Towards Scalable Routing for Wireless Multi-hop Networks

Encaminhamento Escalável em Redes Multi-hop Sem-fios

David Fonseca Palma
Department of Informatics Engineering
Doctoral Program in Information Science and Technology
University of Coimbra

A thesis submitted for the degree of
Doctor of Philosophy

25th September 2012

Towards Scalable Routing for Wireless Multi-hop Networks

David Fonseca Palma
Department of Informatics Engineering
Doctoral Program in Information Science and Technology
University of Coimbra

Advised by Prof. Dr. Marilia Curado

A thesis submitted for the degree of
Doctor of Philosophy

25th September 2012

Dedicated to no one special & everyone in particular.

Acknowledgements

Having reached the final stage of this work, credit must be given to those who supported me throughout the entire process, undoubtedly leaving their mark. To all of them I hereby leave a token of my deepest gratitude.

First and foremost to my advisor, for introducing me to the Laboratory of Communications and Telematics and for believing in me from the start. For all the meetings, discussions, revisions and perennial encouragement.

To several members of the community of the Department of Informatics Engineering of the University of Coimbra, responsible for ensuring a great research environment for the past years. In particular to current and past members of CISUC, and all the LCT investigators with whom I had the pleasure to interact with, for their companionship and insightful debates.

To the Portuguese National Foundation for Science and Technology for the PhD Scholarship (SFRH/BD/43790/2008), to the IST 6th Framework Programme Integrated Project WEIRD (IST-034622), which is partially funded by the Commission of the European Union, to the User Centric Routing (UCR) Project (PTDC/EEATEL/103637/2008) and to the National Project MORFEU (PTDC/EEA-CRO/108348/2008).

To the OPNET University Program for providing the required licenses for the OPNET's Wireless Modeler Suite and its accurate simulation models, as well as for the taught training sessions.

For those who have been with me since the very beginning, for their attention, support, advice, energy and everything else. To my entire Family, but in particular to my Mother and Father, and to the Palma Nascimento family for their unconditional support and outrageous boost of my happiness (TD and JP).

Kindred in blood and spirit, both sibling and ascendant, to my personal Athena in this Odysseia of life, a fully open hearted thank you!

Within the cosy embrace of the G.6.6, where many hours have passed by, strong friendship bonds have also been established. We have all laughed, argued, cursed and shot each other but, above all, we have been really good friends. Thank you all that have been part of this journey and, in particular, thank you João Gonçalves, Tiago Cruz and Vítor Bernardo, without you the outcome would not have been the same.

Last, but most definitely not least. In all those moments where I was grumpy, harsh, defeated and boring. When I was working at night, weekends and holidays. Every time I needed an "expression check" and revisions. Thank you for being unconditionally by my side, you have played a Stellar role in my life.

Abstract

The growing diffusion of wireless interfaces (namely using the IEEE 802.11 standard) in the most diverse type of equipments has led to a myriad of new networking scenarios. Several wireless capable gadgets are expected to be interconnected, demanding an increasing amount of resources to existing network infrastructures. In order to suppress these networking needs, in any possible scenario, the Ad-hoc paradigm allows the creation of autonomous infrastructure-less networks, capable of heterogeneously guarantee communication between these wireless devices.

Even though previous works already exist regarding routing in Mobile Ad-hoc Networks (MANETs), the increasing demand for these networks revealed that these protocols do not scale accordingly. Bearing this issue in mind, the presented work addresses the scalability of routing protocols, proposing a new routing paradigm capable of handling large-scale networks, benefiting from the contextual proximity between users.

By introducing a well defined network organisation with different granularity levels, the Deferred Routing scheme, presented in this thesis, uses a hierarchical structure to handle existing clusters of nodes, in conjunction with virtual clusters that aggregate the real clusters. This organisation provides nodes with more stable network views, detracting the unwanted effects of mobility between neighbour clusters. The routing stability and scalable mechanisms are also achieved by using a packet forwarding technique in which the routing information is progressively more accurate, as the level of routing information detail increases when nodes are closer to the desired destination. During the forwarding process, nodes in the borders of clusters, or Gateway nodes, are identified in order to cross different clusters. In the gateway selection process a link quality estimation model is used, ensuring that the best existing gateway nodes are selected, implicitly achieving a balanced load between the available gateways. This forwarding approach further improves the performance of the proposed protocol, being

extremely resilient to network changes and enabling self-healing properties of the chosen paths, as they are maintained by the different gateway nodes across clusters.

Through an extensive performance analysis of Deferred Routing, resorting to different evaluation scenarios, an improvement on traffic delivery was registered when compared to well-known routing alternatives. The obtained results also revealed an increased stability regarding topology changes and routing table calculations, leading to a lower control traffic overhead in all the assessed scenarios, outperforming the analysed competitors.

These results motivate a new era of applications for MANETs and suggest that ubiquitous communication is likely to take place in a near future. This results not only from the increasing availability of wireless capable devices, but also due the possibility of managing large-scale networks using lightweight routing mechanisms.

Resumo

A crescente disponibilização de interfaces sem-fios (nomeadamente seguindo o standard IEEE 802.11) nos mais diversos tipos de equipamentos, deu origem a uma grande diversidade de novos cenários de redes de comunicação. Espera-se que vários dispositivos com capacidades de comunicação sem-fios se encontrem interligados, exigindo uma crescente quantidade de recursos às infraestruturas de rede existentes. De modo a dar resposta a estas necessidades de comunicação nos vários cenários possíveis, o paradigma Ad-hoc possibilita a criação de redes sem-infraestrutura autónomas e capazes de garantir de forma heterogénea a comunicação entre estes dispositivos sem-fios.

Apesar de já existirem trabalhos anteriores que dizem respeito ao encaminhamento em redes móveis Ad-hoc, a crescente procura destas redes revelou que os seus protocolos não são escaláveis. Tendo este problema em consideração, o trabalho apresentado aborda a escalabilidade dos protocolos de encaminhamento, propondo um novo paradigma capaz de lidar com redes de larga-escala, que beneficia da proximidade contextual entre utilizadores.

Por meio de uma organização de rede bem definida com diferentes níveis de granularidade, o esquema de Encaminhamento Diferido (*Deferred Routing*) apresentado nesta tese usa uma estrutura hierárquica para lidar com os grupos de nós existentes, utilizando em simultâneo grupos virtuais que agregam os grupos reais. Esta organização fornece aos nós vistas da rede mais estáveis, reduzindo os efeitos indesejados da mobilidade entre grupos vizinhos. A estabilidade do encaminhamento e mecanismos de escalabilidade são também alcançados através da utilização de uma técnica de reencaminhamento de pacotes onde a informação de encaminhamento é progressivamente mais precisa, uma vez que o nível de detalhe da informação de encaminhamento aumenta quando os nós se encontram mais perto do destino desejado. Durante o processo de reencaminhamento, os nós na fronteira do seu grupo, denominados nós fronteira, são identificados por forma a atravessar vários grupos. No processo de selecção de nós fronteira é utilizado um modelo de estimação da qualidade de uma ligação sem-fios, garantindo um equilíbrio da carga entre os nós fronteira existentes. Esta abordagem de reencaminhamento melhora a performance do protocolo proposto, sendo extremamente resiliente a alterações na rede e atribuindo propriedades de auto-reparação dos caminhos escolhidos, uma vez que estes são mantidos por diferentes nós fronteira através dos grupos de nós.

Através de uma extensa análise da performance do protocolo de Encaminhamento Diferido, recorrendo a diferentes cenários de avaliação, foi registada uma melhoria na entrega de tráfego quando comparado com outras alternativas de encaminhamento conhecidas. Os resultados obtidos revelam também uma crescente estabilidade no que diz respeito a alterações de topologia e cálculos da tabela de encaminhamento, resultando numa menor sobrecarga do tráfego de controlo em todos os cenários avaliados, superando os concorrentes analisados.

Estes resultados sugerem que é provável o início de uma nova era de aplicações para redes móveis Ad-hoc e comunicação ubíqua. Isto ocorre não só devido ao aumento do número de equipamentos sem-fios existentes, mas também devido à possibilidade de manutenção de redes de larga-escala utilizando mecanismos de encaminhamento com pouca sobrecarga.

Preface

Under the comradeship and hospitality of the Laboratory of Communications and Telematics, this thesis was conducted at the Centre for Informatics and Systems of the University of Coimbra. Throughout this work contributions were also made to the following research projects:

- IST FP6 WEIRD - WiMAX Extension to Isolated Research Data networks (Integrated Project IST-FP6-034622).
- IST FP6 CONTENT - Network-of-Excellence on Content Networks and Services for Home Users (Integrated Project IST-FP6-0384239).
- National Project MORFEU - Multi-Objective Robot Fleet for improved commUnication (PTDC/EEA-CRO/108348/2008).
- User Centric Routing (UCR) Project (PTDC/EEATEL/103637/2008).

Within the scope of the developed work, the bachelor thesis of a Bulgarian Erasmus student (entitled *"Routing in Mobile Ad-hoc Networks"*), as well as the work of master students from the Department of Informatics Engineering of the University of Coimbra have also been informally co-advised. From this work two master thesis stand out:

- Luís Conceição, "Analysis of Clustering Schemes for Mobile Ad-Hoc Networks", Department of Informatics Engineering of the University of Coimbra, Master Thesis, July 2009.
- Maria Clara Marto, "Avaliação de protocolos de encaminhamento em redes móveis ad-hoc", Department of Informatics Engineering of the University of Coimbra, Master Thesis, July 2012.

The work that has been published during the development of this thesis is presented next:

- David Palma, Marilia Curado, “Analysis of Trends in Routing”, in Proceedings. of the Routing in Next Generation Workshop (RiNG Workshop), Madrid, Spain, 13 to 14 December 2008.
- David Palma, Marilia Curado. “NODRoP, nature optimized deferred routing protocol”. In IEEE International Conference on Computer Communications, INFOCOM Workshops 2009, IEEE, pages 1-2. IEEE, 2009.
- David Palma, Marilia Curado. “Inside-out olsr scalability analysis”. In Proceedings of the 8th International Conference on Ad-Hoc, Mobile and Wireless Networks, ADHOC-NOW ’09, pages 354-359, Berlin, Heidelberg, 2009. Springer-Verlag.
- David Palma, Marilia Curado. “Increasing Reliability in Large Scale Ad-hoc Networks”, in European Research Consortium for Informatics and Mathematics (ERCIM) e-Mobility workshop, May 2010.
- David Palma, Marilia Curado, “DASH, Deferred aggregated routing for scalable ad-hoc networks”, Wireless Days (WD), 2010 IFIP, pp.1-6, 20-22 Oct. 2010, doi: 10.1109/WD.2010.5657744.
- David Palma, Joao Goncalves, Marilia Curado, “Challenging Wireless Networks, an Underground Experience”, 1st International Workshop on Opportunistic Sensing and Processing in Mobile Wireless Sensor and Cellular networks, May 2011, Bilbao, Spain, May 9-10, 2011.
- David Palma, Marilia Curado, “Onto scalable Ad-hoc networks: Deferred Routing”, Journal of Computer Communications, Elsevier, Available online 8 May 2012, ISSN 0140-3664, 10.1016/j.comcom.2012.04.026, Impact Factor: 1.044, 5-Year Impact Factor: 1.067.
- David Palma, Marilia Curado, “Impact Analysis of Hierarchical Transitions in Multi-hop Clustered Networks”, 13th IEEE/IFIP Network Operations and Management Symposium (NOMS 2012), Hawaii, USA, 16-20 April, 2012.
- David Palma, Marilia Curado, “Scalability and Routing Performance of Future Autonomous Networks”, 3rd IEEE International Workshop on SmArt COmmunications in NEtwork Technologies (SaCoNeT-III), Ottawa, Canada, June 11, 2012.
- David Palma, Helder Araujo, Marilia Curado, “Link quality estimation in wireless multi-hop networks using Kernel based methods”, Journal

of Computer Networks, Elsevier, Available online 31 July 2012, ISSN 1389-1286,
10.1016/j.comnet.2012.07.012, Impact Factor: 1.200, 5-Year Impact Factor: 1.589.

