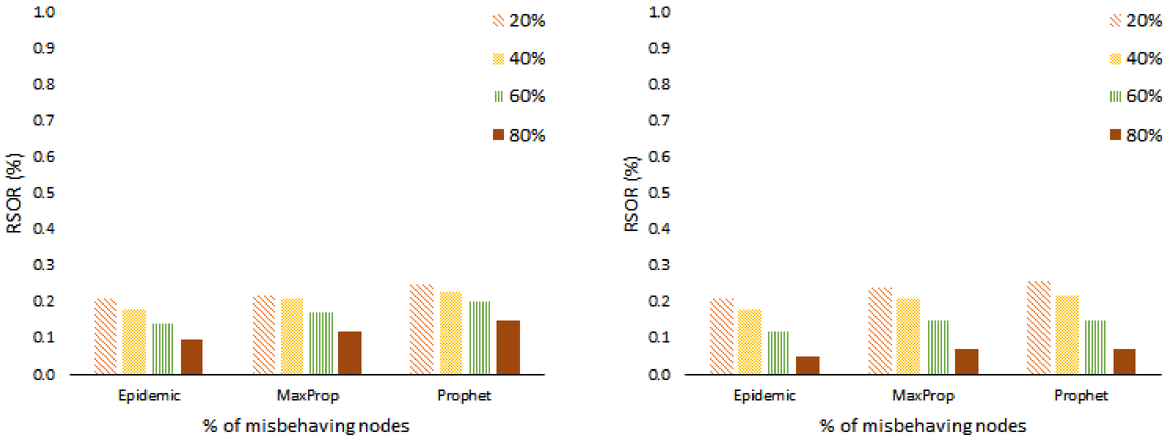
(c) Prophet

Figure 6.5: Reputation, Trust, Node's behavior and Trustworthiness false positives and negatives ratios for Epidemic, MaxProp and Prophet routing protocols and for 20, 40, 60 and 80% of black-hole nodes

## 6.6 Summary

In this chapter, a robust and distributed reputation system for DTNs is proposed. The reputation system takes into account all the available information and uses Bayesian decision theory to classify nodes. The system is robust because despite taking into account all the available information, it is resilient against false accusations and praise, and distributed, as the decision to interact with another node depends entirely on each node.

Simulation results show that the system is able, for the node's classification problem, to classify correctly on the fly liars that collude using the maximum likelihood decision criterion, and, for the node's trustworthiness problem, it is also able to classify correctly nodes in most cases. Moreover, there are tradeoffs in this system. For instance, if the evidence's TTL is too high, the reputation system will take more time to converge as the detection time increases. On the other hand, if the TTL is too small, PORs might not have enough time to be effectively disseminated over the network that will increase the number of misclassifications. On the other hand, by attempting to isolate misbehaving nodes (i.e., to reduce false positives ratio), good nodes that up to a given instant only accepted messages will be also misclassified, that is, classified as DO_NOT_FORWARD, therefore increasing the ratio of false negatives.



(a) Liar

(b) Black-hole

Figure 6.6: The reputation system overhead ratio (RSOR) for Epidemic, MaxProp and Prophet routing protocols and for 20, 40, 60 and 80% of misbehaving nodes

# Chapter 7

# PRIVO: A PRIvacy-preserVing Opportunistic routing protocol for Delay-Tolerant Networks

In this chapter, the PRIvacy-preserVing Opportunistic routing protocol for Delay-Tolerant Networks (PRIVO) [MBPC17] is presented. PRIVO models a DTN as a time-varying neighboring graph where edges correspond to the neighboring relationship among pairs of nodes. PRIVO ensures privacy by protecting each node's sensitive information even if it has to be processed elsewhere. In addition, nodes also compare their routing metrics in a private manner using the Paillier homomorphic encryption scheme. The effectiveness of PRIVO is supported through extensive simulations with synthetic mobility models and real mobility traces.

## 7.1 The PRIVO protocol

The PRIvacy-preserVing Opportunistic routing protocol for Delay-Tolerant Networks (PRIVO) detects and utilizes the inherent social network structure to facilitate packet forwarding in DTNs. It models a DTN as a time-varying neighboring graph where vertices correspond to nodes and edges correspond to the neighboring relationship among pairs of nodes.

PRIVO ensures privacy by means of anonymization and homomorphic encryption. It uses anonymization to avoid disclosing historical information associated to each node's neighboring graph. Moreover, when two nodes meet, they do not share private information associated with their routing metrics, which is necessary to identify the best message forwarder. Nodes compare these metrics in a private manner using homomorphic encryption.

The PRIVO protocol is composed of the following steps: construction and anonymization of the neighboring graph, determination of routing metrics and the routing algorithm.

In order to facilitate future references, frequently used notations in this chapter are listed below in Table 7.1.

| Notation | Meaning |
|---|---|
| $x_{i,j}(T)$ | The separation period between nodes $i$ and $j$ |
| $\tau$ | The elapsed time or time period |
| $\delta_{i,j}(x)$ | The average separation period |
| $w_{i,j}$ | The PRIVO weight |
| $\rho$ | The anonymization threshold |
| $\varepsilon$ | The weight threshold |
| $\eta$ | The number of timeslots |
| $\chi$ | A random number |

Table 7.1: The most used notations in this chapter

### 7.1.1 Construction of the neighboring graph

Let $x_{i,j}(T)$ denote the separation period between nodes $i$ and $j$, $\tau$ denote the elapsed time and $n_{i,j}$ be the number of times that nodes $i$ and $j$ were away from each other. So, $x_{i,j}\left(T\right) = 0$ means that nodes $i$ and $j$ are within communication range at time $T$, otherwise $x_{i,j}\left(T\right) = 1$. The time-varying average separation period (hereafter average separation period) is given by

$$\delta_{i,j}(x) = \frac{\int_{\tau} x_{i,j}\left(T\right) dT}{n_{i,j}} \tag{7.1}$$

The normalized average separation period $\hat{\delta}_{i,j}$ is given by

$$\hat{\delta}_{i,j} = 1 - \frac{\delta_{i,j}}{\tau} \tag{7.2}$$

and the unbiased variance estimator is given by

$$\hat{\sigma}(x) = \frac{1}{l}\sum_{k=1}^{l}\left(x_k - \delta(x)\right)^2 \tag{7.3}$$

The average separation period aims at capturing the evolution of social interactions in similar time-periods (or timeslots). In here, daily timeslots were considered. The average separation period in the same timeslot over consecutive days is updated using an exponential weighted moving average as follows

$$\hat{\delta}^{T} = (1-\alpha)\cdot\hat{\delta}^{T-1} + \alpha\cdot\hat{\delta}^{T} \tag{7.4}$$

where $\alpha$ is the smoothing factor, and $0 < \alpha < 1$, and it is depreciated over consecutive timeslots as follows

$$\hat{\delta}^{\tau} = (1-\alpha)\cdot\hat{\delta}^{\tau-1} \tag{7.5}$$

The unbiased variance estimator is updated as follows

$$\hat{\sigma}^{T} = (1-\beta)\cdot\hat{\sigma}^{T-1} + \beta\cdot\left(x-\hat{\delta}\right)^{T} \tag{7.6}$$

The social strength among nodes in a specific daily timeslot may provide insights on their social strength in consecutive timeslots on the same day, therefore increasing the probability of nodes being capable of transmitting data as transmissions could be resumed, with high probability, on the same timeslot on the next day. The time-varying PRIVO weight $w_{i,j}$ (hereafter *pweight*) over a daily timeslot is given by

$$w_{i,j} = \frac{1}{|\tau|} \sum_{k=1}^{|\tau|} \hat{\delta}_k \tag{7.7}$$

where $|\tau|$ is the number of sub-intervals (or timeslots); *pweight* shows the neighboring relationship among nodes and gives hints about the forwarding opportunities between them, i.e., larger $w_{i,j}$ indicates a better future contact probability between nodes $i$ and $j$.

In PRIVO, nodes' routines are used to quantify the time-varying strength of social ties between nodes. For instance, if daily routines are considered, each node computes the average separation periods to other nodes during the same set of daily timeslots over consecutive days.

### 7.1.2 Anonymization of the neighboring graph

A DTN node may disclose private information by sending private data to other nodes. Privacy-preservation techniques allow protecting privacy through masking, modification and/or generalization of the original data without sacrificing data utility.

PRIVO deals with link disclosure since each node's ego network contains the list of neighbors and their social strengths. PRIVO proposes two anonymization techniques that are suitable for DTNs as they ensure data utility: neighborhood randomization and binary anonymization.

*Neighborhood randomization* consists in partially hiding each node's neighboring graph containing its historical encounter information. When two nodes are in communication range, they only exchange the least possible number of nodes in their neighboring graphs. If $w_{i,j}$ is high, it might mean that nodes $i$ and $j$ have a strong tie (i.e., that they meet often), or even that they have met recently. The latter may be a random link, that is, a recent occasional connection that looks like a strong tie.

Neighborhood randomization works as follows: upon an encounter between nodes $i$ and $j$, each node selects a random number between $\chi \in \mathbb{N}$ and the total number of nodes in its neighboring graph. $\chi$ should be selected taking into account the amount of information that each node is willing to share. Note that there is a tradeoff between the amount of information to share and the performance of the routing protocol. Sharing less information might compromise the utility of the randomized neighboring graph. If too much information is shared, the node might be disclosing too much private information. The set of nodes to share between $i$ and $j$ is then randomly selected among all possible ones and it is limited by the smallest previous randomly selected number. This allows hiding each node's degree. Randomly selecting nodes to add to the anonymized neighboring graph that will be shared allows mixing random contacts with strong contacts, therefore hiding the contact patterns among neighbors since *pweights* are constantly being updated. If $i$ and $j$ re-encounter after a short period of time, they can share the same previous information therefore avoiding to disclose more historical information. Ideally, upon an encounter between nodes $i$ and $j$, the anonymized neighboring graph of $j$ should only contain information of common nodes it has with $i$. This information is useful for $i$ to update its ego network.

*Binary anonymization* consists in replacing the *pweight* associated to a given link with 1 or 0, if the weight is above or below a given anonymization threshold ($\rho$), respectively. This technique converts the weighted (randomized or not) neighborhood graph into an unweighted one, therefore hiding the *pweight* associated to a given edge.

The selection of $\rho$ is also limited by the utility of the neighboring graph. Consider, for example, that node $a$ has nodes $b$, $c$ and $d$ as its neighbors with *pweights* $(w_{a,b} = 0.05, w_{a,c} = 0.15, w_{a,d} = 0.65)$. If $\rho$ is set to 0.1, the anonymized *pweights* are $(w^*_{a,b} = 0, w^*_{a,c} = 1, w^*_{a,d} = 1)$. But, if instead $\rho$ is set to 0.25, the anonymized *pweights* would become $(w^*_{a,b} = 0, w^*_{a,c} = 0, w^*_{a,d} = 1)$. If node $a$ meets another node, say node $e$, $a$ would tell $e$ that its neighbors are $(w_{a,c} = 1, w_{a,d} = 1)$ for $\rho = 0.1$ and $(w_{a,d} = 1)$ for $\rho = 0.25$.

### 7.1.3 Determination of routing metrics

Previous work [WLX14] succeeded in identifying social structures, but the routing performance is affected as they did not take into consideration the dynamics of the network, i.e., the making and breaking of social ties. If, for instance, social similarity is considered, it is important to map actual interactions among nodes into social connectivity graphs comprising only stable social contacts in order to improve forwarding performance.

PRIVO represents the dynamics of the social structure as time-varying weighted neighboring graphs, where the weights (i.e., social strengths among nodes) express the average separation period over different timeslots.

#### Ego betweenness centrality

In PRIVO, each node's ego network corresponds to its neighboring graph if the *pweigths* are above a given weight threshold ($\varepsilon$). Since the connections among the ego direct neighbors are also necessary for the ego network, each node shares its anonymized neighboring graph (as explained in Section 7.1.2) with its neighbors.

Given a set of configuration parameters (see Section 7.2 for more details), the determination of $\varepsilon$ can be seen as an optimization problem consisting in finding the $\varepsilon$ that maximizes (or minimizes) a certain routing performance metric (e.g., finding $\varepsilon$ that maximizes the delivery ratio).

#### Weighted similarity to the destination

Let $\mathcal{A}_n$ be the weighted adjacency matrix of node $n$ at a given timeslot. Let $\mathcal{A}_{n_{i,j}} = w_{i,j}$. If nodes $i$ and $j$ have met before, then $w_{i,j} \neq 0$; otherwise, $w_{i,j}$=0. The weighted similarity of $n$ to a destination node $d$ $(s_d)$ is obtained by summing the non-zero row entries in $\mathcal{A}_{n\,i,d}|i \neq n$. If $n$ never met $d$ but node $i$ belonging to $n$'s neighboring graph did, $n$ may infer that $i$ is a more suitable forwarder to $d$ than him through $i$'s anonymized neighboring graph.