- David Palma, Marilia Curado, “Scalable Multi-hop Routing in Wireless Networks”, submitted to the Journal on Wireless Communications and Networking, EURASIP, Impact Factor: 0.870.
- David Palma, Marilia Curado, “Scalability and Routing Performance of Future Autonomous Networks”, submitted to the International Journal of Internet Protocol Technology, Inderscience, EI.

The following works, not necessarily related to Routing in Mobile Ad-hoc Networks, were also co-authored during the production of this thesis:

- Luis Cordeiro, Marilia Curado, Edmundo Monteiro, Vitor Bernardo, David Palma, “Hybrid on-path off-path approach for end-to end signalling across NSIS domains (HyPath) v5”, in 71st IETF, Philadelphia, PA, USA, March 9-14, 2008.
- Bruno Sousa, Pedro Neves, Gabriela Leão, David Palma, Jorge Sá Silva, Susana Sargento, Francisco Fontes, Marilia Curado, Fernando Boavida, “The Cost of Using IEEE 802.16d Dynamic Channel Configuration”, in Proceedings of the IEEE International Conference on Communications (ICC2008), Beijin, China, 19 to 23 May 2008.
- Luís Conceição, David Palma, Marilia Curado, “A novel stable and low maintenance clustering scheme”, in Proceedings ACM Symposium of Applied Computing, SAC, Sierre, Switzerland, 22 - 26 March 2010.
- Maria Clara Marto, David Palma, Marilia Curado, “Encaminhamento em redes móveis ad-hoc: NS-3 versus OPNET”, 11ª Conferência sobre Redes de Computadores, CRC 2011, Coimbra, Portugal, November 2011.

Contents

Contents xiii

List of Figures xvii

List of Tables xix

List of Algorithms xxi

Acronyms xxiii

1 Introduction 1
- 1.1 Background and Motivation . . . 2
- 1.2 Assumptions and Research Questions . . . 4
- 1.3 Approach and Contributions . . . 8
- 1.4 Thesis Structure . . . 10

2 Multi-hop Routing in Wireless Ad-hoc Networks 13
- 2.1 Flat Routing Protocols . . . 14
  - 2.1.1 Proactive Routing Protocols . . . 15
  - 2.1.2 Reactive Routing Protocols . . . 19
  - 2.1.3 Hybrid Routing Protocols . . . 24
- 2.2 Hierarchical Routing Protocols . . . 28
  - 2.2.1 Proactive Routing Protocols . . . 28
  - 2.2.2 Reactive Routing Protocols . . . 32
  - 2.2.3 Hybrid Routing Protocols . . . 34
- 2.3 Open Issues . . . 37
- 2.4 Summary . . . 38

3 Deferred Routing 41
3.1 Concept and Model Definition . . . 42
3.2 Deferred Routing Mechanisms . . . 47
  3.2.1 Reduced Routing Information Overhead . . . 47
  3.2.2 Route Establishment in Deferred Routing . . . 49
  3.2.3 Efficient Gateway Selection . . . 51
3.3 Deferred Routing Examples . . . 51
3.4 Complexity Analysis . . . 54
3.5 Summary . . . 57

4 Gateway Selection in Deferred Routing 59
4.1 Link Quality Estimation . . . 59
4.2 An Accurate Model for Link Quality Estimation . . . 63
4.3 Model Parametrisation . . . 66
  4.3.1 Simulation-based Traces . . . 67
  4.3.2 Reality-based Traces . . . 70
  4.3.3 Additional Results . . . 73
  4.3.4 Model Validation . . . 75
4.4 Summary . . . 78

5 Deferred Routing Performance 81
5.1 Evaluation Metrics . . . 82
5.2 Scalability Assessment . . . 85
  5.2.1 Description . . . 85
  5.2.2 Obtained Results . . . 86
5.3 Resilience to Mobility . . . 93
  5.3.1 Description . . . 93
  5.3.2 Obtained Results . . . 96
5.4 Hierarchy Robustness . . . 102
  5.4.1 Description . . . 103
  5.4.2 Obtained Results . . . 107
5.5 Realistic Scenario . . . 113
  5.5.1 Description . . . 113
  5.5.2 Obtained Results . . . 115
5.6 Summary . . . 117

6 Conclusion and Future Work 119
6.1 Synthesis of the Thesis 119
6.2 Contributions 120
6.3 Future Work 122

References 125

A Deferred Routing Specification 145
A.1 Required Algorithms 146
A.2 Summary 160

List of Figures

1.1 Thesis Structure 11

2.1 Taxonomy of MANET routing protocols 14

3.1 Contexts with embedded clusters 43
3.2 Clusters' Hierarchy 43
3.3 Network View by Cluster 4 44
3.4 Network View by Cluster 4 (updated) 44
3.5 Additional Network Views 45
3.6 Additional Network Views 46
3.7 Forwarding Process – Path Taken 52

4.1 Periodic Routing Message Exchange 64
4.2 The Epanechnikov Kernel Regression Results (simulation) 68
4.3 The Gaussian Kernel Regression Results (simulation) 69
4.4 The Epanechnikov Kernel Regression Results (real) 71
4.5 The Gaussian Kernel Regression Results (real) 72
4.6 The Epanechnikov Kernel Regression Results for 3 Packets 74
4.7 The Gaussian Kernel Regression Results for 3 Packets 75
4.8 Oulu Mobility Trajectories 77
4.9 High Density Scenario - Real-time Performance 79
4.10 Oulu Scenario - Real-time Performance 79

5.1 Increasing Number of Clusters 86
5.2 Average Percentage of Losses 87
5.3 Average End-to-end Delay 88
5.4 Average Number of Hops 90
5.5 Topology Changes per Routing Table Calculation (AToCRT) 91

5.6 Number of Forwards per *TC* message 92
5.7 Sent Routing Traffic Overhead 93
5.8 Received Routing Traffic Overhead 94
5.9 Mobile Models' Trajectories 95
5.10 Average Percentage of Losses 97
5.11 Average End-to-end Delay 97
5.12 Average Number of Hops 99
5.13 Topology Changes per Routing Table Calculation (AToCRT) 100
5.14 Number of Forwards per *TC* message 101
5.15 Sent Routing Traffic Overhead 102
5.16 Received Routing Traffic Overhead 103
5.17 Network Hierarchy 104
5.18 Same-Level Transition Example for 2 Clusters 105
5.19 Same-Level Transition Example for 3 Clusters 105
5.20 One-Level Transition Example for 3 Clusters 106
5.21 Same-Level Transition Example for 4 Clusters 106
5.22 Two-Level Transition Example for 4 Clusters 107
5.23 Average Percentage of Losses 107
5.24 Average End-to-end Delay 108
5.25 Average Number of Hops 110
5.26 Topology Changes per Routing Table Calculation (AToCRT) 110
5.27 Number of Forwards per *TC* message 111
5.28 Sent Routing Traffic Overhead 112
5.29 Received Routing Traffic Overhead 112
5.30 Defined Scenarios Representation 114
5.31 Sent and Received Routing Traffic Overhead 117

A.1 DefeR Procedures in the Network 145

List of Tables

2.1 Comparison of Flat Proactive Routing Protocols 19
2.2 Comparison of Flat Reactive Routing Protocols 24
2.3 Comparison of Flat Hybrid Routing Protocols 28
2.4 Comparison of Hierarchical Proactive Routing Protocols 31
2.5 Comparison of Hierarchical Reactive Routing Protocols 34
2.6 Comparison of Hierarchical Hybrid Routing Protocols 37

4.1 Wireless Link Modelling 62
4.2 Obtained Kernels (simulation) 67
4.3 Obtained Kernels (real) 70
4.4 Obtained Kernels for 3 Packets 73
4.5 Combined Results (50 runs) 78

5.1 Clusters' Description 114
5.2 Flows' Characteristics 114
5.3 Traffic Delivery Perfomance 115
5.4 Routing Performance 116

List of Algorithms

1 Hierarchical Cluster View 47
2 Sent and Received Routing Messages 48
3 Inter-Cluster Routing 50
4 Packet Forwarding Step by Step 53
5 Updated Packet Forwarding 54
6 View Determination Procedure 146
7 Cluster Containment Procedure 147
8 Periodic *HELLO* messages 148
9 Topology Control Messages 149
10 Processing Received Routing Messages 150
11 Inter-Cluster Routing 151
12 Inter-Cluster Routing (continued) 152
13 IP-Context Mapping Procedure 153
14 IP-Context Mapping Procedure (continued) 154
15 Creation of Context Connectivity 155
16 Creation of Gateway Connectivity 156
17 Creation of Gateway Connectivity (continued) 157
18 Creation of Gateway Connectivity (continued) 158
19 Creation of Temporary Context Connectivity 159

Acronyms

ABR Associativity Based Routing.

AODV Ad hoc On-demand Distance Vector.

AODV-BR Ad hoc On-demand Distance Vector Backup Routing.

AS Autonomous System.

ASE Averaged Squared Error.

AToCRT Average Topology Changes per Routing Table calculation.

BGP Border Gateway Protocol.

BRP Bordercast Resolution Protocol.

BSP Broadcast Storm Problem.

CBRP Cluster Based Routing Protocol.

CCIR Consultative Committee on International Radio.

CGSR Cluster-head Gateway Switch Routing.

C-*HELLO* Cluster *HELLO*.

CID Cluster Identifier.

C-OLSR Cluster-based OLSR.

C-*TC* Cluster Topology Control.

DART Dynamic Address Routing.

DASH Deferred Aggregated routing for Scalable ad-Hoc networks.

DDR Distributed Dynamic Routing.

DefeR Deferred Routing.

DHCP Dynamic Host Configuration Protocol.

DHT Distributed Hash Table.