#### Mean time to encounter

Besides *pweight*, PRIVO also uses a metric called *mean time to encounter* (MTTE) to determine the best message forwarder to a given destination taking into account the average separation period at each timeslot and the expected time necessary for the two nodes to re-encounter. Specifically, given that in PRIVO each node keeps an estimate of the average separation period at each timeslot that is updated as nodes encounter each other, PRIVO predicts the most probable timeslot for future contacts also taking into account the shortest time to re-encounter. As an example, consider that node $a$ meets nodes $b$ and $c$ at 2pm and 5pm for 10 and 15 minutes, respectively. At 8pm, node $a$ receives a message destined to node $d$ that is expected to meet nodes $b$ and $c$ on the next day. When node $a$ computes the average separation periods of $b$ and $c$, it also considers the time to re-encounter nodes $b$ and $c$ in the following day assuming that these nodes maintain similar habits.

### 7.1.4 Routing algorithm

This section describes PRIVO's routing algorithm, i.e., the messages exchanged using the Paillier homomorphic encryption scheme and the routing decision process.

#### The attribute privacy mechanism

PRIVO ensures attribute privacy, as regardless of the metric ($\mathfrak{m}$) used by the routing algorithm (*pweight*, similarity to the destination, or ego betweenness centrality – $\{w_{i,j},\ s_d,\ c_{\text{EBC}}\} \subset \mathfrak{m}$), which represent utility, when two nodes meet they find the best forwarder in a private manner using the Paillier homomorphic encryption scheme.

Let $A$ be a node carrying a set of messages $\mathcal{M}$ and node $B$ be a neighbor of $A$. Let $A \rightarrow B\ :< message >$ denote a message sent from $A$ to $B$. Upon an encounter, $A$ wants to know if $B$ is the best forwarder to carry $m \in \mathcal{M}$ destined to node $D$. Let $p_k$ and $s_k$ be public and private key, respectively.

The exchange of messages in PRIVO works as follows:

1. Node $A$ calculates metric $\mathfrak{m}$ for each $m \in \mathcal{M}$ using the information it has available.

2. Each time $A$ establishes a contact with another node, it announces: $-\mathfrak{m}_i\ \forall\ D_i \subset m_i|\ i = 1, 2,\ \ldots |\mathcal{M}|$, the destination of the message and its public key ($p_{k_A}$) to $B$. Node $A$ multiplies the metric $\mathfrak{m}_i$ by -1 to reduce the number of cryptographic operations to be performed by node $B$.

$$A \rightarrow B\ :< \mathcal{E}_{p_{k_A}}\left(-\mathfrak{m}_{A_i}\right), D_{m_i},\ p_{k_A} >$$

3. Node $B$ performs for each metric received the following operations: first, $B$ sums $-\mathfrak{m}_{A_i}$ to the corresponding metric $\mathfrak{m}_{B_i}$, then it multiplies the result by a random one-use number (nonce) to randomize it. Without the multiplication, $A$ would be able to obtain $\mathfrak{m}_{B_i}$. Then $B$ sends the result $\mathcal{R}_i = nonce \cdot (-\mathfrak{m}_{A_i} + \mathfrak{m}_{B_i})$ to $A$.

$$B \rightarrow A\ :\ < \mathcal{E}_{p_{k_A}}\left(\mathcal{R}_i\right) >$$

4. $A$ decrypts the received comparisons for each $m_i$.

$$\mathcal{D}_{s_{k_A}}\left(\mathcal{E}_{p_{k_A}}\left(\mathcal{R}_i\right)\right)$$

   Node $A$ knows that if $\mathcal{R}_i > 0 \rightarrow \mathfrak{m}_A < \mathfrak{m}_B$ which means that node $B$ is the best forwarder. If that is the case, $A$ forwards $m_i$ to $B$.

$$A \rightarrow B\ :< m_i >$$

Obtaining the best forwarder can be demanding in terms of CPU, energy, etc., due to the number of messages that have to be exchanged in the process. In PRIVO, each node has a secure forwarding table (SFT) containing entries $< DestinationNode\ (DN),\ BestForwarder\ (BF) >$ that is updated each time a node meets another one that is a better forwarder than him. When the average separation period between two nodes is updated, if one of those nodes is a BF in the SFT, the entry is removed. SFT allows to reduce the number of messages exchanged when two nodes meet therefore reducing also PRIVO's consumption of resources.

**The routing decision process**

Because of the different routing metrics considered here, four variants of PRIVO are proposed. PrivoASP uses as routing metric *pweight*. PrivoMTTE uses as routing metric the mean time to encounter. PrivoSDBC, which is the social version of PRIVO, uses as routing metric weighted similarity to the destination and ego betweenness centrality. PrivoCOMBINED is a combination of PrivoMTTE and PrivoSDBC. It results from multiplying the routing metrics of PrivoMTTE and PrivoSDBC.

However, independently of the routing metric used, all messages to send whose next forwarders are in the SFT are sorted based on the messages TTL, i.e., priority is given to new messages.

PrivoSDBC first compares the nodes' weighted similarity to the destination, in a secure manner. Only if the previous are equal is the ego betweenness centrality considered. Therefore, the message is sent first to the most similar node to the destination of the message and then to the more central node, if both nodes have the same similarity.

## 7.2 Simulation model

PRIVO was implemented in the Opportunistic Network Environment (ONE) simulator [KOK09]. Different simulation scenarios consisting of two synthetic mobility models and two real mobility traces were considered. It is assumed here, as in most networks of interest, that there is some social structure between the nodes participating in the network. Each source node generated a new message according to the following intervals: 0.5 to 1 min (0.5-1), 1 to 2 min (1-2), 2 to 4 min (2-4), 4 to 8 min (4-8), 6 to 12 min (6-12) and 8 to 16 min (8-16). The length of the timeslots varied from 5, 10, 15, 30 and 60 min corresponding to 288, 144, 96, 48 and 24 timeslots per day, respectively. Similarly to values normally use in the Round Trip Time (RTT) estimation in the Transport Control Protocol (TCP) [KR12], $\alpha$ and $\beta$ were set to 0.125 and 0.25, respectively.

### 7.2.1 Synthetic mobility models

The simulation time was 7 days with an update interval of 1.0 s. A map-based mobility model of the Helsinki city over an area of 4.5 × 3.4 Km was used. The message size varies from 500 kB to 1 MB. Only two nodes within range can communicate with each other at a time. The communication range between nodes was 10 m, and the communication was bidirectional at a constant transmission rate, for Bluetooth and Wi-Fi interfaces, of 2 Mbit/s and 10 Mbit/s, respectively. From time to time, a source node randomly chosen generated one message to a randomly chosen destination. Two mobility modes were considered:

**Shortest-path Map-Based Movement (SPMBM)**

SPMBM consisted of a network with 40 pedestrians, 20 cars and 6 trams. Pedestrians were moving at a speed varying between 0.8 to 1.4 m/s. Cars and trams were moving at a speed varying between 2.7 to 13.9 m/s. Each time a tram reaches its destination, it paused for 10 to 30 s. The TTL attribute of each message was 5 h. The pedestrians and cars had a buffer size of 10 MB. Trams had a buffer size of 100 MB for DTN traffic.

**Working Day Movement (WDM)**

WDM consisted of a network with 100 pedestrians and 18 buses. There were 50 offices and the working day length was 8 h. The probability of going shopping after work was 50% and there were 10 meeting points. Pedestrians and buses were moving at a speed varying between 0.8 to 1.4 m/s and 7 to 10 m/s, respectively. Each time a bus reaches its destination, it paused for 10 to 30 s. The TTL attribute of each message was 24 h. All nodes had a buffer size of 20 MB for DTN traffic.

### 7.2.2 Real mobility traces

The haggle-one-infocom2005 (INFO5) [Ake16] and taxicabs in Rome (TR) [BBL+14] traces, across different network and mobile environments, were used to provide additional support to the analysis and findings of this paper. In INFO5, 41 iMotes were distributed to students attending Infocom 2005 over 2.97 days. TR contains GPS coordinates of approximately 320 taxicabs collected over 30 days in Rome, Italy. The simulation duration and number of nodes of TR were reduced to 3 days and 304 nodes, respectively. All nodes had a buffer size of 10 MB for DTN traffic. The TTL attribute of each message was 24 h.

## 7.3 Simulation results

In this section, several simulation results describing the performance of PRIVO are presented. For each setting, i.e., protocol-configuration parameter pair, fifteen independent simulations using different message generation seeds were conducted, and the results averaged, for statistical confidence. PRIVO was compared with well-known DTN routing protocols [SCRB16]: two non-social-based routing protocols, namely Epidemic and Prophet, and two social-based routing protocols, namely BubbleRap and dLife [MMS12].

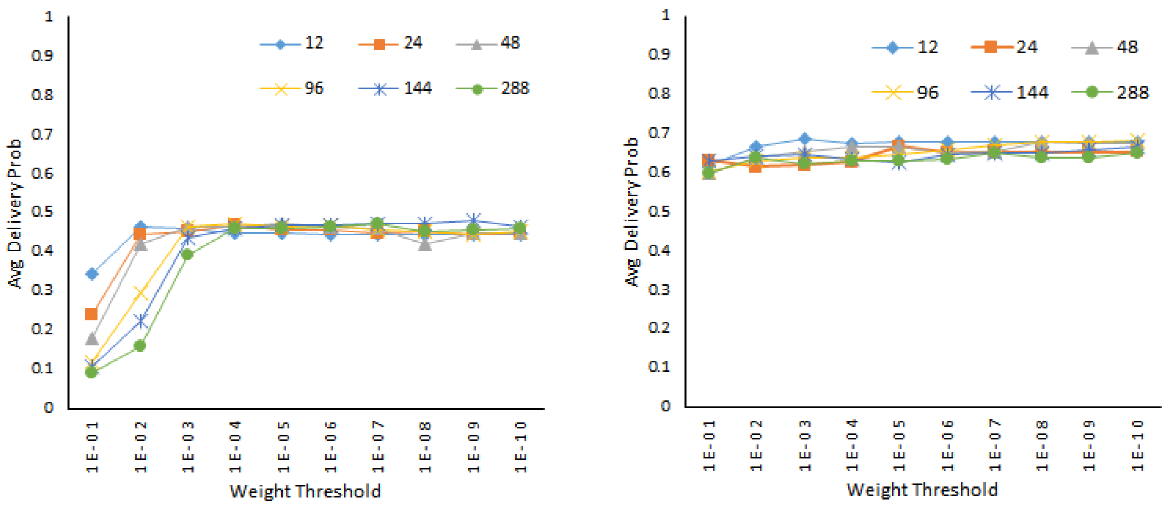
The four variants of PRIVO were considered: PrivoASP, PrivoMTTE, PrivoSDBC and PrivoCOMBINED.

The performance of PRIVO was evaluated according to the following metrics: delivery ratio, overhead ratio and cryptographic cost. The delivery ratio is a key performance indicator as it tells the percentage of successfully received packets of all sent. The overhead ratio is the number of message transmissions for each delivered message. The cryptographic cost, because of homomorphic encryption, gives the computation and transmission cost incurred by cryptographic operations.

In addition, information loss (or data utility) due to the use anonymization methods will also be evaluated. This will be accomplished by analyzing the correlation coefficients between a non-anonymized version of PRIVO and the anonymized ones over the simulations.

### 7.3.1 The selection of the parameters: number of timeslots and weight threshold

This section analyses the selection of two important configuration parameters: number of timeslots ($\eta$) and weight threshold ($\varepsilon$). In Fig. 7.1, the influence of $\eta$ and $\varepsilon$ in PrivoSDBC is analyzed through simulation for a synthetic (WDM) and a real (INFO5) scenario. In these scenarios, source nodes were generating messages every 6 to 12 min, $\varepsilon$ varied from $1 \times 10^{-1}$ to $1 \times 10^{-10}$ and $\eta$ varied from 24 to 288.



(a) WDM

(b) INFO5

Figure 7.1: Delivery ratio of PrivoSDBC for WDM and INFO5 scenarios with $\eta$ varying from 24 to 288 and $\varepsilon$ varying from $1 \times 10^{-1}$ to $1 \times 10^{-10}$

The delivery ratio in all scenarios increased with the reduction of $\varepsilon$ and $\eta$. In general, it starts low as many links are ignored because of $\varepsilon$'s high value and as $\varepsilon$ reduces it increases and tends to stabilize, starting to decrease again as $\varepsilon$ becomes very small (i.e., $\varepsilon \to 0$).

The highest delivery ratio was obtained when $\eta$ was 144 and 96 for WDM and INFO5, respectively. In terms of $\varepsilon$, $1 \times 10^{-7}$ and $1 \times 10^{-10}$ provided the highest delivery ratio for WDM and INFO5, respectively. However, the gains for the same $\eta$ and different $\varepsilon$ were below 3% from $1 \times 10^{-7}$ to $1 \times 10^{-8}$ for WDM and 0.4% from $1 \times 10^{-8}$ to $1 \times 10^{-9}$ for INFO5, if compared to the highest ones.

The weight threshold can be dynamically adjusted in a similar manner to the TCP congestion window [21] but starting at a high value, e.g., $1 \times 10^{-3}$, and gradually reducing it in a time-interval basis. If at the end of a given time interval the delivery ratio increased, then $\varepsilon$ is reduced and vice-versa.

Now, the goal is to make a comparative analysis of PrivoASP and PrivoMTTE in terms of $\eta$ since they do not depend on $\varepsilon$. Fig. 7.2 presents the delivery ratio for PrivoASP, PrivoMTTE in different scenarios.

Generally, two tendencies can be observed from Fig. 7.2 depending on the metric used. On the one hand, if estimates of the average separation periods are considered, the increase of $\eta$ results in a slight increase of the delivery ratio. This is a direct result of having timeslots of smaller length, which offer estimates that are more accurate. PrivoMTTE uses these estimates and its performance slightly increases with the increase of $\eta$ in all scenarios. As previously stated, MTTE allows to identify among all existing timeslots the best ones, that is, the ones with the highest value of average separation period and smallest duration to the next re-encounter, assuming, for example, that nodes' movements obey a certain pattern.

On the other hand, the other PRIVO variants use *pweight*, i.e., an average of the estimates of the average separation period. In this case, two behaviors were observed in Fig. 7.2. The best performance was with $\eta = 288$ and $\eta = 96$ for SPMBM and TR scenarios, respectively.

The increase of $\eta$ also leads to disadvantages as more slots require more storage. Nevertheless, it is possible in all the PRIVO variants except PrivoMTTE to reduce storage by only keeping an estimate of *pweight* that is updated

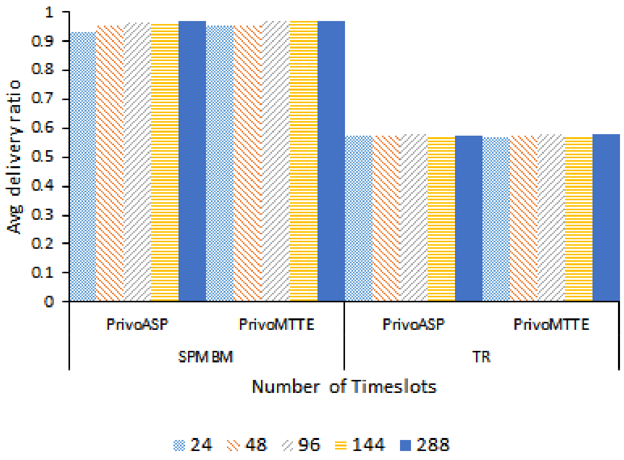


Figure 7.2: Delivery ratio for PrivoASP and PrivoMTTE in SPMBM and TR scenarios with $\eta$ varying from 24 to 288

at the end of each timeslot, therefore not being necessary to keep estimates of the average separation period for each timeslot.

### 7.3.2 Routing performance

This section analyses PRIVO's routing performance without the use of homomorphic encryption. Based on the previous section, $\eta$ and $\varepsilon$ were set, respectively, to $144$ and $1 \times 10^{-8}$ for all PRIVO variants.

Fig. 7.3 presents the average delivery ratio and overhead ratio for different scenarios, routing protocols and message generation rates. As expected, with the decrease of the data rate there is an increase in delivery ratio and a decrease of the overhead ratio as fewer messages circulated in the network.

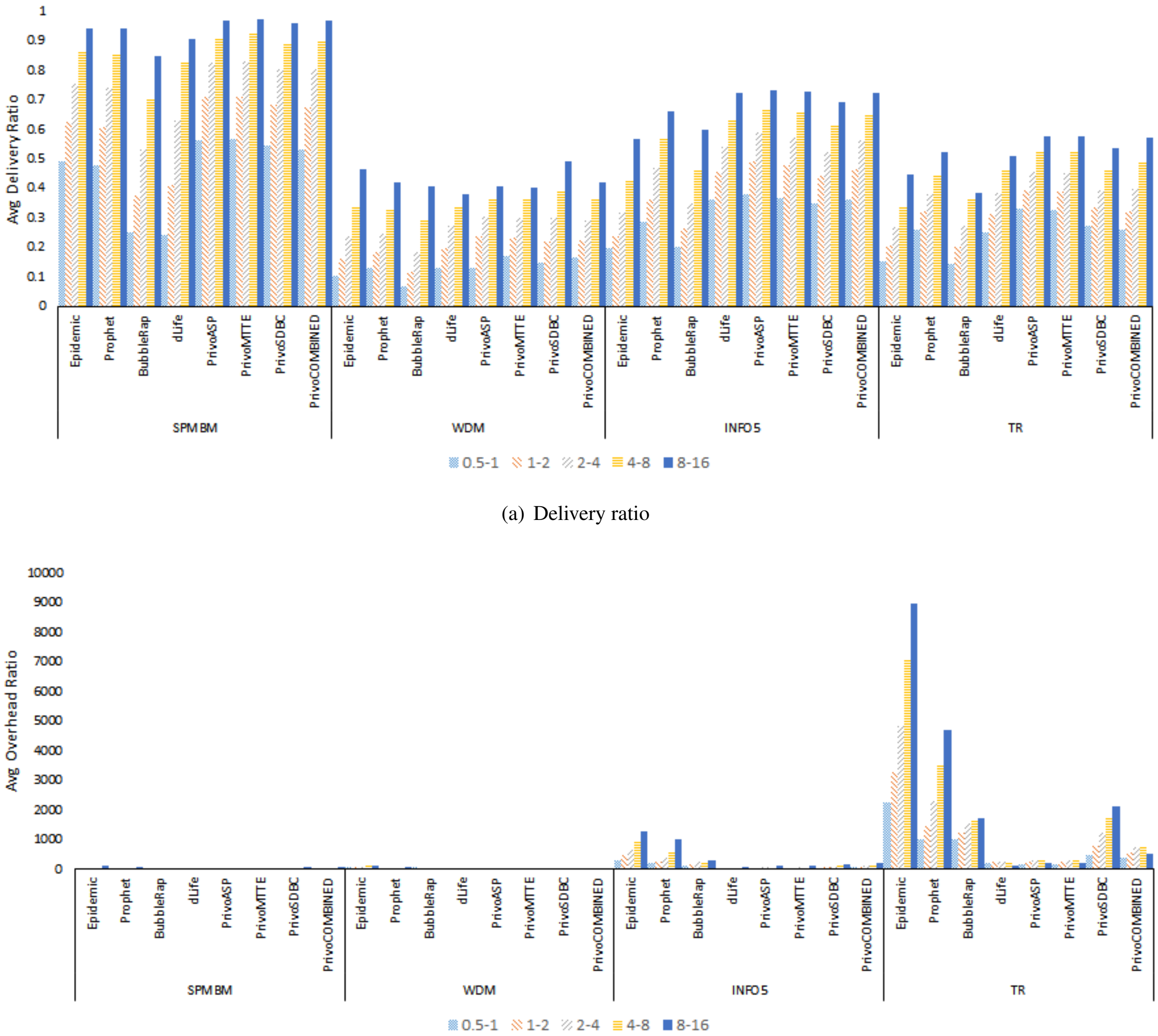
Overall, PRIVO performed better than other routing protocols in all message generation rates and scenarios in terms of delivery and overhead ratios. However, the performance of each PRIVO variant depends on the scenario. The routing protocols that presented the highest delivery ratio were PrivoMTTE for SPMBM and TR, PrivoASP for INFO5 and PrivoSDBC for WDM. The maximum gains obtained were 14.6%, 29.9%, 29.8% and 49.5% for SPMBM, WDM, INFO5 and TR, respectively. Among the non-PRIVO routing protocols, the ones that presented the highest delivery ratios were Epidemic for SPMBM and WDM, dLife for INFO5 and Prophet for TR. Therefore, if there are some repetitive movement patterns then PrivoSDBC is the best choice otherwise, it is PrivoMTTE.

### 7.3.3 Cryptographic costs

This section analyses the cryptographic cost of using the Paillier homomorphic encryption scheme.

#### Additive homomorphic encryption

A set of experiments were performed to evaluate the performance of additive homomorphic encryption using the Paillier cryptosystem. The experiments were performed in a personal computer with the following specifications: Intel® CORE™ i7-2600 CPU @ 3.40GHz, 16 GB RAM and Windows 10 Pro (64-bits). Table 7.2 presents



(a) Delivery ratio

(b) Overhead ratio

Figure 7.3: Delivery and overhead ratios for all the routing protocols considered in different scenarios for different message generation rates

the average Paillier execution time of five operations, namely encryption $\mathcal{E}(a)$, decryption $\mathcal{D}(c)$, sum $\mathcal{E}(a+b)$, difference $\mathcal{E}(a-b)$ and multiplication by a constant $\mathcal{E}(k \cdot a)$. The difference is performed by multiplying the second term by -1 followed by summing the numbers, therefore being slower than sum and multiplication by a constant. The operations were repeated 100 times.

| Key Size | $\mathcal{E}(a)$ | $\mathcal{D}(c)$ | $\mathcal{E}(a+b)$ | $\mathcal{E}(a-b)$ | $\mathcal{E}(k \cdot a)$ |
|---|---|---|---|---|---|
| 512 | 1.73 ± 0.0342 | 1.74 ± 0.0314 | 0.01 ± 0.0005 | 0.38 ± 0.02 | 0.02 ± 0.0016 |
| 1024 | 11.03 ± 0.1261 | 11.29 ± 0.3552 | 0.03 ± 0.0019 | 0.74 ± 0.0425 | 0.05 ± 0.0023 |
| 2048 | 83.49 ± 0.3029 | 83.9 ± 0.4546 | 0.06 ± 0.0033 | 1.74 ± 0.0719 | 0.14 ± 0.0038 |

Table 7.2: Average Paillier execution times (ms)

**PRIVO's performance with Paillier**

Now, messages were generated every 6 to 12 min. Table 7.3 presents PRIVO's average delivery ratio losses (+) and gains (-) using the Paillier cryptosystem with key sizes of 512, 1024 and 2048 bits for $\eta = 144$. Each table entry results from averaging losses and gains of all PRIVO variants per key. From Table 7.3, it is possible to see that the losses are below or equal to 1% in all scenarios, with exception of WDM, therefore the use of the Paillier homomorphic encryption scheme was not considered in the previous subsection (Section 7.1.2).

A more detailed analysis was performed for the legacy key size (i.e., 1024 bits) [28]. Table 7.4 presents the average delivery ratio losses (+) and gains (-) using the Paillier cryptosystem for $\eta = 144$.

It was concluded, based on simulation results, that if a message was not transmitted because of the additional delay caused by homomorphic encryption, it would be transmitted later on. In some cases (see Table 7.3 and Table 7.4), this additional delay is beneficial to the routing protocol, as it may contribute to the reduction of the network load, even though the maximum achieved gains being negligible (at most 0.50% for the legacy key).

| Scenarios | Key size (bits) | | |
|---|---|---|---|
| | 512 | 1024 | 2048 |
| SPMBM | -0.08 | -0.01 | 1.01 |
| WDM | 1.11 | 4.77 | 21.73 |
| INFO5 | 0.00 | -0.21 | -0.09 |
| TR | -0.11 | -0.06 | -0.88 |

Table 7.3: Average delivery ratio losses and gains using the Paillier cryptosystem (%)

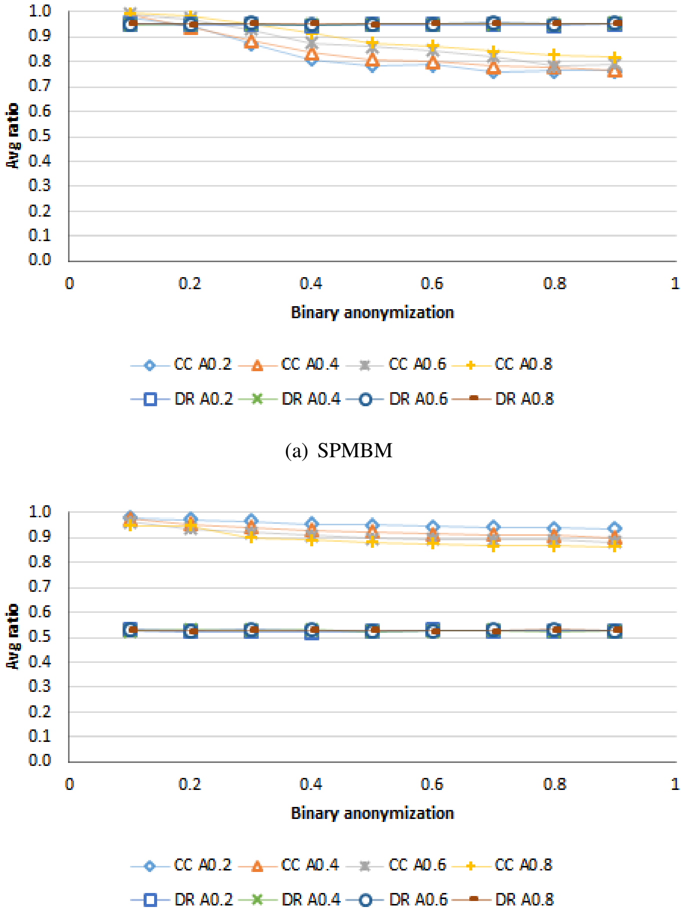
| Privo Variants | Scenarios | | | |
|---|---|---|---|---|
| | SPMBM | WDM | INFO5 | TR |
| PrivoASP | -0.04 | 4.61 | -0.06 | -0.09 |
| PrivoMTTE | 0.34 | 4.05 | 0.04 | 0.14 |
| PrivoSDBC | -0.17 | 5.72 | -0.50 | -0.35 |
| PrivoCOMBINED | -0.15 | 4.69 | -0.32 | 0.06 |