DIFS Distributed InterFrame Space.

DSDV Destination-Sequenced Distance-Vector.

DSR Dynamic Source Routing.

DTN Delay-Tolerant Network.

DYMO Dynamic MANET On-demand.

ERS Expanded Ring Search.

ETT Expected Transmission Time.

FBR Friendship-based Routing.

FoHEAT Follow Heat.

FSR Fisheye State Routing.

FZRP Fisheye Zone Routing Protocol.

GPS Global Positioning System.

GW Gateway.

HARP Hybrid Ad Hoc Routing Protocol.

Hi-AODV Hierarchical AODV.

HMC *HELLO* Message Counter.

HRAN Heat Routing for Ad-hoc Network.

IARP Intra-zone Routing Protocol.

ID Identifier.

IEEE Institute of Electrical and Electronics Engineers.

IERP Inter-zone Routing Protocol.

IETF Internet Engineering Task Force.

IP Internet Protocol.

IPv4 Internet Protocol version 4.

ISE Integrated Squared Error.

IZRP Independent Zone Routing Protocol.

KNN K-Nearest Neighbour.

LCC Least Cluster Change.

LCR Layered Cluster-based Routing.

LLT Link Lifetime.

LOS Line-of-Sight.

LSP Link State Packet.

LTE Long Term Evolution.

MAC Medium Access Control.

MAHR Mobility Aware Hybrid Routing.

MANET Mobile Ad-hoc Network.

MID Multiple Interface Declaration.

MISE Mean Integrated Squared Error.

MMWN Multimedia support in Mobile Wireless Networks.

MPR Multipoint Relay.

NHDP Neighborhood Discovery Protocol.

OLSR Optimized Link State Routing.

OLSRv2 Optimized Link State Routing version 2.

PIFS Point coordination InterFrame Space.

RBF Radial Basis Functions.

RDM Random Direction Mobility.

RERR Route Error.

RoIMP Route Improvement.

RoREP Route Reply.

RoREPAIR Route Repair.

RREP Route Reply.

RREQ Route Request.

RSS Received Signal Strength.

RSSI Received Signal Strength Indication.

RWM Random Walk Mobility.

RWP Random Waypoint.

RwREQ Random Walk Request.

SIFS Short InterFrame Space.

S-LLT Single-Node Link Lifetime.

STAR Source-Tree Adaptive Routing.

SVM State Vector Machines.

TAB Time Aware Bloom Filter.

TBCA Tiered Based Clustering Algorithm.

TC Topology Control.

TLDV Tree-like Distance Vector.

TORA Temporally-Ordered Routing Algorithm.

TZRP Two-Zone Routing Protocol.

UDP User Datagram Protocol.

WCN Wireless Community Network.

Wi-Fi Wireless Fidelity.

WiMAX Worldwide Interoperability for Microwave Access.

WLAN Wireless Local Area Network.

WRP Wireless Routing Protocol.

ZHLS Zone-based Hierarchical Link-State.

ZHLSGF ZHLS Gateway Flooding.

ZRP Zone Routing Protocol.

Chapter 1

Introduction

Computer networking has long evolved since the first packet switching that existed in the early 60's. Not only has the number of connections between users and devices increased, but these connections have also diversified from copper cables, through optical fibre into the wireless medium. In particular, wireless technologies have registered a remarkable evolution in order to cope with the increasing portability of computers and other gadgets such as personal-digital-assistants, media players, cell-phones among others.

The massive dissemination of wireless capable devices is leading to a trend where, in a near future, users are expected to own several hundreds of gadgets requiring wireless connections [Cimmino and Donadio, 2009] amongst themselves and other users. Such demand of intra and inter networking capabilities will compel researchers and network providers to create alternative communication paradigms to the existing ones and deploy suitable infrastructures.

Despite the flexibility provided by new long-range wireless technologies, such as Worldwide Interoperability for Microwave Access (WiMAX) [WiMAX Forum, 2011] and Long Term Evolution (LTE) [Shen et al., 2012], these networks are still expensive and do not scale with ease. For instance, in events where thousands of people are gathered, such as a football game or a concert, these networks are known to fail in delivering a good quality of experience when users try to share their emotions by sending emails, photos and other content. Moreover, in rural areas or disaster scenarios, the coverage provided by these approaches is usually limited or unavailable either by option from the operators or as result of existing damage on the infrastructures.

Bearing in mind the necessity to handle the restrains of existing infrastructures, and the urge to provide alternatives where they are not available, the concept of Ad-hoc networks has been suggested. This has enabled the impromptu creation of wireless multi-hop networks, where each wireless node behaves as router. By using this approach, users are capable of maintaining their own network, being able to locally share their contents without requiring additional infrastructures. User mobility is of course an important requirement and thus Mobile Ad-hoc Networks (MANETs) must be able to handle the creation and destruction of new links between different users, a task usually delivered to a routing protocol.

While MANETs give users the freedom to create networks in the spur of the moment, without any particular restrictions, these networks may also suffer from scalability problems. In fact, the role of routing protocols may become extremely challenging when the number of connected nodes increases. This difficulty results from the non-existence of a well defined organization and from interference phenomena intrinsic to wireless technologies.

In this Chapter existing work on multi-hop wireless networks and their scalability issues will be addressed in Section 1.1, followed by the motivation that drives this thesis. Afterwards, in Section 1.2, the taken assumptions and raised research questions are presented. Finally, the followed approach and given contributions are detailed in Section 1.3, and the document structure is outlined in Section 1.4.

1.1 Background and Motivation

Recent technological advances have promoted a massive dissemination of wireless capable devices with greater processing power, higher memory and autonomy, increasing the connectivity between users and different services and applications. As a result, in a near future each person is expected to be surrounded by hundreds or even thousands of these devices [Cimmino and Donadio, 2009], motivating the development of networks capable of connecting them whilst supporting several applications' requirements, demanding a considerable amount of physical resources from the available infrastructures. The Ad-hoc creation of wireless multi-hop networks to handle these demands may be a solution. However, the management of a large scale infrastructure-less network is still a challenge.

Another typical characteristic of the spreading wireless gadgets is their portability, creating new challenges related with mobility. This aspect is crucial for users who expect seamless connectivity regardless of their movement and action. However, different trajectories may reduce connectivity coverage or, on the other hand, increase the number of connections and consequently the number of packet collisions, resulting in the disruption of paths established by routing protocols.

Conventional routing used in wired and in infrastructure-based wireless networks could not be applied to these spontaneously created wireless networks, due to their dynamics. Distance-vector and link-state routing approaches have been used to establish routes in these networks, using techniques such as Multipoint Relay nodes, in order to optimize the forwarding of topology-related routing packets. However, these proactive routing schemes were not always suitable for networks where high mobility patterns were registered, motivating the creation of reactive on-demand routing alternatives. These protocols strive for typically having a reduced amount of control traffic, finding paths solely when required. Since on-demand routing suffered from an initial delay when retrieving paths and is prone to increased overhead in networks with high number of traffic flows, hybrid routing approaches were developed, trying to join the best of both proactive and reactive routing approaches. Other routing schemes take advantage of knowledge about nodes' positions. This class of routing protocols, geographical routing protocols [Beigh and Peer, 2012], are characterized for having low overhead and memory requirements, however positioning information may not be available or it may be inaccurate in several scenarios, such as indoor scenarios or large and dense urban areas [Piran et al., 2011].

Wireless multi-hop networks have increasingly stood out for being available anywhere, without requiring any existing infra-structures, and for being self-organised, self-administrated and self-maintained. For this purpose, as previously mentioned, several works already exist on this topic. However, maintaining routing performance for large scale networks is a critical issue [Jemili et al., 2011a]. Taking this problem into account, different works propose schemes involving techniques such as dynamic addressing, keeping network nodes organized in a well defined topology; geographic partitioning, in order to easily create stable clusters; and also typical clustering solutions, to simply reduce the total amount of routing traffic.

While some approaches aim at scalable routing using different approaches, they lack a thorough evaluation of the impact of different mobility models. In fact, regarding this aspect, most routing solutions disregard the dynamics of different mobility models, focusing only on one mobility pattern. Nevertheless, in order to appropriately evaluate the efficiency of an Ad-hoc network and the performance of routing protocols, these aspects have to be taken into account. Moreover, other works that study the impact of mobility fail to provide an extensive evaluation with existing mobility models [Prasad et al., 2009].

A different perspective on wireless multi-hop routing has been provided with the definition of Delay-Tolerant Networks (DTNs). In these networks, routing protocols are designed to deliver traffic that is not delay sensitive, despite the sparse intermittently connected properties of such network. Conventional routing in wireless multi-hop networks is not suitable for highly dynamic scenarios, as it needs to establish an end-to-end path before starting the routing of data packets, which may not be possible at a given moment.

Even though most wireless networks are in fact intermittently connected due to interferences in the wireless medium, the mobility of nodes has also an important role in this aspect. Typical DTN solutions such as PRoPHET [Lindgren et al., 2012] are capable of operating with delay tolerant traffic when wireless connections are not reliable, but fails to perform well with completely unknown node mobility. Other approaches focus on more stable parameters, such as social interactions between nodes. For instance, the Friendship-based Routing (FBR) protocol [Bulut and Szymanski, 2010] or the Social Aware Networking (SANE) scheme [Mei et al., 2011] take into account social interactions, both physical and virtual, in order to make a packet forward decision. Nonetheless, these schemes fail to determine a possible path to deliver their packets in real time and are therefore not comparable with the presented routing solution. Moreover, scalability issues are not taken into consideration, rendering these approaches useless in highly populated scenarios where a large amount of data traffic may depend on one node alone.

Motivated by the lack of a routing scheme where large clusters of wireless nodes may exist, a new routing approach that takes into account the increased interaction between users within a same context, regardless of the used mobility pattern, should be proposed. Previous studies show that content is exchanged between millions of individuals resorting to phone interactions or on-line services. In this sense, clusters of users can be identified in friendship circles, or in common interest groups where clusters within clusters exist [Santo and Fortunato, 2010]. Therefore, and due to the registered growth of wireless capable portable devices, a scalable routing scheme, which is resilient to mobility phenomena and capable of taking advantage of existing clusters and hierarchies, shall be defined.

1.2 Assumptions and Research Questions

Considering the dynamic nature and versatility of MANETs, both limited and enhanced by the nature of the devices where they are likely to be used, such as smart-phones and laptops, several restrictions must be contemplated when developing a scalable routing protocol. Due to the portability and availability of these wireless capable devices it is assumed that they are both used in indoor and outdoor scenarios. For instance, positioning information is expected not to be available in indoor scenarios, since Global Positioning System (GPS) solutions are not reliable in such scenarios, nor in urban scenarios where tall buildings exist. An alternative to GPS could be considered through the use of cellular fingerprinting [Piran et al., 2011] or by using sensors such as odometers [Grzonka et al., 2012], among other techniques. However, taking into account the reduced size of mobile devices, limitations regarding memory and processing power, as well as autonomy must be considered. Therefore, complex localization algorithms which frequently involve graph theory and which require substantial computational power, may not be suitable in many scenarios.