Table 7.4: Average delivery ratio losses and gains using the Paillier cryptosystem with 1024Bits key (%)

### 7.3.4 Information loss

This section analyses the utility of the data (or information loss) because of the use of anonymization methods. Information loss is measured comparing the correlation coefficients [Fis15] of the ego betweenness centrality values of all the nodes in the simulation with and without anonymization. The ego betweenness values were collected at the end of each day and the values were compared for different percentages of total anonymization with the case were no anonymization was used. Total anonymization corresponds to the total number of nodes in the neighboring graph that are anonymized. Binary anonymization was applied over a percentage of the latter. At the end of each simulation, the correlation coefficients were averaged taking into account the number of days of the simulation. Different percentages of binary and total anonymization were used. The former varied from 10% to 90% with increments of 10% and the latter varied from 20% to 80% with increments of 20%.

Fig. 7.4 presents the average correlation coefficient (CC) and delivery ratio (DR) for PrivoSDBC in SPMBM and TR scenarios. Between binary anonymization and neighborhood randomization, the former is the one to cause



(a) SPMBM

(b) TR

Figure 7.4: Average correlation coefficient (CC) and delivery ratio (DR) for PrivoSDBC in SPMBM and TR scenarios for different percentages of total (A) and binary anonymizations

a reduction on the average correlation coefficients as it increases, and this effect worsens as the percentage of total anonymization increases. Nonetheless, since PrivoSDBC uses ego betweenness centrality and weighted similarity to the destination and the latter is more frequently used as a routing metric, the effects of the lowest values of correlation coefficients (i.e., 0.82 for SPMBM and 0.86 for TR corresponding to 90% of binary anonymization and 80% of total anonymization) are not significant as can be seen by the steady average delivery ratio in Fig. 7.4.

## 7.4 Summary

This chapter proposed PRIVO, a PRIvacy-preserVing Opportunistic routing protocol for DTNs. PRIVO ensures link privacy by means of binary anonymization and neighborhood randomization, and attribute privacy by means of the Paillier homomorphic encryption scheme.

The effectiveness of PRIVO is supported through extensive simulations, with synthetic mobility models and real mobility traces, taking into account routing metrics such as delivery and overhead ratios, cryptographic costs and information loss.

Simulations results show that PRIVO presents on average cryptographic costs below 1% in most scenarios (i.e., SPMBM, INFO5 and TR). If there are some repetitive movement patterns as in WDM, then PrivoSDBC is the best choice. Otherwise, it is PrivoMTTE. Furthermore, PRIVO presents on average gains of 22.2% and 39.7% in terms of delivery ratio for the scenarios considered if compared with non-social-based (Epidemic and Prophet) and social-based routing protocols (BubbleRap and dLife), respectively.

# Chapter 8

# Conclusions

This chapter presents concluding remarks and future work.

## 8.1 Achievements

The objectives of this thesis were to develop and implement efficient routing protocols for WANETs. In particular, the proposed approaches should: satisfy WMSNs QoS requirements such as throughput, delay, energy efficiency; and, use the available network information in DTNs to find the most suitable next node to forward messages to. In addition, the design and implementation of robust and distributed reputation system as well as secure routing protocol for DTNs was also envisioned.

A high throughput low coupling multipath routing protocol for WMSNs, known as HTLC-MeDSR, was proposed and evaluated through simulation on Chapter 3. In HTLC-MeDSR, at the routing layer, nodes use ETX together with the first and second degree correlation factors between paths to find multiple high throughput paths with minimal interference between them. At the MAC layer, a modified value of SRL is used. In addition, probe packets are used to identify routing failures, which reduce packet loss and unnecessary route maintenance operations. Simulation results have shown that HTLC-MeDSR presents considerable gains in comparison to other routing protocols in terms of the routing metrics' considered, namely throughput, delay, packet loss, control overhead, energy efficiency and path length.

On Chapter 4, a multi-objective optimization approach for the WMSN routing problem that takes into account QoS parameters such as delay and ETX was proposed. A multi-objective optimization algorithm aims at producing (1) solutions as close as possible to the Pareto-optimal set, and (2) solutions as diverse as possible in the obtained non-dominated set. This diverse set of optimal solutions expresses trade-offs between different objectives. Simulation results have shown that the proposed MOEA found paths that were at least 2 and 4 times better in terms of ETX and end-to-end delay that those found by HTLC-MeDSR, respectively.

Chapter 5 presented a study of the impact of misbehaving nodes on several representative DTN routing protocols for three types of misbehavior. Simulation results have shown that the delivery probability of DTN routing protocols depends on two factors: (1) the contact characteristics provided by the mobility model and the topology; (2) the type of misbehavior. With Type I misbehavior, Maxprop and Rapid presented the best results in both scenarios followed by Epidemic and Prophet. With Type II misbehavior, Prophet and Epidemic were the best for scenario 1, and both versions of Spray and Wait were the best for scenario 2. With Type III misbehavior, Maxprop and Rapid were the best followed by Epidemic and Prophet. However, First Contact was the routing protocol most affected by misbehavior because it is single-copy. Moreover, the routing protocols considered were classified in terms of two metrics, "*-copy" and "estimation-based". In terms of the first, clearly the best protocols were the unlimited-copy ones' and the worst was FC that is single-copy, with Spray and Wait in the middle. In relation to the second metric, estimation-based protocols are apparently better. Epidemic is not estimation-based and fares well, but it is also a brute-force protocol that does flooding therefore having the highest overhead.

Chapter 6 presented the design of a reputation system for DTNs that is both robust against false ratings and efficient at detecting nodes' misbehavior. The proposed reputation system makes use of all the available information, i.e., first and second hand. It is based on a Bayesian approach that uses the Beta distribution and can be integrated with any DTN routing protocol. Simulation results show that the system is able, for the node's classification problem, to classify correctly on the fly liars that collude using the maximum likelihood decision criterion, and, for the node's trustworthiness problem, it is also able to classify correctly nodes in most cases. Moreover, there are tradeoffs in this system. For instance, if the evidence's TTL is too high, the reputation system will take more time to converge as the detection time increases. On the other hand, if the TTL is too small, proofs of relay might not have enough time to be effectively disseminated over the network, which will increase the number of misclassifications. On the other hand, by attempting to isolate misbehaving nodes, good nodes that up to a given instant only accepted messages will be also misclassified, therefore increasing the ratio of false negatives.

A PRIvacy-preserVing Opportunistic routing protocol for DTNs, known as PRIVO, was proposed on Chapter 7. PRIVO models a DTN as a time-varying neighboring graph where edges correspond to the neighboring relationship among pairs of nodes. It ensures privacy by protecting each node's sensitive information even if it has to be processed elsewhere. The effectiveness of PRIVO was supported through extensive simulations with synthetic mobility models as well as real mobility traces. Simulations results have shown that PRIVO presents on average very low cryptographic costs in most scenarios, and if there were some repetitive movement patterns then PrivoSDBC was the best choice, otherwise it was PrivoMTTE. In general, PRIVO outperformed other routing protocols in terms of delivery ratio, for the scenarios considered.

## 8.2 Future work

The proposed efficient multipath routing protocol for WMSNs, i.e., HTLC-MeDSR, was targeted to networks with low mobility, which is expected to be the most common situation in WMSNs. An evaluation and improvement of the stability of paths and the protocol's overhead in face of failures and node mobility was left for future work. Other topics left for future work are: using the received signal strength indicator (RSSI) to complement ETX since having a low ETX value does not mean that the nodes are close to each other, and the distance among nodes has direct impact on the achievable throughput; adding priority schemes, hence allowing the protocol to adjust its behavior according to the QoS requirements that multimedia traffic (in real time or not) may have; and, scalability by analyzing the impact of different compression schemes on HTLC-MeDSR.

HTLC-MeDSR uses first a shortest-path finding algorithm and then a sorting metric to find the best pairs of paths. Instead of the previous mechanism, the MOEA algorithm proposed on Chapter 4 could be used in HTLC-MeDSR route set's building procedure. This modification and the subsequent evaluation was left for future work. Despite the good results presented in Chapter 4, SPEA still has some potential weaknesses [ZLT01]. Using better MOEA algorithms such as NSGA-II and SPEA2 was left for future work. Moreover, to preserve diversity in non-dominated sets, concepts such as weight mapping crossover based recombination and dynamic crowding distances will be applied.

The evaluation of DTN routing protocols integrated with the reputation system, proposed on Chapter 6, in terms of routing metrics, such as delivery probability, buffer time, routing protocol overhead was also left for future work. The analysis of the detection times of more elaborate attacks such as grey-hole attacks where nodes alternate their behavior between good and bad thus making more difficult to detect them was also left for future work. Other ideas for future work are: the study of the convergence speed of distributed algorithms in stochastic networks and its application to reduce detection time of distributed reputation systems in DTNs; to consider more challenging attacks, such as collusion, since they have not been extensively studied in DTNs. One idea is to use novel approaches proposed in literature, such as GlobalTrust [CCZ14], that addresses this issue in DTNs. In addition, besides Bayesian inference, other statistical learning concepts could be leveraged by the distributed reputation engine. For example, for reputation representation, instead of graphical models such Bayesian networks, probabilistic logic (e.g., Markov logic networks) and weighted formulas could be used. Posterior probability or user-defined objective functions depending on the goal could be used for reputation evaluation. And last, but not least, for optimization, formula discovery and weighted learning could be used. The combination of all of the previous statistical learning concepts in one unique learning algorithm is yet under research.

Finally, concerning privacy-preserving schemes for DTNs, the combination of two cryptographic mechanisms, namely identity-based encryption (IBE) [BF01], which is more suitable for DTNs, and homomorphic encryption, known as identity-based fully homomorphic encryption [CM14] was left for future work. IBE is a cryptographic method that enables message encryption and signature verification using a public identifier (e.g., an email address) of the target node as a key. An IBE system consists of message senders and recipients, i.e., principals, and a TTP commonly known as the Private Key Generator. Another open research issue related to IBE is how to obtain new keys. Additionally, applying privacy-preserving schemes such as PRIVO in a real testbed, e.g., a mobile social network composed of two or more smartphones, was left for future work.

# Bibliography

[AA21] Floyd H Allport and Gordon W Allport. Personality Traits: Their Classification and Measurement. *The Journal of Abnormal Psychology and Social Psychology*, 16(1):6, 1921.

[AB09] Kartik Anand and Ginestra Bianconi. Entropy measures for networks: Toward an information theory of complex topologies. *Physical Review E*, 80:045102, Oct 2009.

[AE03] Luzi Anderegg and Stephan Eidenbenz. Ad hoc-VCG: A Truthful and Cost-efficient Routing Protocol for Mobile Ad Hoc Networks with Selfish Agents. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, MobiCom '03, pages 245–259, New York, NY, USA, 2003. ACM.

[AF12] Erman Ayday and Faramarz Fekri. An Iterative Algorithm for Trust Management and Adversary Detection for Delay-Tolerant Networks. *IEEE Transactions on Mobile Computing*, 11(9):1514–1531, Sep 2012.

[Ake16] Dimitrios-Georgios Akestoridis. CRAWDAD dataset uoi/haggle (v. 2016-08-28): derived from cambridge/haggle (v. 2009-05-29). Downloaded from `http://crawdad.org/uoi/haggle/20160828`, August 2016.

[AKK04] Jamal N Al-Karaki and Ahmed E Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6):6–28, Dec 2004.

[ALPH01] Lada A. Adamic, Rajan M. Lukose, Amit R. Puniyani, and Bernardo A. Huberman. Search in power-law networks. *Physical Review E*, 64(4):046135, 2001.

[ALRL04] Algirdas Avizienis, J-C Laprie, Brian Randell, and Carl Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1):11–33, 2004.

[ALV$^+$09] Jó Ágila, Bitsch Link, Nicolai Viol, André Goliath, Klaus Wehrle, and Bitsch Link. SimBetAge : Utilizing Temporal Changes in Social Networks for Pocket Switched Networks. In *Proceedings of the 1st ACM workshop on User-provided networking: challenges and opportunities - U-NET '09*, pages 13–18, New York, USA, December 2009. ACM Press.

[AMC07] Ian F. Akyildiz, Tommaso Melodia, and Kaushik R. Chowdhury. A survey on wireless multimedia sensor networks. *Computer Networks*, 51(4):921 – 960, 2007.

[Ant71] Jac M Anthonisse. The rush in a directed graph. *Stichting Mathematisch Centrum. Mathematische Besliskunde*, BN 9/71:1–10, 1971.

[ASSC02] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393 – 422, 2002.

[AY05] Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3(3):325 – 349, 2005.

[BBC$^+$01] Ljubica Blazevic, Levente Buttyan, Srdjan Capkun, Silvia Giordano, J-P Hubaux, and J-Y Le Boudec. Self organization in mobile ad hoc networks: the approach of Terminodes. *IEEE Communications Magazine*, 39(6):166–174, 2001.

[BBL$^+$14] Lorenzo Bracciale, Marco Bonola, Pierpaolo Loreti, Giuseppe Bianchi, Raul Amici, and Antonello Rabuffi. CRAWDAD dataset roma/taxi (v. 2014-07-17). Downloaded from `http://crawdad.org/roma/taxi/20140717`, July 2014.

[BBQ$^+$05] Roberto Baldoni, Roberto Beraldi, Leonardo Querzoni, Gianpaolo Cugola, and Matteo Migliavacca. Content-based routing in highly dynamic mobile ad hoc networks. *International Journal of Pervasive Computing and Communications*, 1(4):277–288, November 2005.

[BDD$^+$05] Eric Brewer, Michael Demmer, Bowei Du, Melissa Ho, Matthew Kam, Sergiu Nedevschi, Joyojeet Pal, Rabin Patra, Sonesh Surana, and Kevin Fall. The case for technology in developing regions. *Computer*, 38(6):25–38, 2005.

[BDFV10] Levente Buttyán, László Dóra, Márk Félegyházi, and István Vajda. Barter trade improves message delivery in opportunistic networks. *Ad Hoc Networks*, 8(1):1–14, Jan 2010.

[BDT99] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *From Natural to Artificial Swarm Intelligence*. Oxford University Press, September 1999.

[BE05] Ulrik Brandes and Thomas Erlebach. *Network Analysis: Methodological Foundations*. Springer Science & Business Media, 2005.

[BE06] Stephen P. Borgatti and Martin G. Everett. A Graph-theoretic perspective on centrality. *Social Networks*, 28(4):466–484, October 2006.

[Ber13] James O. Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.

[BF01] Dan Boneh and Matt Franklin. Identity-Based Encryption from the Weil Pairing. In Joe Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer Berlin Heidelberg, 2001.

[BFL96] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *IEEE Symposium on Security and Privacy*, pages 164–173. Security and Privacy, 1996.

[BGJL06] John Burgess, Brian Gallagher, David Jensen, and Brian Neil Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *IEEE INFOCOM 2006*, volume 6, pages 1–11. IEEE, 2006.

[BH00] Levente Buttyán and Jean-Pierre Hubaux. Enforcing service availability in mobile ad-hoc WANs. In *Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking & Computing*, pages 87–96. IEEE Press, 2000.

[BH03] Levente Buttyán and Jean-Pierre Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *Mobile Networks and Applications*, 8(5):579–592, 2003.

[BH11] Greg Bigwood and Tristan Henderson. IRONMAN: Using Social Networks to Add Incentives and Reputation to Opportunistic Networks. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pages 65–72, Oct 2011.

[Bia00] Giuseppe Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, Mar 2000.

[BKMM07] David A Bader, Shiva Kintali, Kamesh Madduri, and Milena Mihail. Approximating Betweenness Centrality. *Algorithms and Models for the Web-Graph*, pages 124–137, 2007.

[BL04] Sonja Buchegger and Jean-Yves Le Boudec. A Robust Reputation System for Peer-to-Peer and Mobile Ad-hoc Networks. In *P2PEcon 2004*, 2004.

[BLO$^+$14] Chiara Boldrini, Kyunghan Lee, Melek Önen, Jörg Ott, and Elena Pagani. Opportunistic networks. *Computer Communications*, 48:1–4, July 2014.

[BLV07] Aruna Balasubramanian, Brian Levine, and Arun Venkataramani. DTN routing as a resource allocation problem. *ACM SIGCOMM Computer Communication Review*, 37(4):373–384, 2007.

[BP07] Ulrik Brandes and Christian Pich. Centrality Estimation In Large Networks. *International Journal of Bifurcation and Chaos*, 17(07):2303–2318, July 2007.

[Bra01] Ulrik Brandes. A faster algorithm for betweenness centrality. *The Journal of Mathematical Sociology*, 25(2):163–177, June 2001.

[Bra08] Ulrik Brandes. On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, 30(2):136–145, May 2008.

[BS10] Eyuphan Bulut and Boleslaw K. Szymanski. Friendship Based Routing in Delay Tolerant Mobile Social Networks. In *2010 IEEE Global Telecommunications Conference (GLOBECOM 2010)*, pages 1–5. IEEE, December 2010.

[Bur95] Ronald S. Burt. *Structural Holes: The Social Structure of Competition*. Harvard University Press, 1995.

[CABM05] Douglas Couto, Daniel Aguayo, John Bicket, and Robert Morris. A High-throughput Path Metric for Multi-hop Wireless Routing. *Wireless Networks*, 11(4):419–434, July 2005.

[CC10] Bin Bin Chen and Mun Choon Chan. MobiCent: a Credit-Based Incentive System for Disruption Tolerant Network. In *IEEE INFOCOM 2010*, pages 1–9. IEEE, 2010.

[CCZ14] Xin Chen, Jin-Hee Cho, and Sencun Zhu. GlobalTrust: An attack-resilient reputation system for tactical networks. In *2014 Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 275–283, Jun 2014.

[CFQS10] Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-Varying Graphs and Dynamic Networks. In *Ad-hoc, Mobile, and Wireless Networks*, pages 1–20. Springer Berlin Heidelberg, November 2010.

[CHC$^+$07] Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Impact of Human Mobility on Opportunistic Forwarding Algorithms. *IEEE Transactions on Mobile Computing*, 6(6):606–620, June 2007.

[CKSS02] Tami Carpenter, George Karakostas, David Shallcross, and South Street. Practical Issues and Algorithms for Analyzing Terrorist Networks. In *Proceedings of the Western Simulation MultiConference*, 2002.

[CLRS09] Thomas H. Cormen, Charles Eric Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. The MIT Press, 3rd edition, 2009.

[CLWW16] Honglong Chen, Wei Lou, Zhibo Wang, and Qian Wang. A Secure Credit-Based Incentive Mechanism for Message Forwarding in Noncooperative DTNs. *IEEE Transactions on Vehicular Technology*, 65(8):6377–6388, Aug 2016.

[CM14] Michael Clear and Ciarán McGoldrick. *Bootstrappable Identity-Based Fully Homomorphic Encryption*, pages 1–19. Springer International Publishing, Cham, 2014.

[CMS12] Gianpiero Costantino, Fabio Martinelli, and Paolo Santi. Privacy-preserving interest-casting in opportunistic networks. In *IEEE Wireless Communications and Networking Conference, WCNC*, pages 2829–2834. IEEE, Apr 2012.

[COC11] Miguel Camelo, C Omaña, and Harold Castro. QoS routing algorithms based on multi-objective optimization for mesh networks. *IEEE Latin America Transactions*, 9(5):875–881, Sep 2011.

[Coo01] Karen Cook. *Trust in society*. Russell Sage Foundation, 2001.

[CP11] Marcello Caleffi and Luigi Paura. M-DART: multi-path dynamic address routing. *Wireless Communications and Mobile Computing*, 11(3):392–409, 2011.

[CRVW04] Shih-Fen Cheng, Daniel M Reeves, Yevgeniy Vorobeychik, and Michael P Wellman. Notes on equilibria in symmetric games. In *Proceedings of Workshop on Game Theory and Decision Theory*, 2004.

[CS13] Yue Cao and Zhili Sun. Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. *IEEE Communications Surveys & Tutorials*, 15(2):654–677, 2013.

[CSC11] Jin-Hee Cho, Ananthram Swami, and Ray Chen. A survey on trust management for mobile ad hoc networks. *IEEE Communications Surveys & Tutorials*, 13(4):562–583, 2011.

[CSEJ$^+$07a] Qi Chen, Felix Schmidt-Eisenlohr, Daniel Jiang, Marc Torrent-Moreno, Luca Delgrossi, and Hannes Hartenstein. Overhaul of IEEE 802.11 Modeling and Simulation in NS-2. In *Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, MSWiM '07, pages 159–168, New York, NY, USA, 2007. ACM.

[CSEJ$^+$07b] Qi Chen, Felix Schmidt-Eisenlohr, Daniel Jiang, Marc Torrent-Moreno, Luca Delgrossi, and Hannes Hartenstein. Overhaul of IEEE 802.11 Modeling and Simulation in NS-2. In *Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, MSWiM '07, pages 159–168, New York, NY, USA, 2007. ACM.

[CSY15] Kang Chen, Haiying Shen, and Li Yan. Multicent: A Multifunctional Incentive Scheme Adaptive to Diverse Performance Objectives for DTN Routing. *IEEE Transactions on Parallel and Distributed Systems*, 26(6):1643–1653, Jun 2015.

[CT12] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[Dav03] Anthony Christopher Davison. *Statistical Models*. Cambridge University Press, 2003.

[DCABM05] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A High-throughput Path Metric for Multi-hop Wireless Routing. *Wireless Networks*, 11(4):419–434, July 2005.

[DD99] Marco Dorigo and Gianni Di Caro. The ant colony optimization meta-heuristic. In David Corne, Marco Dorigo, Fred Glover, Dipankar Dasgupta, Pablo Moscato, Riccardo Poli, and Kenneth V. Price, editors, *New Ideas in Optimization*, pages 11–32. McGraw-Hill Ltd., UK, Maidenhead, UK, England, January 1999.

[DD12] Gianluca Dini and Angelica Lo Duca. Towards a reputation-based routing protocol to contrast blackholes in a delay tolerant network. *Ad Hoc Networks*, 10(7):1167–1178, Sep 2012.

[Deb08] Kalyanmoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley paperback series. Wiley, 2008.

[DF10] Yezid Donoso and Ramon Fabregat. *Multi-objective optimization in computer networks using metaheuristics*. CRC Press, 2010.

[DH07] Elizabeth M. Daly and Mads Haahr. Social Network Analysis for Routing in Disconnected Delay-tolerant MANETs. In *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '07, pages 32–40, New York, NY, USA, 2007. ACM.

[DI01] Camil Demetrescu and Giuseppe F. Italiano. Fully dynamic all pairs shortest paths with real edge weights. In *Proceedings 2001 IEEE International Conference on Cluster Computing*, pages 260–267. IEEE Comput. Soc, 2001.

[DI04] Camil Demetrescu and Giuseppe F. Italiano. A new approach to dynamic all pairs shortest paths. *Journal of the ACM*, 51(6):968–992, November 2004.

[Dij59] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, December 1959.

[DL16] Franz Dietrich and Christian List. Probabilistic Opinion Pooling. In Alan Hájek and Christopher Hitchcock, editors, *The Oxford Handbook of Probability and Philosophy*, chapter 25, page 832. Oxford University Press, 2016.

[DMW03] Peter Sheridan Dodds, Roby Muhamad, and Duncan J Watts. An experimental study of search in global social networks. *Science (New York)*, 301(5634):827–9, August 2003.

[DPAM02] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, Apr 2002.

[DRXM15] J. A. F. F. Dias, J. J. P. C. Rodrigues, F. Xia, and C. X. Mavromoustakis. A cooperative watchdog system to detect misbehavior nodes in vehicular delay-tolerant networks. *IEEE Transactions on Industrial Electronics*, 62(12):7929–7937, Dec 2015.

[EB05] Martin Everett and Stephen P. Borgatti. Ego network betweenness. *Social Networks*, 27(1):31–38, January 2005.

[ECCD08] Vijay Erramilli, Augustin Chaintreau, Mark Crovella, and Christophe Diot. Delegation forwarding. In *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, page 251, New York, USA,, 2008. ACM Press.

[EH12] Samina Ehsan and Bechir Hamdaoui. A Survey on Energy-Efficient Routing Techniques with QoS Assurances for Wireless Multimedia Sensor Networks. *IEEE Communications Surveys & Tutorials*, 14(2):265–278, 2012.