The identified limitations motivate the usage of a non-geographic protocol, to avoid the dependency on the assumptions of location information availability. Moreover, they further suggest that the overhead of the used routing protocol should be minimized, reducing the number of sent and received routing messages, avoiding state changes in the wireless card, therefore saving energy and increasing the devices' autonomy [Bernardo et al., 2011]. Taking this into account, reactive routing protocols aim at minimizing the number of sent routing messages by only sending them when data needs to be transmitted, finding routing paths on-demand. However simultaneous path requests may result in a heavy flooding of routing packets, known as the Broadcast Storm problem [Chuang et al., 2012], introducing additional interferences and collisions, possibly leading to a more expensive routing procedure than when compared with proactive routing protocols.

Another important aspect to be considered is the size of the network. More and more devices are expected to participate in networks where users exchange data amongst themselves. In this scenario, there is no need for a centralized server and data may be exchanged locally. There is a need, however, to balance the network load, using for instance clusters where routing information is contained, maintaining the routing operations scalable.

A well known scenario where MANETs are also considered important involves the transmission of critical data that must be delivered. In disaster scenarios, infrastructure-based communications are often unavailable and MANETs stand as an alternative. For emergency teams and other authorities, having a backup network capable of delivering sensitive information is paramount. Due to the nature of these networks, which are often intermittently connected, an important assumption is that the network's paths are self-repaired and that, even though some delay (i.e. a few seconds) may be introduced by these mechanisms, most of the important information is delivered instead of being dropped.

In order to allow a proper definition of the identified problem, only realistic assumptions were taken into consideration. These assumptions are expected to correctly identify the existing problems with scalable routing in MANETs and are listed bellow:

Assumption 1: *Increasing Number of Wireless Capable Devices.* Taking into account the evolution in technology and the increasing portability of wireless capable devices, each user is expected to individually own dozens of these devices [Cimmino and Donadio, 2009].

Assumption 2: *Portable and Ubiquitous.* Computers, phones and other gadgets tend to become smaller, and are carried by people in their every-day life. This results in constant changes in topology where some connections are broken and others created. Moreover, the demand for the ubiquity of services is also greater than ever, taking advantage of social-based interactions and professional networks.

Assumption 3: *Limited Resources.* A myriad of wireless technologies exists, however, off-the-shelf devices are considered in this work, since only mass produced devices are likely to be disseminated. Even though the number of features in these devices tends to increase, they still have a limited autonomy, which also limits their processing power. Memory limitations may also exist, as well as limited positioning information due to weak GPS modules. Usually these devices do not provide more than one wireless interface per technology and have a simple omnidirectional antenna and wireless card, which does not provide many configuration options.

Assumption 4: *Intermittently Connected Networks.* As a result of all the above mentioned restrictions and of the dynamic nature of wireless multi-hop networks, connectivity disruptions occur frequently [Nordemann and Tonjes, 2012]. Some existing solutions focus on DTN in order to avoid dropping packets, however a compromise must exist and protocols should consider a general constraint delay D so that most of the possible applications for MANETs are still feasible [Zhou and Ying, 2010]. Regarding these applications, most of them use unicast transmissions and therefore shall be considered during this work. Nevertheless, other applications, such as group communication, may take advantage of the broadcast nature of wireless networks and protocols can either use multicast or anycast transmissions for their purposes.

Having identified the assumptions involved within the scope of this work, the problem statement becomes clearer and the need for a scalable routing protocol in future autonomous MANETs is more obvious. However, several Research Questions arise in order to correctly define it. These research questions influence the approach followed in the development of a new routing scheme and are presented next.

Research Question 1: Protocol Design: How can the routing overhead be kept to a minimum?

  Research Question 1.1: Maintain a hierarchically organized clustered network.

  - ∟ What is the appropriate topology for such a hierarchical tree?
  - ∟ How can the clusters and their network views be correctly mapped to each child node of the tree?

  Research Question 1.2: Ensure routing amongst different clusters with lightweight updates.

  - ∟ What routing information is needed between different clusters?
  - ∟ Which method for exchanging routing updates is more efficient?

Research Question 2: Forwarding Decisions: How can Gateway nodes be efficiently selected according to their link quality?

  Research Question 2.1: Define a realistic link quality metric.

  - ∟ Are link metric extensions, such as those based on Received Signal Strength Indication (RSSI) efficient enough for wireless multi-hop networks?
  - ∟ How much do assumptions on node mobility and link behaviour influence link-quality estimators?

  Research Question 2.2: Determine a link quality estimator.

  - ∟ Which modelling tools provide a more flexible and robust estimator?
  - ∟ Are simulation traces valid for statistical-based models?
  - ∟ Does the obtained model perform well in scenarios completely different from those from where the traces were obtained?

Research Question 3: Protocol Performance: How can the performance of new protocol be assessed regarding future MANETs?

Research Question 3.1: Communication and Storage Complexity.

- ⌞ How can the protocol be tested? Are simulation based results accurate enough?
- ⌞ How expensive is the communication of routing packets?
- ⌞ How complex are the routing tables of the routing protocol?

Research Question 3.2: Ability to Scale.

- ⌞ In accordance with its expected complexity, is the protocol able to efficiently handle large-scale networks?

Research Question 3.3: Resilience to Node Mobility.

- ⌞ How much does a moving node disrupt the performance of the protocol?
- ⌞ What is the impact of a moving node in the existing hierarchical organization?
- ⌞ If a route is compromised should pending packets be dropped or rerouted?

Research Question 3.4: Robust Traffic Delivery.

- ⌞ In spite of all the existing mechanisms to keep the protocol scalable and robust, does the traffic delivery performance improve?
- ⌞ Which type traffic of patterns are expected? How much delay is reasonable? Are traffic losses acceptable?

1.3 Approach and Contributions

In order to address the aforementioned research questions, the approach taken began by identifying the scenarios where MANETs are potentially important and by performing an in-depth analysis of related works in the state of the art of routing in multi-hop wireless networks. Through this analysis it was revealed that there is a lack of efficient scalable routing protocols for these networks and, therefore, the following steps involved in the definition of a methodology to tackle this issue. Since a large number of wireless nodes is required to assess such a protocol, the usage of a network simulator was defined from the beginning. Nevertheless, the will to pursue a thorough evaluation of the work to be developed motivated the usage of the OPNET Wireless Suite simulation tool in order to use realistic propagation and interference models. Moreover, small scale tests were also planned using off-the-shelf wireless devices, in order to better understand the behaviour of the wireless link and how a protocol may be influenced.

These early decisions were helpful in the definition of a new routing scheme which aims at reducing the impact of node mobility and link failure, aggregating several clusters of nodes into different hierarchical views of the network. This process was named Deferred Routing, as it presents an innovative approach towards routing in MANETs, and is implemented in a protocol named Deferred Aggregated routing for Scalable ad-Hoc networks (DASH). Additionally, taking into consideration the dynamic behaviour of the wireless link, the routing process was further improved by devising a link-quality estimator that allows an efficient gateway selection among different clusters. The implementation of this solution in a well-known and validated simulator allowed an extensive performance study of the protocol, analysing not only its traffic performance, but also its scalability and resilience to network changes.

As a result of the work developed during the described process, several mechanisms were defined and implemented in the used network simulator. From the defined routing architecture the following contributions stand out:

Contribution 1, A new routing paradigm The definition of the Deferred Routing scheme represents a significant contribution, as it introduces a new perspective on how routing can be addressed in MANETs. This provides users and future researchers an alternative to existing approaches such as delay-tolerant, opportunistic or epidemic routing, efficiently dealing with mobility disruptions in real-time, while maintaining the entire process extremely scalable. The presented paradigm defines an aggregated management of clusters into hierarchical views which allows existing and future works to optimize the handling of routing information in large scale networks. The proposed scheme and hierarchy are capable of dealing autonomously with the transition of a node between clusters, requiring no additional routing messages and limiting the number of routing table updates solely to the two clusters involved in the node transition. Moreover, even though intra and inter cluster routing exists, the only exchanged routing messages are issued by the intra-cluster routing protocol. Whenever an intra-cluster routing packet is received by a node in a different cluster, instead of simply dropping that packet, the node "overhears" it and becomes a gateway node candidate, recovering all the relevant information about that cluster and its knowledge of the network, using it for future inter-cluster routing decisions.

Contribution 2, Deferred Routing The most significant contribution, eponymous of the routing scheme itself, is the process in which data packets are forwarded. The defined approach does not only take into account the end-to-end number of hops, instead it focuses on minimizing the number of traversed clusters (cluster hops), while choosing the most suitable gateways within each cluster.

Contribution 3, Efficient Gateway Selection A crucial aspect in Deferred Routing is the choice of appropriate gateways. This process not only considers the number of hops within the cluster but also the reliability of a gateway as such and its link-quality. In order to correctly perform this task, an assumption free Link-quality Model has been defined and validated, by using both simulation and real wireless traces. The obtained results suggest that the usage of Kernel Methods may provide robust estimators suitable for future implementations of Mobile Ad-hoc Networks.

Contribution 4, Performance Evaluation of Ad-hoc Protocols The main object under evaluation was the Deferred Routing (DefeR) protocol, however the obtained comparisons show that existing protocols suffer from scalability issues and fail to deliver traffic in challenging scenarios. The presented simulation results in this work provide important insights on how existing and future protocols can be correctly assessed, using new evaluation metrics, different scenarios and mobility models which are usually disregarded.

1.4 Thesis Structure

The organisation and structure of the thesis is depicted in Figure 1.1, and is the following:

Chapter 2 – Multi-hop Routing in Wireless Ad-hoc Networks Already existing works on Multi-hop Routing in Wireless Ad-hoc Networks are presented in Chapter 2, where the remaining open issues regarding scalability are analysed.

Chapter 3 – Deferred Routing In Chapter 3 the Deferred Routing approach is presented, detailing the proposal and the required steps to achieve scalable routing in Mobile Ad-hoc Networks.

Chapter 4 – Gateway Selection in Deferred Routing An inherent aspect of the DefeR routing protocol is the efficient selection of Gateway nodes, allowing a load-balanced forwarding of data packets. This is presented in Chapter 4.