[EP05] Nathan Eagle and Alex (Sandy) Pentland. CRAWDAD dataset mit/reality (v. 2005-07-01). Downloaded from `http://crawdad.org/mit/reality/20050701`, July 2005.

[Epp04] David Eppstein. Fast Approximation of Centrality. *Graph Algorithms and Applications*, 8(1):39–45, 2004.

[FBW91] Linton C. Freeman, Stephen P. Borgatti, and Douglas R. White. Centrality in valued graphs: A measure of betweenness based on network flow. *Social Networks*, 13(2):141–154, June 1991.

[FES03] Jonathan E Fieldsend, Richard M Everson, and Sameer Singh. Using unconstrained elite archives for multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 7(3):305–323, Jun 2003.

[FF87] Lester R Ford and Delbert R Fulkerson. Maximal Flow Through a Network. In Gian-Carlo Gessel, Ira and Rota, editor, *Classic Papers in Combinatorics*, pages 243–248. Birkhäuser Boston, 1987.

[Fig04] Mário A. T. Figueiredo. Lecture notes on Bayesian estimation and classification. Technical report, Instituto de Telecomunicações, Instituto Superior Técnico, Lisboa, 2004.

[Fis15] Ronald A Fisher. Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika*, 10(4):507–521, May 1915.

[FL01] James A. Freebersyser and Barry Leiner. A DoD Perspective on Mobile Ad Hoc Networks. In *Ad Hoc Networking*, pages 29–51. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[FNP04] Michael J. Freedman, Kobbi Nissim, and Benny Pinkas. Efficient Private Matching and Set Intersection. In Christian Cachin and JanL. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 1–19. Springer Berlin Heidelberg, 2004.

[Fre77a] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[Fre77b] Richard Broke Freeman. *The Works of Charles Darwin: An Annotated Bibliographical Handlist*. Archon Books, 2nd edition, 1977.

[Fre78] Linton C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239, January 1978.

[FSB$^+$07] Kevin Fall, Keith L Scott, Scott C Burleigh, Leigh Torgerson, Adrian J Hooke, Howard S Weiss, Robert C Durst, and Vint Cerf. Delay-Tolerant Networking Architecture. Technical report, RFC 4838, 2007.

[FT87] Michael L. Fredman and Robert E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, 34(3):596–615, 1987.

[FV11] Kevin Fall and Kannan Varadhan. The ns manual (formerly ns notes and documentation), November 2011. http://www.isi.edu/nsnam/ns/ns-documentation.html.

[GA08] Anargyros Garyfalos and Kevin C. Almeroth. Coupons: A Multilevel Incentive Scheme for Information Dissemination in Mobile Networks. *IEEE Transactions on Mobile Computing*, 7(6):792–804, Jun 2008.

[GADP89] Simon Goss, Serge Aron, Jean-Louis Deneubourg, and Jacques Marie Pasteels. Self-organized shortcuts in the Argentine ant. *Naturwissenschaften*, 76(12):579–581, December 1989.

[Gar01] Joel Garreau. Disconnect the dots, 2001. http://www.seity.com/wp-content/themes/seity/pdf/disconnect the dots.pdf.

[GC10] Wei Gao and Guohong Cao. Fine-grained Mobility Characterization: Steady and Transient State Behaviors. In *Proceedings of the Eleventh ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '10, pages 61–70, New York, USA, 2010. ACM.

[GCPR14] Barbara Guidi, Marco Conti, Andrea Passarella, and Laura Ricci. Distributed protocols for Ego Betweenness Centrality computation in DOSNs. In *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*, pages 539–544, Mar 2014.

[GSK15] Nikhil Gondaliya, Mehul Shah, and Dhaval Kathiriya. A node scheduling approach in community based routing in social delay tolerant networks. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 594–600, Aug 2015.

[GSS08] Robert Geisberger, Peter Sanders, and Dominik Schultes. Better Approximation of Betweenness Centrality. *ALENEX*, pages 90–100, 2008.

[HCS$^+$05] Pan Hui, Augustin Chaintreau, James Scott, Richard Gass, Jon Crowcroft, and Christophe Diot. Pocket switched networks and human mobility in conference environments. In *Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking - WDTN '05*, pages 244–251, New York, USA,, 2005. ACM SIGCOMM workshop on Delay-tolerant networking - WDTN, ACM Press.

[HCY11] Pan Hui, Jon Crowcroft, and Eiko Yoneki. BUBBLE Rap: Social-Based Forwarding in Delay-Tolerant Networks. *IEEE Transactions on Mobile Computing*, 10(11):1576–1589, November 2011.

[HEZ15] Faten Hajjej, Ridha Ejbali, and Mourad Zaied. Quality of Services based routing using evolutionary algorithms for Wireless Sensor Network. In *2015 15th International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 237–242, Dec 2015.

[HHBL16] Jianhui Huang, Qin Hu, Jingping Bi, and Zhongcheng Li. Stackelberg game based incentive mechanism for data transmission in mobile opportunistic networks. In *Wireless Algorithms, Systems, and Applications: 11th International Conference, WASA 2016, Bozeman, MT, USA, Aug 8-10, 2016. Proceedings*, pages 377–388. Springer International Publishing, Cham, 2016.

[HHM08] John Michael Harris, Jeffry L Hirst, and Michael J Mossinghoff. *Combinatorics and graph theory*. Springer Berlin Heidelberg, New York, USA, 2nd edition, 2008.

[Hin11] Max Hinne. *Local approximation of centrality measures*. PhD thesis, Radboud University Nijmegen, The Netherlands, 2011.

[HKRZ08] Woochang Hwang, Taehyong Kim, Murali Ramanathan, and Aidong Zhang. Bridging centrality: graph mining from element level to group level. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 08*, page 336, New York, USA, Aug 2008. ACM Press.

[HNR68] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, Jul 1968.

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[HOSC$^+$12] Enrique Hernández-Orallo, Manuel D. Serrat Olmos, Juan-Carlos Cano, Carlos T. Calafate, and Pietro Manzoni. Evaluation of collaborative selfish node detection in MANETS and DTNs. In *Proceedings of the 15th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems - MSWiM '12*, page 159, New York, USA, 2012. ACM Press.

[HPS10] Sariel Har-Peled and Micha Sharir. Relative (p,$\epsilon$)-Approximations in Geometry. *Discrete & Computational Geometry*, 45(3):462–496, February 2010.

[HXL$^+$09] Pan Hui, Kuang Xu, Victor OK Li, Jon Crowcroft, Vito Latora, and Pietro Lio. Selfishness, altruism and message spreading in mobile social networks. In *IEEE INFOCOM Workshops 2009*, pages 1–6. IEEE, 2009.

[IEE12] IEEE. IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pages 1–2793, Mar 2012.

[INC09] Mouhamad Ibrahim, Philippe Nain, and Iacopo Carreras. Analysis of relay protocols for throwbox-equipped DTNs. In *2009 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pages 1–9, Jun 2009.

[Jel11] Márk Jelasity. Gossip. In Giovanna Di Marzo Serugendo, Marie-Pierre Gleizes, and Anthony Karageorgos, editors, *Self-organising Software*, Natural Computing Series, pages 139–162. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[JFP04] Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a Delay Tolerant Network. *ACM SIGCOMM Computer Communication Review*, 34(4):145–158, August 2004.

[JHB03] Markus Jakobsson, Jean-Pierre Hubaux, and Levente Buttyán. A micro-payment scheme encouraging collaboration in multi-hop cellular networks. In *Financial Cryptography*, pages 15–33, 2003.

[JHM07] David Johnson, Yin-chun Hu, and David Maltz. The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4. Technical report, RFC 4728, 2007.

[JHMB11] Jingwei Miao, O. Hasan, S.B. Mokhtar, and L. Brunie. An adaptive routing algorithm for mobile delay tolerant networks. In *2011 14th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pages 1–5, 2011.

[JMT16] Qingfeng Jiang, Chaoguang Men, and Zeyu Tian. A Credit-Based Congestion-Aware Incentive Scheme for DTNs. *Information*, 7(4), 2016.

[Jos99] Audun Josang. Trust-based decision making for electronic transactions. In *Proceedings of the Fourth Nordic Workshop on Secure Computer Systems (NORDSEC'99)*, pages 496–502, 1999.

[JQ09] Audun Jøsang and Walter Quattrociocchi. Advanced Features in Bayesian Reputation Systems. In Simone Fischer-Hübner, Costas Lambrinoudakis, and Günther Pernul, editors, *Trust, Privacy and Security in Digital Business*, volume 5695 of *Lecture Notes in Computer Science*, pages 105–114. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[JS07] Juan Jose Jaramillo and R. Srikant. Darwin. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking - MobiCom '07*, page 87, New York, USA,, 2007. Mobile Computing and Networking - MobiCom, ACM Press.

[KAF12] Maurice J Khabbaz, Chadi M Assi, and Wissam F Fawaz. Disruption-Tolerant Networking: A Comprehensive Survey on Recent Developments and Persisting Challenges. *IEEE Communications Surveys & Tutorials*, 14(2):607–640, January 2012.

[Kal60] Rudolph Kalman. A New Approach to Linear Filtering and Prediction Problems. *Basic Engineering*, 82(1):35, Mar 1960.

[KAS$^+$12] Nicolas Kourtellis, Tharaka Alahakoon, Ramanuja Simha, Adriana Iamnitchi, and Rahul Tripathi. Identifying high betweenness centrality nodes in large social networks. *Social Network Analysis and Mining*, 3(4):899–914, July 2012.

[KFV11] Raghavendra V Kulkarni, Anna Forster, and Ganesh Kumar Venayagamoorthy. Computational Intelligence in Wireless Sensor Networks: A Survey. *IEEE Communications Surveys & Tutorials*, 13(1):68–96, 2011.

[KHAY07] David Kotz, Tristan Henderson, Ilya Abyzov, and Jihwang Yeo. CRAWDAD data set dartmouth/campus (v. 2007-02-08). Downloaded from `http://crawdad.org/dartmouth/campus/`, February 2007.

[KHYJ13] Chan-Myung Kim, Youn-Hee Han, Joo-Sang Youn, and Young-Sik Jeong. Betweenness of Expanded Ego Networks in Sociality-Aware Delay Tolerant Networks. In Youn-Hee Han, Doo-Soon Park, Weijia Jia, and Sang-Soo Yeo, editors, *Ubiquitous Information Technologies and Applications*, volume 214 of *Lecture Notes in Electrical Engineering*, pages 499–505. Springer Netherlands, Dordrecht, 2013.

[KHYJ14] Chan-Myung Kim, Youn-Hee Han, Joo-Sang Youn, and Young-Sik Jeong. A Socially Aware Routing Based on Local Contact Information in Delay-Tolerant Networks. *The Scientific World Journal*, 2014(408676), 2014.

[KK89] David Kraines and Vivian Kraines. Pavlov and the prisoner's dilemma. *Theory and decision*, 26(1):47–79, 1989.

[KLKH14] Huseyin Kusetogullari, Mark S. Leeson, Burak Kole, and Evor L. Hines. Meta-heuristic algorithms for optimized network flow wavelet-based image coding. *Applied Soft Computing*, 14, Part C:536 – 553, 2014.

[KLY08] Sehoon Kim, Jinkyu Lee, and Ikjun Yeom. Neighbor-Aware Adaptive Retry Limit for IEEE 802.11-Based Mobile Ad Hoc Networks. In *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, pages 1402–1407, Mar 2008.

[KOK09] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. The ONE simulator for DTN protocol evaluation. In *Proceedings of the 2nd international conference on simulation tools and techniques*, page 55. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

[KP07] Ketan Kotecha and Sonal Popat. Multi objective genetic algorithm based adaptive QoS routing in MANET. In *IEEE Congress on Evolutionary Computation*, pages 1423–1428, Sept 2007.

[KPVO11] Ari Keränen, Mikko Pitkänen, Mikko Vuori, and Jörg Ott. Effect of non-cooperative nodes in mobile DTNs. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–7. IEEE, 2011.

[KR12] James Kurose and Keith Ross. *Computer Networking - A top-down approach.* Pearson, 2012.

[LD13] Na Li and Sajal K. Das. A trust-based framework for data forwarding in opportunistic networks. *Ad Hoc Networks*, 11(4):1497–1509, Jun 2013.

[LDS03] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 7(3):19–20, 2003.

[LDS07] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. Technical report, RFC 6639, 2007.

[LHK$^+$08] Kyunghan Lee, Seongik Hong, Seong Joon Kim, Injong Rhee, and Song Chong. Demystifying levy walk patterns in human walks. Technical report, CSC, NCSU, Tech. Rep., 2008.

[LHT$^+$10] Xiaofeng Lu, Pan Hui, Don Towsley, Juahua Pu, and Zhang Xiong. Anti-localization anonymous routing for Delay Tolerant Network. *Computer Networks*, 54(11):1899–1910, 2010.

[LLS00] Yi Li, PM Philip M. Long, and Aravind Srinivasan. Improved bounds on the sample complexity of learning. In *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, pages 309–318. Society for Industrial and Applied Mathematics, February 2000.

[LPP$^+$07] Suk-Bok Lee, Gabriel Pan, Joon-Sang Park, Mario Gerla, and Songwu Lu. Secure incentives for commercial ad dissemination in vehicular networks. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing - MobiHoc '07*, page 150, New York, USA,, 2007. ACM Press.

[LSHG16] Tiezhu Li, Lijuan Sun, Chong Han, and Jian Guo. Packet-Forwarding Algorithm in DTN Based on the Pheromone of Destination Node. *Information*, 7(3), 2016.

[Lux07] Ulrike Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, August 2007.

[LVW09] Zhengyi Le, Gauri Vakde, and Matthew Wright. PEON: privacy-enhanced opportunistic networks with applications in assistive environments. In *Proceedings of the 2nd International Conference on PErvasive Technologies Related to Assistive Environments*, page 76. ACM, 2009.

[LW07] Feng Li and Jie Wu. Mobility Reduces Uncertainty in MANETs. In *IEEE INFOCOM 2007*, pages 1946–1954. IEEE, 2007.

[LW09] Feng Li and Jie Wu. LocalCom: A Community-based Epidemic Forwarding Scheme in Disruption-tolerant Networks. In *2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 1–9. IEEE, June 2009.

[LZQ16] Li Li, Xiaoxiong Zhong, and Yang Qin. A secure routing based on social trust in opportunistic networks. In *2016 IEEE International Conference on Communication Systems (ICCS)*, pages 1–6, Dec 2016.

[Mar02] Peter V Marsden. Egocentric and sociocentric measures of network centrality. *Social networks*, 24(4):407–422, 2002.

[MBPC17] Naércio Magaia, Carlos Borrego, Paulo Rogério Pereira, and Miguel Pupo Correia. PRIVO: A PRIvacy-preserVing Opportunistic routing protocol for Delay Tolerant Networks. In *IFIP Networking*, Jun 2017.

[MC78] James L. Melsa and Davíd L. Cohn. *Decision and estimation theory*. McGraw-Hill, 1978.

[MD01] Mahesh K Marina and Samir R Das. On-demand multipath distance vector routing in ad hoc networks. In *Network Protocols, 2001. Ninth International Conference on*, pages 14–23, Nov 2001.

[MFPC15] Naércio Magaia, Alexandre P. Francisco, Paulo Pereira, and Miguel Correia. Betweenness centrality in delay tolerant networks: A survey. *Ad Hoc Networks*, 33:284 – 305, 2015.

[MGLB00] Sergio Marti, Thomas J Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 255–265. ACM, 2000.

[MGPed] Naércio Magaia, Pedro Dinis Gomes, and Paulo Rogério Pereira. FinalComm: Leveraging dynamic communities to improve forwarding in DTNs. In *an international conference*, submitted.

[MHN$^+$15] Naércio Magaia, Nuno Horta, Rui Neves, Paulo Rogério Pereira, and Miguel Correia. A multi-objective routing algorithm for wireless multimedia sensor networks. *Applied Soft Computing*, 30:104 – 112, 2015.

[MJS06] Fabio Milan, Juan José Jaramillo, and R. Srikant. Achieving cooperation in multihop wireless networks of selfish nodes. In *Proceeding from the 2006 workshop on Game theory for communications and networks - GameNets '06*, page 3, New York, USA,, 2006. ACM.

[MM02] Pietro Michiardi and Refik Molva. Core: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks. In *Advanced Communications and Multimedia Security*, volume 100 of *IFIP — The International Federation for Information Processing*, pages 107–121. Springer US, 2002.

[MM09] Mirco Musolesi and Cecilia Mascolo. CAR: Context-Aware Adaptive Routing for Delay-Tolerant Mobile Networks. *IEEE Transactions on Mobile Computing*, 8(2):246–260, Feb 2009.

[MMDA10] Abderrahmen Mtibaa, Martin May, Christophe Diot, and Mostafa Ammar. PeopleRank: Social Opportunistic Forwarding. In *IEEE INFOCOM 2010*, pages 1–5, Mar 2010.

[MMS12] Waldir Moreira, Paulo Mendes, and Susana Sargento. Opportunistic routing based on daily routines. In *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2012 - Digital Proceedings*, pages 1–6. IEEE, Jun 2012.

[Moh05] Prasant Mohapatra. *AD HOC NETWORKS: Technologies and Protocols*. Springer US, 1st edition, 2005.

[MOV10] Alfred Menezes, Paul Van Oorschot, and Scott Vanstone. *Handbook of applied cryptography*. CRC press, 2010.

[MPC13a] Naércio Magaia, Paulo Rogério Pereira, and Miguel P Correia. Nodes' misbehavior in Vehicular Delay-Tolerant Networks. In *Conference on Future Internet Communications (CFIC)*, pages 1–9, May 2013.

[MPC13b] Naércio Magaia, Paulo Rogério Pereira, and Miguel P Correia. Selfish and malicious behavior in Delay-Tolerant Networks. In *Future Network and Mobile Summit (FutureNetworkSummit)*, pages 1–10. IEEE, 2013.

[MPC15] Naércio Magaia, Paulo Rogério Pereira, and Miguel P Correia. *Cyber Physical Systems: From Theory to Practice*, chapter Security in Delay-Tolerant Mobile Cyber-Physical Applications. CRC Press, 2015.

[MPC17] Naércio Magaia, Paulo Rogério Pereira, and Miguel Pupo Correia. REPSYS: A Robust and Distributed Reputation System for Delay-Tolerant Networks. In *20th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (accepted)*, Jun 2017.

[MPG11] Naércio Magaia, Paulo Rogério Pereira, and António Grilo. A Multipath Extension to the Dynamic Source Routing Protocol for Wireless Multimedia Sensor Networks. In *Proc. 11a Conferência sobre Redes de Computadores, Coimbra, Portugal*, pages 49–56, 2011.

[MPG15] Naércio Magaia, Paulo Rogério Pereira, and António Grilo. High Throughput Low Coupling Multipath Routing for Wireless Multimedia Sensor Networks. *Ad Hoc & Sensor Wireless Networks*, 25(3-4):165–198, 2015.

[MPS15] Sudip Misra, Sujata Pal, and Barun Kumar Saha. Distributed Information-Based Cooperative Strategy Adaptationin Opportunistic Mobile Networks. *IEEE Transactions on Parallel and Distributed Systems*, 26(3):724–737, Mar 2015.

[MRD16] R. Murugeswari, S. Radhakrishnan, and D. Devaraj. A multi-objective evolutionary algorithm based qos routing in wireless mesh networks. *Applied Soft Computing*, 40:517 – 525, 2016.

[MRV$^+$10] Engin Masazade, Ramesh Rajagopalan, Pramod K Varshney, Chilukuri K Mohan, Gullu Kiziltas Sendur, and Mehmet Keskinoz. A Multiobjective Optimization Approach to Obtain Decision Thresholds for Distributed Detection in Wireless Sensor Networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 40(2):444–457, Apr 2010.

[MS12] Alessandro Mei and Julinda Stefa. Give2Get: Forwarding in Social Mobile Wireless Networks of Selfish Individuals. *IEEE Transactions on Dependable and Secure Computing*, 9(4):569–582, Jul 2012.

[MSLC01] Miller Mcpherson, Lynn Smith-Lovin, and James M Cook. Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology*, 27(1):415–444, 2001.

[Mur12] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[New01] Mark EJ Newman. Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. *Physical Review E*, 64(1):016132, June 2001.

[New04] Mark EJ Newman. Analysis of weighted networks. *Physical Review E*, 70(5):056131, 2004.

[New05] Mark EJ Newman. A measure of betweenness centrality based on random walks. *Social Networks*, 27(1):39–54, January 2005.

[NHJK05] Kitae Nahm, Ahmed Helmy, and C.-C. Jay Kuo. TCP over Multihop 802.11 Networks: Issues and Performance Enhancement. In *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '05, pages 277–287, New York, NY, USA, 2005. ACM.

[NLSD13] Jianwei Niu, Yazhi Liu, Lei Shu, and Bin Dai. A P2P query algorithm based on Betweenness Centrality Forwarding in opportunistic networks. In *2013 IEEE International Conference on Communications (ICC)*, pages 3433–3438. IEEE, June 2013.

[NLX$^+$16] Zhaolong Ning, Li Liu, Feng Xia, Behrouz Jedari, Ivan Lee, and Weishan Zhang. CAIS: A Copy Adjustable Incentive Scheme in Community-based Socially-Aware Networking. *IEEE Transactions on Vehicular Technology (accepted)*, 2016.

[OAS10] Tore Opsahl, Filip Agneessens, and John Skvoretz. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, 32(3):245–251, July 2010.

[Oka05] Samir Okasha. Altruism, group selection and correlated interaction. *The British journal for the philosophy of science*, 56(4):703–725, 2005.

[OR94] Martin J Osborne and Ariel Rubinstein. *A course in game theory*. MIT press, 1994.

[Pai99] Pascal Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer Berlin Heidelberg, 1999.

[Pat10] Al-Sakib Khan Pathan. *Security of self-organizing networks: MANET, WSN, WMN, VANET*. Auerbach Publications, 2010.

[PBRD03] Charles Perkins, Elizabeth Belding-Royer, and Samir Das. Ad hoc on-demand distance vector (AODV) routing. RFC 3561, Internet RFCs, 2003.

[PDFV05] Gergely Palla, Imre Derenyi, Illes Farkas, and Tamas Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[PH12] Iain Parris and Tristan Henderson. Privacy-enhanced social-network routing. *Computer Communications*, 35(1):62–74, 2012.

[PVS07] Antonis Panagakis, Athanasios Vaios, and Ioannis Stavrakakis. On the effects of cooperation in DTNs. In *2nd International Conference on Communication Systems Software and Middleware*, pages 1–6. IEEE, 2007.

[RAD78] Ronald L Rivest, Len Adleman, and Michael L Dertouzos. On data banks and privacy homomorphims. *Foundations of Secure Computation*, 1978.

[RC15] Vitor G. Rolla and Marilia Curado. Enabling wireless cooperation in delay tolerant networks. *Information Sciences*, 290:120 – 133, 2015.

[REF14] Alma As-Aad Mohammad Rahat, Richard M. Everson, and Jonathan E. Fieldsend. Multi-objective Routing Optimisation for Battery-powered Wireless Sensor Mesh Networks. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*, GECCO '14, pages 1175–1182, New York, NY, USA, 2014. ACM.

[Res06] Steve Ressler. Social network analysis as an approach to combat terrorism: past, present, and future research. *Homeland Security Affairs*, 2(2):1–10, 2006.

[RK14] Matteo Riondato and Evgenios M. Kornaropoulos. Fast approximation of betweenness centrality through sampling. *Proceedings of the 7th ACM International Conference on Web Search and Data Mining - WSDM '14*, pages 413–422, 2014.

[RMP17] Miguel Rodrigues, Naércio Magaia, and Paulo Pereira. LBHD: Drop Policy for Routing Protocols with Delivery Probability Estimation. In *12th Iberian Conference on Information Systems and Technologies (accepted)*, Jun 2017.

[RMPed] Miguel Rodrigues, Naércio Magaia, and Paulo Pereira. Drop Policies for DTN Routing Protocols with Delivery Probability Estimation. In *an international journal*, submitted.

[RV12] Milena Radenkovic and Ivan Vaghi. Adaptive user anonymity for mobile opportunistic networks. In *7th ACM International Workshop on Challenged Networks, CHANTS 2012*, pages 79–81, New York, USA, 2012. ACM Press.

[SBS02] Eugene Shih, Paramvir Bahl, and Michael J. Sinclair. Wake on Wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices. In *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, MobiCom '02, pages 160–171, New York, NY, USA, 2002. ACM.

[SCRB16] Adrián Sánchez-Carmona, Sergi Robles, and Carlos Borrego. PrivHab+: A secure geographic routing protocol for DTN. *Computer Communications*, 78:56–73, 2016.

[Sen10] Jaydip Sen. A survey on reputation and trust-based systems for wireless communication networks. *arXiv preprint arXiv:1012.2529*, 2010.

[SF07] Gyorgy Szabó and Gabor Fath. Evolutionary games on graphs. *Physics Reports*, 446(4):97–216, 2007.

[Sha76] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[SKLA12] Tossaporn Srisooksai, Kamol Keamarungsi, Poonlap Lamsrichan, and Kiyomichi Araki. Practical data compression in wireless sensor networks: A survey. *Journal of Network and Computer Applications*, 35(1):37 – 59, 2012.

[SKW12] Umair Sadiq, Mohan Kumar, and Matthew Wright. CRISP: Collusion-resistant Incentive-compatible Routing and Forwarding in Opportunistic Networks. In *Proceedings of the 15th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '12, pages 69–78, New York, NY, USA, 2012. ACM.