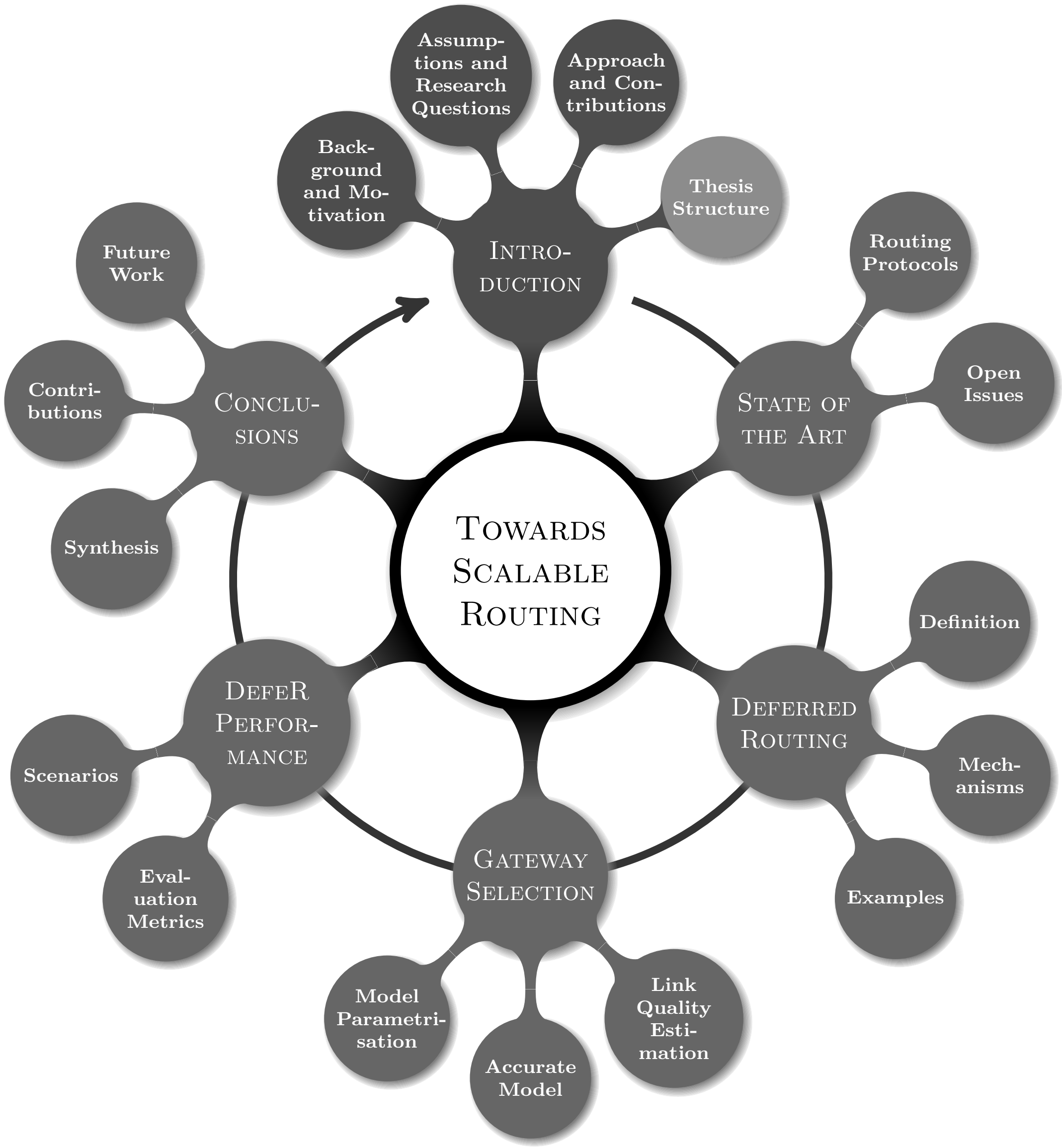


Figure 1.1: Thesis Structure

Chapter 5 – Deferred Routing Performance New and existing performance metrics are presented in Chapter 5, enabling a correct evaluation of the proposed routing scheme. Moreover, through an extensive set of simulations, using several different scenarios, this protocol is assessed and compared against two others.

Chapter 6 – Conclusion and Future Work Finally, in Chapter 6 the concluding remarks and final thoughts are presented, as well as future steps to further improve this work.

Chapter 2

Multi-hop Routing in Wireless Ad-hoc Networks

This chapter presents the State of the Art on Multi-hop Routing in Wireless Ad-hoc Networks. There are several protocols that had a paramount importance in the development of these networks, which will be described and analysed in this State of the Art chapter. However, taking into account the research questions and assumptions of this work, the main focus will be on hierarchical and cluster based protocols which aim at increasing the scalability of Ad-hoc networks. A taxonomy of these routing protocols is depicted in Figure 2.1, showing the relationships between the different approaches to routing.

This State of the Art on Mobile Ad-hoc Networks (MANETs) does not consist on an exhaustive listing of existing routing protocols, but instead on a thorough analysis of works relevant to the development of a new scalable routing solution, based on realistic assumptions as far as possible. Other developed works provide a more generalized and broad list of routing protocols including position-based, multicast, multipath or even power-aware approaches [Gupta and Gupta, 2010; Alotaibi and Mukherjee, 2012; Boukerche et al., 2011]. However, these works do not focus on the routing concept and present protocols which provide only minor changes to existing routing approaches. Some of these works are solely concerned with network awareness and dynamic routing, presenting extensions and new metrics to known routing protocols [Paillassa et al., 2011]. Even though several other extensions and approaches exist for MANETs, such as anycast routing [Chen and Wang, 2012] or store-carry-forward in Delay-Tolerant Networks (DTNs) [Nordemann and Tonjes, 2012], such topics are beyond the scope of this work.

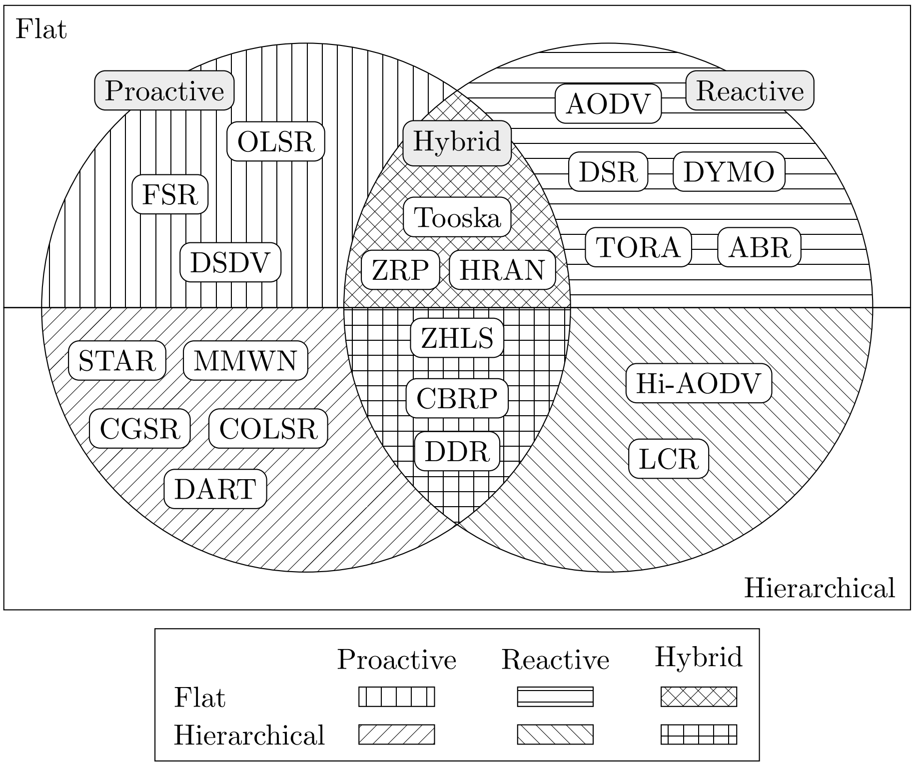


Figure 2.1: Taxonomy of MANET routing protocols

Despite all the provided mechanisms by each routing protocol, they are all subject to certain limitations and may fail in their purpose of scaling in large networks. Regarding the scalability of existing routing protocols their communication and storage complexity play an important role. Even though hierarchical solutions aim at being more scalable, this is not always true, since the complexity of these protocols is not necessarily better than flat solutions. Therefore, for each class of routing protocols, a comparison table highlighting the techniques used by each routing scheme as well as their complexity communication and storage complexity will be provided.

2.1 Flat Routing Protocols

Flat Routing Protocols are characterized for not having any particular hierarchy to help in the organization of the network. These are most the commonly found protocols and represent the foundation of MANET routing, being usually divided into Proactive, Reactive and Hybrid routing protocols.

2.1.1 Proactive Routing Protocols

Following the inspiration provided by typical routing protocols used in wired networks, proactive routing protocols are based on the periodic exchange of routing messages in order to maintain updated routing tables. This paradigm allows a prompt retrieval of the next-hop to where data should be forwarded. However, this periodic update always occurs, even when there is no data to be transmitted, wasting resources without need.

2.1.1.1 Highly Dynamic Destination-Sequenced Distance-Vector Routing Protocol

A routing protocol that results from a modification to the well known Distributed Bellman Ford algorithm [Hutson et al., 2007], the Destination-Sequenced Distance-Vector (DSDV) protocol [Perkins and Bhagwat, 1994] is a routing solution where looping related issues are efficiently solved. It is a multi-hop pro-active protocol where each node stores a routing table with one entry to all possible destinations and the number of hops to each node [Spaho et al., 2012]. In addition to this, not being dependent of any intermodal coordination mechanism allows the DSDV protocol to be robust solution for routing in MANETs. The protocol is also designed taking into account Medium Access Control (MAC) Layer details and sleeping nodes which should not be disturbed unless necessary, thus improving the total network lifetime.

The DSDV protocol periodically broadcasts update packets or whenever relevant information is available. These packets contain a new sequence number and information about the destination's address, the number of hops required to reach the destination and the sequence number of the previously received information regarding that destination. Routes containing the most recent sequence numbers are preferred when a path calculation is to be made.

A drawback from the standard DSDV implementation is observed when an existing path becomes invalid due to one or more broken links. When this occurs, the DSDV protocol assigns infinity to the path's metric and an odd sequence number (greater than the older one), which will be propagated through the network. However, while the link failure information is being propagated, some nodes will still drop several packets due to inaccurate information. This phenomenon is referred in literature as a stale route, requiring additional mechanisms to improve the response to a link failure. The Improved DSDV protocol [Lu et al., 2011] tackles this problem by maintaining a secondary routing table, which contains alternative routes to all the available destinations.

2.1.1.2 Optimized Link-State Routing Protocol

The Optimized Link-State Routing Protocol [Clausen and Jacquet, 2003] is a variant of the typical link-state routing protocols which inherits the advantage of having routes immediately available while, at the same time, providing adequate optimisations for Ad-hoc Networks. The main mechanisms used by Optimized Link State Routing (OLSR) to improve its performance are the exchange of a reduced and non-synchronized amount of control packets used for link sensing and neighbourhood detection. This improvement consists of an efficient flooding technique based on the selection of Multipoint Relays (MPRs), minimizing the required bandwidth for protocol operations and avoiding the reception of redundant control messages [Aponte and Bohacek, 2012]. An additional mechanism ensures that the required topology information is efficiently selected and diffused throughout the network.