[SNCR03] Vikram Srinivasan, Pavan Nuggehalli, Carla-Fabiana Chiasserini, and Ramesh R Rao. Cooperation in wireless ad hoc networks. In *IEEE INFOCOM 2003*, volume 2, pages 808–817. IEEE, 2003.

[SNR$^+$07] Lorenzo Strigini, Nuno Neves, Michel Raynal, Michael Harrison, Mohamed Kaaniche, Friedrich Von Henke, Deliverable D, Technische Universität Darmstadt, Deep Blue Srl, and Universität Ulm. Resilience-Building Technologies: State of Knowledge, 2007.

[SOM10] Abdullatif Shikfa, Melek Onen, and Refik Molva. Privacy and confidentiality in context-based and epidemic forwarding. *Computer Communications*, 33(13):1493–1504, 2010.

[SPR04] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S Raghavendra. Single-copy routing in intermittently connected mobile networks. In *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pages 235–244. IEEE, 2004.

[SPR05] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252–259. ACM, 2005.

[SRJB03a] Rahul Shah, Sumit Roy, Sushant Jain, and Waylon Brunette. Data MULEs: modeling a three-tier architecture for sparse sensor networks. In *IEEE International Workshop on Sensor Network Protocols and Applications*, pages 30–41, May 2003.

[SRJB03b] Rahul C. Shah, Sumit Roy, Sushant Jain, and Waylon Brunette. Data MULEs: modeling and analysis of a three-tier architecture for sparse sensor networks. *Ad Hoc Networks*, 1(2-3):215–233, September 2003.

[SRT$^+$10] Thrasyvoulos Spyropoulos, Rao Naveed Bin Rais, Thierry Turletti, Katia Obraczka, and Athanasios Vasilakos. Routing for disruption tolerant networks: taxonomy and design. *Wireless Networks*, 16(8):2349–2370, September 2010.

[Sta07] William Stallings. *Network security essentials: applications and standards*. Pearson Education India, 2007.

[Ste01] Thomas A. Stewart. Six degrees of Mohamed Atta, 2001. http://faculty.cbpp.uaa.alaska.edu/afgjp/PADM610/Six Degrees of Mohamed Atta.pdf.

[Ste10] Angus Stevenson. *Oxford dictionary of English*. Oxford University Press, 2010.

[SZ89] Karen Stephenson and Marvin Zelen. Rethinking centrality: Methods and examples. *Social Networks*, 11(1):1–37, March 1989.

[SZ08] Upendra Shevade and Yin Zhang. Incentive-aware routing in DTNs. In *IEEE International Conference on Network Protocols*, pages 238–247. IEEE, 2008.

[TM67] Jeffrey Travers and Stanley Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.

[TM06] Jack Tsai and Tim Moors. A review of multipath routing protocols: From wireless ad hoc to mesh networks. In *ACoRN early career researcher workshop on wireless multihop networking*, volume 30. Citeseer, 2006.

[TMM$^+$10] John Tang, Mirco Musolesi, Cecilia Mascolo, Vito Latora, and Vincenzo Nicosia. Analysing Information Flows and Key Mediators through Temporal Centrality Metrics Categories and Subject Descriptors. In *Proceedings of the 3rd Workshop on Social Network Systems - SNS '10*, pages 1–6, New York, USA, April 2010. ACM Press.

[TvS12] Theodore L Turocy and Bernhard von Stengel. *Game Theory. Encyclopedia of Information System*. Academic Press, 2012.

[VB00] Amin Vahdat and David Becker. Epidemic routing for partially connected ad hoc networks. Technical report, Technical Report CS-200006, Duke University, 2000.

[VMDV11] Ana Cristina B. Kochem Vendramin, Anelise Munaretto, Myriam Regattieri Delgado, and Aline Carneiro Viana. A Greedy Ant Colony Optimization for routing in delay tolerant networks. In *2011 IEEE GLOBECOM Workshops*, pages 1127–1132. IEEE, December 2011.

[VMDV12] Ana Cristina Barreiras Kochem Vendramin, Anelise Munaretto, Myriam Regattieri de Biase da Silva Delgado, and Aline Carneiro Viana. CGrAnt: a swarm intelligence-based routing protocol for delay tolerant networks. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference - GECCO '12*, page 33, New York, USA, July 2012. ACM Press.

[VYL14] Nikolaos Vastardis, Kun Yang, and Supeng Leng. Socially-aware multi-phase opportunistic routing for distributed mobile social networks. *Wireless Personal Communications*, 79(2):1343–1368, 2014.

[WCZ11] Lifei Wei, Zhenfu Cao, and Haojin Zhu. MobiGame: A user-centric reputation based incentive protocol for delay/disruption tolerant networks. In *2011 IEEE Global Telecommunications Conference (GLOBECOM 2011)*, 2011.

[WCZ$^+$13] Fan Wu, Tingting Chen, Sheng Zhong, Chunming Qiao, and Guihai Chen. A Game-Theoretic Approach to Stimulate Cooperation for Probabilistic Routing in Opportunistic Networks. *IEEE Transactions on Wireless Communications*, 12(4):1573–1583, Apr 2013.

[WEWL06] Weizhao Wang, Stephan Eidenbenz, Yu Wang, and Xiang-Yang Li. OURS: optimal unicast routing systems in non-cooperative wireless networks. In *Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 402–413. ACM, 2006.

[WH01] Kui Wu and Janelle Harms. On-demand multipath routing for mobile ad hoc networks. In *Proceedings of EPMCC*, pages 1–7. Citeseer, 2001.

[Win60] Peter R Winters. Forecasting sales by exponentially weighted moving averages. *Management Science*, 6(3):324–342, 1960.

[WLW04] WeiZhao Wang, Xiang-Yang Li, and Yu Wang. Truthful multicast routing in selfish wireless networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 245–259. ACM, 2004.

[WLX14] Kaimin Wei, Xiao Liang, and Ke Xu. A survey of social-aware routing protocols in delay tolerant networks: applications, taxonomy and design-related issues. *IEEE Communications Surveys & Tutorials*, 16(1):556–578, January 2014.

[WXH13] Jie Wu, Mingjun Xiao, and Liusheng Huang. Homing spread: Community home-based multi-copy routing in mobile social networks. In *IEEE INFOCOM 2013*, pages 2319–2327, Apr 2013.

[WYLC09] Xintao Wu, Xiaowei Ying, Kun Liu, and Lei Chen. A Survey of Algorithms for Privacy-Preservation of Graphs and Social Networks. *Managing and Mining Graph Data*, page 37, 2009.

[WYX$^+$10] Eric Ke Wang, Yunming Ye, Xiaofei Xu, SM Yiu, LCK Hui, and KP Chow. Security issues and challenges for cyber physical system. In *Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, pages 733–738. IEEE Computer Society, 2010.

[WZ04] Wei Wei and Avideh Zakhor. Robust multipath source routing protocol (RMPSR) for video communication over wireless ad hoc networks. In *2004 IEEE International Conference on Multimedia and Expo*, pages 1379–1382, 2004.

[WZ10] Zijian Wang and Jun Zhang. Interference aware multipath routing protocol for wireless sensor networks. In *2010 IEEE GLOBECOM Workshops (GC Wkshps)*, pages 1696–1700, Dec 2010.

[WZCS13] Lifei Wei, Haojin Zhu, Zhenfu Cao, and Xuemin Shen. SUCCESS: A secure user-centric and social-aware reputation based incentive scheme for DTNs. *Ad-Hoc and Sensor Wireless Networks*, 19(1-2):95–118, 2013.

[WZGX14] Kaimin Wei, Deze Zeng, Song Guo, and Ke Xu. On Social Delay-Tolerant Networking: Aggregation, Tie Detection, and Routing. *IEEE Transactions on Parallel and Distributed Systems*, 25(6):1563–1573, Jun 2014.

[XSG03] Feng Xue, Arthur C. Sanderson, and Robert J. Graves. Pareto-based multi-objective differential evolution. In *2003 Congress on Evolutionary Computation, CEC 2003 - Proceedings*, volume 2, pages 862–869. IEEE, 2003.

[XSG06] Feng Xue, A. Sanderson, and R. Graves. Multi-Objective Routing in Wireless Sensor Networks with a Differential Evolution Algorithm. In *Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control, 2006*, pages 880–885, 2006.

[XSG16] Qichao Xu, Zhou Su, and Song Guo. A Game Theoretical Incentive Scheme for Relay Selection Services in Mobile Social Networks. *IEEE Transactions on Vehicular Technology*, 65(8):6692–6702, Aug 2016.

[XWH14] Mingjun Xiao, Jie Wu, and Liusheng Huang. Community-Aware Opportunistic Routing in Mobile Social Networks. *IEEE Transactions on Computers*, 63(7):1682–1695, Jul 2014.

[XZD$^{+}$16] Fang Xu, Huyin Zhang, Min Deng, Ning Xu, and Zhiyong Wang. Social-aware data forwarding in smartphone-based delay-tolerant networks. In *2016 IEEE International Conference on Computational Electromagnetics (ICCEM)*, pages 84–86, Feb 2016.

[YCH12] Halil Yetgin, Kent Tsz Kan Cheung, and Lajos Hanzo. Multi-objective routing optimization using evolutionary algorithms. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 3030–3034, Apr 2012.

[YL01] Xin Yuan and Xingming Liu. Heuristic algorithms for multi-constrained quality of service routing. In *IEEE INFOCOM 2001*, volume 2, pages 844–853 vol.2, 2001.

[YMF15] Peiyan Yuan, Huadong Ma, and Huiyuan Fu. Hotspot-entropy based data forwarding in opportunistic social networks. *Pervasive and Mobile Computing*, 16, Part A:136 – 154, 2015.

[YMH$^{+}$16] Lin Yao, Yanmao Man, Zhong Huang, Jing Deng, and Xin Wang. Secure Routing Based on Social Similarity in Opportunistic Networks. *IEEE Transactions on Wireless Communications*, 15(1):594–605, Jan 2016.

[ZAZ04] Wenrui Zhao, Mostafa Ammar, and Ellen Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing - MobiHoc '04*, page 187, New York, USA, May 2004. ACM Press.

[ZC12] Jun Zhou and Zhenfu Cao. TIS: A threshold incentive scheme for secure and reliable data forwarding in vehicular Delay Tolerant Networks. In *2012 IEEE Global Communications Conference (GLOBECOM 2012)*, pages 985–990, Dec 2012.

[ZDG$^+$14] Haojin Zhu, Suguo Du, Zhaoyu Gao, Mianxiong Dong, and Zhenfu Cao. A probabilistic misbehavior detection scheme toward efficient trust establishment in delay-tolerant networks. *IEEE Transactions on Parallel and Distributed Systems*, 25(1):22–32, Jan 2014.

[ZEPP13] Niels Zeilemaker, Zekeriya Erkin, Paolo Palmieri, and Johan Pouwelse. Building a privacy-preserving semantic overlay for peer-to-peer networks. In *Information Forensics and Security (WIFS), 2013 IEEE International Workshop on*, pages 79–84, Nov 2013.

[ZGH07] Ge Zhong, Ian Goldberg, and Urs Hengartner. Louis, Lester and Pierre: Three Protocols for Location Privacy. *Pets'07*, pages 62–76, 2007.

[ZLG$^+$11] Xuejun Zhuo, Qinghua Li, Wei Gao, Guohong Cao, and Yiqi Dai. Contact duration aware data replication in Delay Tolerant Networks. In *2011 19th IEEE International Conference on Network Protocols*, pages 236–245. IEEE, October 2011.

[ZLL$^+$09] Haojin Zhu, Xiaodong Lin, Rongxing Lu, Yanfei Fan, and Xuemin Shen. Smart: A secure multilayer credit-based incentive scheme for delay-tolerant networks. *IEEE Transactions on Vehicular Technology*, 58(8):4628–4639, 2009.

[ZLLY07] Sheng Zhong, Li Erran Li, Yanbin Grace Liu, and Yang Richard Yang. On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks. *Wireless Networks*, 13(6):799–816, 2007.

[ZLT01] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm. TIK-Report 103, Swiss Federal Institute of Technolog, ETH Zentrum, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 2001.

[ZWZ$^+$15] Huan Zhou, Jie Wu, Hongyang Zhao, Shaojie Tang, Canfeng Chen, and Jiming Chen. Incentive-Driven and Freshness-Aware Content Dissemination in Selfish Opportunistic Mobile Networks. *IEEE Transactions on Parallel and Distributed Systems*, 26(9):2493–2505, Sept 2015.

[ZXSW13] Ying Zhu, Bin Xu, Xinghua Shi, and Yu Wang. A survey of social-based routing in delay tolerant networks: positive and negative social effects. *IEEE Communications Surveys & Tutorials*, 15(1):387–401, 2013.