Link Sensing By periodically sending $HELLO$ messages through the available wireless interfaces in which connectivity is confirmed (1-hop exchange between neighbour nodes), the OLSR protocol performs a link sensing operation. From this operation results a link set which contains the available information on “local” (1-hop) interfaces and on “remote” (2-hop) interfaces. This procedure may be replaced by link-layer information, if such feature is both available and sufficient to populate the link set, thus avoiding the exchange of $HELLO$ messages.

Each link contained in the link set is described by a pair of interfaces, the local and the remote interfaces, and it has associated to itself the status of being either symmetric or asymmetric, depending on whether it can respectively send and receive data packets.

Neighbourhood Detection The neighbourhood detection process consists in maintaining a set of neighbourhood tuples directly connected with the nodes’ main address. The relationship between the OLSR main address and additional addresses is defined through the exchange of Multiple Interface Declaration (MID) messages.

There is a clear relationship between the neighbourhood set and the link set earlier described. In fact a node may only be considered “neighbour” of another iif there is a link between each other.

In addition to the neighbour set, there is a 2-hop neighbour set consisting of a set of nodes which have a symmetric link to a symmetric neighbour, being all this information gathered from the exchanged $HELLO$ messages.

Still contained within the neighbourhood detection process, the population of both Multipoint Relay and Multipoint Relay Selector Sets is performed. MPRs are responsible for the existing flooding optimisation in OLSR as only them forward routing messages, avoiding a pure flooding approach where all nodes forward the protocol messages. Additionally they also avoid the transmission of duplicate messages by maintaining a Duplicate Set which records recently received messages as a “duplicate tuple” containing information about the originator address, message sequence number and a boolean indicating whether the message has been transmitted or not.

The selection of the MPR set is performed individually by each node which is responsible for selecting the most suitable nodes in its symmetric 1-hop set. This selection is performed in such a way that the node populating the set is able to reach all its strict 2-hop symmetric neighbours through the neighbours contained in the MPR set. Whenever changes occur in the 1-hop or strict 2-hop symmetric neighbours set, a complete recalculation of the MPR set is performed. Even though the MPR set does not have to be minimal, all strict 2-hop neighbours have to be reached through the selected MPR nodes and, in the worst case scenario, the MPR set may consist of the entire neighbourhood set, resulting in a typical link-state routing full flooding strategy.

The calculated MPR set may vary depending on the existing neighbourhood and on the nodes' willingness to act as MPR. This parameter is defined by the node depending on the available resources and other characteristics, being the values between WILL_NEVER and WILL_ALWAYS.

Finally, the MPR Selector set of a node n consists of all the addresses of nodes which have selected n as MPR.

Topology Discovery By performing the already mentioned link sensing and neighbour detection procedures, each node is able to communicate with the directly connected neighbour nodes and it can participate in an optimised flooding mechanism. However, this information has to be disseminated through the entire network in order to allow the construction of routes to every node. This is done by MPR nodes which periodically send a Topology Control (*TC*) message with a set of links, known as Advertised Link Set, which contains the links to all the nodes in the MPR Selector set.

MPRs broadcast *TC* messages, flooding them to all the nodes in the network using other MPRs to efficiently improve the distribution of topology information, enabling greater scalability.

The generation of Topology Control messages is periodically performed by all MPR nodes in a time interval defined by the constant TC_INTERVAL, which can have several values such that, for a lower interval, a higher capacity of reaction to link failures is achieved. Related with failures, whenever a change to the MPR Selector set is detected, possibly due to a link failure, a new *TC* message should be sent earlier than the next interval generated message.

A common problem inherent to proactive protocols is the synchronization of control messages such as *HELLO* and *TC* messages. This increases the network overhead and may lead to losses due to collisions. In order to avoid this phenomenon, which typically arises with periodically sent messages, the OLSR protocol randomly defines a value named *jitter* which should be between 0 and MAXJITTER. This value is used in the actual message interval by subtracting the *jitter* value to it, thus varying the period in which messages are sent, and avoiding equal message transmission times which may synchronize.

OLSR v2 Currently under development by the Internet Engineering Task Force (IETF) MANET working group, a new version of the OLSR protocol, the Optimized Link State Routing version 2 (OLSRv2) [Clausen et al., 2012], proposes an update to the mechanisms of its predecessor. Even though the main algorithms are maintained, this new version offers a more modular and therefore flexible architecture, allowing, for instance, the addition of security extensions without compromising backwards and forwards compatibility [Herberg and Clausen, 2011]. Moreover, it also uses the Neighborhood Discovery Protocol (NHDP) [Herberg et al., 2012] for the discovery of 1-hop and 2-hop neighbours, as well as discovering whether links are bi-directional, by sending *HELLO* messages similarly to the standard version of OLSR. The OLSRv2 protocol also implements the MPR Flooding process so that the link state information advertised by the protocol is efficiently propagated.

2.1.1.3 Fisheye State Routing Protocol

The proactive Fisheye State Routing (FSR) protocol [Pei et al., 2000] is inspired and takes its name from a well known technique proposed by Kleinrock et al. named Fisheye [Kleinrock and Stevens, 1971], originally used to reduce the size of samples required to represent graphical data. Similarly to a fish's eye, where the images are more detailed closer to the eye's focal point, a node using the FSR protocol has a better perception of its closer neighbourhood, updating information about more distant nodes with a lower periodicity.

As a link-state routing protocol, the FSR protocol maintains a topology map of the network at each node. However, instead of flooding a network change when such is detected, it proposes a different scheme for information dissemination [Sivakumar and Chelliah, 2012].

In order to reduce routing control overhead, instead of sending routing updates at a fixed period, the FSR protocol uses different time intervals to exchange its routing information with nodes at different distances. Each node receives routing updates from further away nodes less frequently, maintaining a less accurate view of distant routes. However, whenever data is forwarded through the network, the precision of the used routes gradually improves as it gets closer to the desired destination.

2.1.1.4 Flat Proactive Routing Protocols – Comparison

Proactive routing protocols stand out for always maintaining routes to all the available destinations. In flat organisations clustering is not typically used however other scalability mechanisms can be found. Table 2.1 shows these mechanisms for the presented routing protocols and analyses their complexity regarding storage and communication.

Table 2.1: Comparison of Flat Proactive Routing Protocols

Protocol	Cluster-based	Scalability Techniques	Communication	Storage
DSDV	no	n/a	$O(N^2)$	$O(N)$
FSR	no	fisheye updates	$O(N^2)$	$O(N)$
OLSR	no	MPR nodes	$O(M^2)$	$O(N)$

N : Total number of nodes
M : Total number MPR nodes

2.1.2 Reactive Routing Protocols

Proposed as an alternative to the expensive periodic update of proactive routing schemes, reactive protocols were introduced, performing route discoveries on-demand to avoid the waste of resources experienced with proactive solutions. This approach seems more suitable for mobile Ad-hoc networks where topology changes occur constantly. However, on-demand solutions suffer from an initial delay on retrieving a routing path which may not be acceptable, while at the same time the flooding of Route Request (RREQ) for route retrieval also adds an increased network overhead. In fact, several works aim solely at reducing the Broadcast Storm Problem (BSP), which was named after the broadcasting process typically involved in the dissemination of routing information of on-demand routing protocols [AlAamri et al., 2012].

2.1.2.1 Ad hoc On-demand Distance Vector Routing Protocol

Designed for mobile wireless Ad-hoc networks, the Ad hoc On-demand Distance Vector (AODV) [Perkins et al., 2003b] routing protocol requires low memory and processing, while providing quick adaptation to dynamic link conditions. In addition to this, AODV has a low communication overhead and provides loop-free unicast routes in a reactive way, without having to maintain routes to destinations that are not currently in use.

Upon a request of a route to a destination, RREQ Packets are broadcasted throughout the network nodes until they reach their final destination or, alternatively, until an intermediate forwarding node, already containing an active/updated path to the destination, responds with a Route Reply (RREP) packet. Since each forwarding node keeps a reference of the source node triggering a RREQ, as well as the neighbour node likely to be used as next hop towards that destination, when processing a response message (a RREP), a node will route it back along the expected hops until it reaches the source node, making the new path available [Gandhi et al., 2012].

For efficiently managing the above described process, AODV uses sequence numbers to avoid loops and keep awareness of updated routes. Additionally, an assumption of bidirectional links is also present when a RREP is sent to its originating node. However, if unidirectional links exist, an alternative procedure needs to be used, in order to allow these packets to be correctly replied.

One important practice to be considered in AODV is the usage of an expanding ring search technique. This measure aims at preventing unnecessary network-wide dissemination of RREQ messages by controlling the extent to which these packets are broadcasted. This optimisation can be achieved by effectively setting some AODV specific parameters to the most appropriate level.

In order to keep accurate information about the active routes and avoid disruptive failures, each node monitors the link status of their next hops in these paths. The monitoring process is typically achieved by exchanging *HELLO* messages through the links, even though other mechanisms may be used. Upon the detection of a link break, a Route Error (RERR) message is used to notify the other nodes present in the path that a link loss occurred. Then, after receiving this message, the source node may decide to re-trigger a new RREQ, setting up a new route. Some extensions to AODV have already been proposed to address this specific point, for instance the Ad hoc On-demand Distance Vector Backup Routing (AODV-BR) Protocol [Lee and Gerla, 2000].

The authors propose a modified version of the AODV protocol which not only uses the Expanded Ring Search (ERS) mechanism, but also a new approach named Hop Prediction, which improves the route search used by AODV. History records are maintained to each discovered route in order to optimise the ERS and reduce the overall routing overhead.

2.1.2.2 Dynamic Source Routing Protocol

An example of a completely reactive protocol with support of unidirectional links is the Dynamic Source Routing (DSR) [Johnson et al., 2007] protocol proposed in the IETF MANET working group [MANETWG, 2012]. Aiming at scalability, in a network of at most two hundred nodes, the DSR protocol provides a soft-state approach where the two basic operations are Route Discovery and Route Maintenance, supporting asymmetric routes and assuming a typically small network diameter.

Designed for Internet Protocol version 4 (IPv4) addresses, provided by any mechanism such as Dynamic Host Configuration Protocol (DHCP) for dynamic assignment or static configurations, the DSR protocol is a loop-free protocol capable of quickly adapting to network topology changes. These adjustments of the topology only have an impact on the protocol when they affect paths currently active, being ignored by any other nodes. However, in order to avoid routing based only on flooding, topology changes related with mobility or other circumstances are not expected to happen so fast that the DSR protocol cannot adapt.

DSR uses explicit source routing, where an ordered list of nodes through which the discovery packet will pass, from source to destination, is used to allow multiple paths that enable the usage of load balancing mechanisms [Rajkumar et al., 2012]. It also enhances the protocol robustness by tolerating path failures, choosing alternative ones immediately. Route caching is also an interesting feature that results from the forwarding and overhearing nodes' action of gathering information that can be used in the future, avoiding the Route Discovery process. When queried about a path, by performing a search in its local cache, a node can immediately retrieve the desired route and avoid further overheads of a Route Discovery process.

In the worst case scenario, when a complete Route Discovery has to be performed, the first node, the initiator, transmits a RREQ that will be broadcasted to all of the nodes until it reaches the destination node, the target. When this node is finally reached, it checks for a previous path cached to the initiator and sends a RREP. Otherwise it will start a new RREQ for the initiator, piggybacking the list retrieved by the first Route Discovery. Optionally, the target could simply reverse the path contained in the list given by the received RREQ, avoiding additional overhead but losing the asymmetric path support property.

Route maintenance in DSR states that each node is responsible for managing the flow over the link from that node to the next hop. This can be done either by using software or hardware acknowledgements, and a limited number of retransmissions. After the maximum number of retransmissions, a link is said broken and so the link is removed from Route Cache and a RERR is returned. If an alternative path exists in the initiator it shall be used, otherwise a new Route Discovery should be triggered.

2.1.2.3 Temporally-Ordered Routing Algorithm

Being a member of the link-reversal algorithms class, the multi-path and loop-free Temporally-Ordered Routing Algorithm (TORA) [Park and Corson, 2001], is an on-demand source initiated routing protocol designed for multihop networks, which can also have destination initiated proactive routing for path optimisation and maintenance purposes.

Concerning routing, TORA routers only keep information about their one-hop neighbours and perform on-demand routing when retrieving a path to a destination. This operation performs best in networks with relatively sparse traffic patterns. At the same time, destination oriented mechanisms can also be triggered to maintain and monitor the path.

Summarizing TORA, it can be defined as four separate basic functions, namely creating routes, maintaining routes, erasing routes and optimising routes. For this, four different packet types are used: Query, Update, Clear and Optimisation [Lim and Datta, 2012]. TORA is an interesting protocol from the point of view that it does not use shortest paths to support its decisions and neither does it follow a link-state nor distance-vector algorithm.

2.1.2.4 Dynamic MANET On-demand Routing Protocol

Much resembling with DSR and AODV, Dynamic MANET On-demand (DYMO) routing protocol [Chakeres and Perkins, 2012] is a reactive loop-free routing protocol. Designed for networks with bidirectional links and capable of handling a wide range of mobility patterns, by dynamically determining routes in large scale networks, DYMO is best suited for sparse traffic scenarios. Having only to maintain minimal routing state information, it is a light-weight protocol applicable to devices with memory constraints.

The most relevant operations of the DYMO routing protocol are similar with DSR's Route Discovery and Route Maintenance. The former starts with the initiator node by sending a RREQ Packet to be broadcasted by all nodes until it reaches the desired target destination, which then replies with a RREP Packet through the best path, defined by a list that contains all the RREQ forwarding nodes. In order to reduce RREQ overhead, a forwarding node containing an active path to the destination may automatically respond with a RREP packet on behalf of the target node, avoiding further propagation of messages. An additional consideration is the usage of an adequate value for the HopLimit parameter which, to delimit the expanding ring of a RREQ, may be defined as described for the AODV protocol.

Complementing the above presented process, the Route Maintenance procedure is responsible for safeguarding the existing routes in use [Kumar et al., 2012]. Route lifetime is extended by routers whenever a packet is correctly forwarded or a RERR packet is sent towards the packet source to indicate that the path contains an invalid or missing node. Additionally, by monitoring links over which traffic is flowing, any broken link detection should also immediately issue a RERR packet in order to swiftly notify DYMO nodes that certain routes are no longer available.

2.1.2.5 Associativity Based Routing

The Associativity Based Routing (ABR) [Toh, 1996] principle consists on the fact that after some migration process, where associativity ticks can be analysed, a certain stability time will exist, where a node will stay dormant within a cell before it moves again. The associativity tickets are analysed on the link layer level allowing to understand the degree of mobility of a node, where low associativity tickets are a synonym of a higher state of mobility and, on the other hand, high associativity ticks represent a stable state [Preveze and Safak, 2010].

Route Discovery and Route Re-Construction are the two phases that compose the ABR protocol. During the Route Discovery phase a Query packet is broadcasted from source to destination, which then replies with a Reply message. The Query message is forwarded by every intermediate node that will keep the information of its upstream peer, removing it from the original packet and adding its own. If a duplicate Query is received by a node, it will be discarded. When a Reply message is sent back by the destination, nodes receiving this packet will set the path from source to destination as valid and active. Other nodes containing alternative paths will have them marked as invalid and will not relay packets to the destination, even if they hear the transmission.

Complementing the Route Discovery process, the Route Re-Construction phase handles possible failures caused by mobility or other situations by performing a partial route discovery, invalid route erasure, valid route update and, in the worst case scenario, new route discovery, which consists in the repetition of the entire processes described for the Route Discovery Phase.

2.1.2.6 Flat Reactive Routing Protocols – Comparison

Reactive routing protocols aim at being more lightweight than proactive ones by sending routing information only when necessary. However, this approach may result in expensive flooding of RREQ whenever a route is required. In additional to this limitation, which more critical in scenarios with several traffic flows, these protocols also suffer from a route retrieval delay. Even though the communication and storage complexity of reactive protocols is expected to be lower than a proactive routing protocol, as they only consider the necessary destinations. In a worst case scenario for reactive routing protocols, each node may be a source and destination node, resulting in a complexity similar to proactive protocols, as shown in Table 2.2.

Table 2.2: Comparison of Flat Reactive Routing Protocols

Protocol	Cluster-based	Scalability Techniques	Communication	Storage
ABR	no	Associativity Tickets	$O(N^2)$	$O(N)$
AODV	no	modified ERS	$O(N^2)$	$O(N)$
DSR	no	n/a	$O(N^2)$	$O(N)$
DYMO	no	HopLimit	$O(N^2)$	$O(N)$
TORA	no	Directed Acyclic Graph	$O(N^2)$	$O(N)$

N : Total number of nodes

2.1.3 Hybrid Routing Protocols

Recognising both the advantages and disadvantages of proactive and reactive routing protocols, hybrid routing protocols were proposed. The concept behind this new alternative is that the best of each approach (proactive and reactive) can be exploited together in the different tasks performed by a routing protocol.

2.1.3.1 Zone Routing Protocol

Combining the advantages of the pro-active and reactive paradigms, the Zone Routing Protocol (ZRP) [Haas et al., 2002], proposes a zone based architecture where three embedded protocols, the Intra-zone Routing Protocol (IARP), the Inter-zone Routing Protocol (IERP) and the Bordercast Resolution Protocol (BRP), are responsible for maintaining the routing operation.

Assuming that a majority of the processed traffic occurs directly between neighbour nodes, the strategy of ZRP is to reduce the scope of proactive traffic into a zone centred on each node. The zones are defined as having a r radius expressed in hops, such that a zone includes nodes whose distance from a given node is at most r hops. Since zones overlap, ZRP is said to have a flat view of the network. This perspective results from the authors' statement that this approach can be used to detect optimal routes and to reduce network congestion.

The IARP is the protocol used within ZRP zones to proactively maintain routing tables up-to-date. In contrast, route discovery outside of a specific zone is made by the reactive IERP protocol. Using the information of IARP, an additional routing procedure is made by BRP, which consists in managing the packet delivery to the peripheral nodes in the border of a zone (bordercasting). The usage of this approach in conjunction with IERP allows a reactive route discovery to efficiently travel between zones [Yelemou et al., 2012].

The size of the zones used by ZRP can be managed by regulating the transmission power of devices (if such option is available). Additionally, mechanisms to efficiently and possibly dynamically choose r should be used, ensuring that a zone is big enough to provide a good connectivity between nodes, but not too big so that update traffic does not become excessive. However, such a dynamic process is complex and not easy to achieve [Patel and Srivastava, 2010]. Further works provide analytical models that determine the routing overhead incurred by the ZRP protocol and its variants. Some examples are the Independent Zone Routing Protocol (IZRP) [Samar et al., 2004], which proposes mechanisms for calculating the optimal zone radius of the node, being more efficient than the standard ZRP [Ravilla et al., 2011]. These mechanisms are known as min-searching and adaptive traffic estimation, and allow each node to have its own independent zone size. The Two-Zone Routing Protocol (TZRP) [Wang and

Olariu, 2004], that presents a zone-based architecture that decouples the (basic hybrid) protocol's ability to adapt to changing traffic patterns from the ability to adapt to different mobility models. And also the Fisheye Zone Routing Protocol (FZRP) [Yang and Tseng, 2007], where the architecture defined by the ZRP uses Fisheye State Routing in its proactive operations.

2.1.3.2 Tooska Scheme and Mobility Aware Hybrid Routing

The Tooska Routing scheme [Dargahi et al., 2008] is a hybrid node-centric protocol which relies on AODV as its default routing protocol, switching to the Wireless Routing Protocol (WRP) [Murthy and Garcia-Luna-Aceves, 1996] when appropriate. By selecting the nodes with more stable fixed neighbours, the core nodes, the protocol defines these intermediate nodes when data needs to be sent, through the analysis of the *HELLO* Message Counter (HMC) field stored by each node. Core nodes periodically update their routing tables by changing to the WRP protocol, informing all the remaining nodes of this change. In order to reduce the overhead introduced by the Tooska scheme, the number of core nodes is minimized by defining a minimum number of required stable neighbours.

Due to node mobility, the selection of core nodes can become inefficient in the Tooska scheme as it is proposed. Bearing this in mind, the Mobility Aware Hybrid Routing (MAHR) [Kang et al., 2012] defines an alternative selection method for core nodes, where the ratio of changing neighbour nodes is used. The routing process is similar to Tooska relying on the AODV protocol for route discovery, where the core nodes are responsible for the maintenance of routing tables by using the OLSR protocol.

2.1.3.3 Heat Routing for Ad-hoc Networks

Parallel to the behaviour of heat trails in the physical world, wireless nodes using the Heat Routing for Ad-hoc Network (HRAN) protocol [Trindade and Vazao, 2011a,b] emit a heat signal to be perceived by neighbour nodes. The amount of heat detected by each node depends on a gradient function such that nodes further away from the heat source register a lower level of heat when compared with 1-hop distant nodes.

The protocol's main mechanisms consist on the creation of a heat overlay, where each node proactively disseminates its topology information, with an amount of heat defined by a Time Aware Bloom Filter (TAB) which is a new type of Bloom Filter defined by the authors. The heat information is included in periodically sent *HELLO* messages, as the size of the used TAB never changes regardless of network size, creating a heat overlay or heat trails.

By using an on-demand approach, the second stage of the HRAN protocol consists on discovering a valid route from source to destination. This is achieved by issuing a predetermined number of Random Walk Request (RwREQ) queries, to be sent throughout the network. Upon receiving a RwREQ, a node checks if the received destination identifier is present in any of its registered heat trails. If a match is obtained, the random walk is terminated and a direct walk takes its place. This walk is started by sending a Follow Heat (FoHEAT) message which is only forwarded by nodes in the same heat trail, allowing the query to quickly reach the destination.

When a RwREQ reaches its intended destination node, it sends a Route Reply (RoREP) message to the source, using the discovered route, inverted. If after sending a RwREQ, a predefined time-out is reached and no RoREP is received by the source node, the protocol falls back to a typical reactive source routing protocol such as AODV. This mechanism is important as it allows the creation of heat tunnels which otherwise are only created after a route establishment, during the route maintenance process.

The final contribution of the HRAN protocol is the maintenance of routes which include the creation of heat tunnels, achieved by adding the destination's identifier in the proactively sent routing messages. This creates a “highway” for future route requests to this destination. Moreover, this process also ensures that failed routes are repaired by sending Route Repair (RoREPAIR) messages and it further aims at improving the found routing path. Since the first retrieved path may not be the shortest path due to the randomness of route discovery process, an additional message named as Route Improvement (RoIMP) is sent by the source within the heat tunnel, until it reaches the destination. In its turn, the destination sends back to the source a new RoREP, using an inverted more efficient path.

2.1.3.4 Flat Hybrid Routing Protocols – Comparison

Table 2.3 presents a comparison of the main characteristics of the analysed hybrid routing protocols with a flat network organisation. As a direct consequence of employing both proactive and reactive routing protocols their complexity is similar to these protocols. However, these protocols also provide optimisations that may enhance the protocols performance in many situations. Moreover, the usage of zones by the ZRP reveals an alternative to achieve a more scalable routing process.

Table 2.3: Comparison of Flat Hybrid Routing Protocols

Protocol	Cluster-based	Scalability Techniques	Communication	Storage
HRAN	no	Heat overlay	$O(N^2)$	$O(N)$
Tooska	no	HMC	$O(N^2)$	$O(N)$
ZRP	yes (zones)	Variable zone radius	$O(Z^2)$	$O(\frac{N}{Z})$

N : Total number of nodes
Z : Total number of zones or cluster-heads

2.2 Hierarchical Routing Protocols

The definition of specific hierarchies by different routing protocols has commonly been used, aiming at keeping the protocols more scalable. In contrast with typical flat routing protocols, hierarchical protocols usually exchange their routing information in different ways, according to a cluster or node hierarchy level. Well defined hierarchies are usually more common in hybrid routing protocols, however, hierarchical routing can also be found in proactive and, even though less frequently, in reactive routing protocols.

2.2.1 Proactive Routing Protocols

The usage of hierarchies in conjunction with proactive routing approaches can be observed as a hierarchy of clusters, as an organized tree of addresses, or even as trees of paths forming a topology. Several schemes exist and all attempt to efficiently handle routing with the least overhead possible, as presented next.

2.2.1.1 Source-Tree Adaptive Routing Protocol

The Source-Tree Adaptive Routing (STAR) protocol [Garcia-Luna-Aceves and Spohn, 1999; Garcia-Luna-Aceves and Roy, 2005], is a link-state protocol which has on average less overhead than on-demand routing protocols. Its bandwidth efficiency is accomplished by restraining the dissemination of link-state information only to the routers in the data path towards the desired destinations. STAR also creates paths that may not be optimal while avoiding loops, such that the total available bandwidth is increased. Moreover, STAR has specific mechanisms to know when update messages must be transmitted to detect new destinations, unreachable destinations, and loops.

Despite being able to scale, as each node only maintains a partial topology graph of the network, the STAR may suffer from large memory and processing overheads in scenarios where constant mobility may report different source trees, and routing paths are too big due to the network size.

2.2.1.2 Multimedia support in Mobile Wireless Networks

In the work entitled Multimedia support in Mobile Wireless Networks (MMWN) [Kasera and Ramanathan, 1997], the authors propose an architecture consisting of two main elements, corresponding to different node types, which can either be switches or endpoints. Both of these can be mobile, however only switch nodes can route packets and only endpoints can be sources or destinations for packets. This protocol also keeps a cluster hierarchy as a location management scheme, capable of obtaining the address of an endpoint. This information is kept as a dynamic distributed database, such that in each node there is a location manager node.

The proposed hierarchy allows the necessary amount of routing messages to be reduced, as only location managers are required to update their information and only then perform the location finding process [Sehgal and Gondal, 2007]. However, this aspect is also negative on the overall performance of the protocol, as routing is strongly related with the hierarchy of the network, making the routing process complex and more vulnerable to disruptions when location managers change.

2.2.1.3 Cluster-head Gateway Switch Routing

Using the mechanisms introduced by DSDV, another proactive hierarchical routing protocol is the Cluster-head Gateway Switch Routing (CGSR) protocol [Ching-Chuan et al., 1997], which uses a routing approach where clusters are formed by electing a cluster head node, aiming to reduce the communication overhead, and thus making routing scalable and efficient. After the election of a cluster head, all nodes within its range will be considered as belonging to that cluster and all route updates should be done within its scope. All route discovery packets are forwarded through the cluster-head node.

One important task of this protocol is, essentially, the clusterhead election process. Authors argue that, when using distributed clustering algorithms, two possible choices are the lowest-Identifier (ID) algorithm and the highest-connectivity (degree) algorithm. The most important aspect to be taken into consideration when picking a clustering algorithm is stability. In order to avoid constant cluster head changes, which can harmfully impact the performance of other underlying protocols being used (such as DSDV), the algorithm chosen by the CGSR protocol is the Least Cluster Change (LCC) clustering algorithm [Suresh and Duraiswamy, 2010]. This clustering algorithm is proposed as an improvement to existing algorithms, achieving enhanced stability.

Even though the proposed two-level cluster hierarchy may reduce the amount of flooding for dissemination of routing information, as only the cluster-heads are responsible for this task, the process of maintaining these clusters involves additional overheads, in particular the election of an appropriate cluster-head node. Moreover, this special node will always represent a bottleneck on each cluster, overloading it and possibly leading to a faster energy depletion and consequent cluster-head re-election.

2.2.1.4 Cluster-based OLSR extensions to reduce control overhead in mobile ad hoc networks

The work entitled "Cluster-based OLSR (C-OLSR) extensions to reduce control overhead in mobile ad hoc networks" [Ros and Ruiz, 2007], proposes an extension to the OLSR protocol by introducing a cluster organised network. The authors propose a scheme where the existing clusters are considered as nodes themselves, using the MPR concept created by OLSR applied to clusters. This structure, in conjunction with the definition of Cluster *HELLO* (C-*HELLO*) and Cluster Topology Control (C-*TC*) messages, allows the maintenance of paths among the existing clusters while reducing the required amount of routing information, as only MPR Clusters generate C-*TC* messages.

Even though this paper uses the OLSR protocol for intra-cluster routing, proposing the mentioned C-*HELLO* and C-*TC* extensions to support a clustered network, the propagation of these new messages across clusters may have a negative impact. Moreover, the proposed mechanisms may suffer from mobility phenomena which, as in other approaches, require an additional overhead of updating the entire network structure.

2.2.1.5 Dynamic Address Routing for Scalable Ad-hoc and Mesh Networks

Inspired on a previously work on a Dynamic Addressing paradigm, the authors propose Dynamic Address Routing (DART) for Scalable Ad-hoc and Mesh Networks [Eriksson et al., 2007], a proactive hierarchical approach that efficiently manages the organization of nodes into zones for large scale networks. Address allocation and lookup are the main drawbacks of this proposal. However, the published work presents schemes to tackle these problems, showing how addresses can be allocated taking into account node positioning, by building a tree with l levels where l is the number of bits used in the routing address. A clear distinction is made between routing address and the identity of a node (a unique identification tag) since the routing address is dynamic and changes with node movement, contrasting with the node identifier which is always the same.

The three most important functionalities in DART are, first, the address allocation responsible for maintaining one routing address per network interface according to the movement and current position of a node; second, the routing which determines how to deliver packets from source to destination and, finally, the node lookup which consists in a distributed lookup table in charge of mapping identifiers to network addresses.

The DART proposal reveals to be an efficient solution for routing in large scale Ad-hoc networks. However, for small networks the Dynamic Address Heuristic has a strong overhead impact and in general it is difficult to implement, as the distributed lookup table is hard to manage.

Tree-like Distance Vector Inspired by the work presented in DART, the Tree-like Distance Vector (TLDV) routing protocol [Wang et al., 2010] uses a $2^b - ary$ tree locator and Distributed Hash Table (DHT), as opposed to DART's binary tree. The protocol also maintains at each node a routing and a neighbourhood table, being the routing table organized into $\lceil \log_{2^b} N \rceil$ rows with $(2^b - 1)$ entries each, in a network

Table 2.4: Comparison of Hierarchical Proactive Routing Protocols

Protocol	Cluster-based	Scalability Techniques	Communication	Storage
CGSR	yes	Cluster-head	$O(C^2)$	$O(N)$
C-OLSR	yes	Cluster-MPRs	$O(C^2)$	$O(N)$
DART	yes (zones)	Dynamic Addresses	$O(\log_2 N)$	$O(N)$
MMWN	yes	Location Managers	$O(N^2)$	$O(N)$
STAR	no	Partial Topology	$O(N)$	$O(D)$

N : Total number of nodes
D : Total number of destinations
C : Average number of nodes per cluster

with N nodes. A major contribution from the TLDV protocol is not being restricted to the binary tree used by DART, exploiting a different space structure. However, the choice of parameter b needs careful consideration and strongly depends on the network's intrinsic properties. A trade-off between lower and higher values of b must be achieved between the size of the routing table, the amount of available locators and also route efficiency.

2.2.1.6 Hierarchical Proactive Routing Protocols – Comparison

Even though hierarchical proactive routing protocols present more scalability oriented features than flat ones, the communication and storage complexities are not necessarily better. Moreover, the mechanisms used for this purpose, presented in Table 2.4 can be quite complex and introduce additional overheads that are not accounted as routing overheads. Nevertheless, the presented routing protocols require that all the presented aspects are present and therefore may not be as flexible as desired.

2.2.2 Reactive Routing Protocols

The usage of Hierarchical Reactive Protocols is modest when compared with proactive or hybrid routing approaches. This is likely due to the fact that most well defined hierarchies require constant updates in order to be efficiently kept, going against the concept behind Reactive Routing, which only exchanges routing information when required. Nevertheless, some Hierarchical Reactive protocols do exist and are described in the following paragraphs.

2.2.2.1 Hierarchical AODV Routing Protocol

As the name indicates, the Hierarchical AODV (Hi-AODV) Routing Protocol [Ohta et al., 2004; Oda et al., 2007] is a hierarchical version of the well known AODV routing protocol, using a tree based on cluster-heads for the creation of the concept of virtual nodes, which correspond to a typical cluster. The cluster-head is the only node responsible for handling control packets and managing the routing table of its own internal cluster. Having a tree composed of clusters seen as a virtual node allows Hi-AODV to reduce the number of control packets and avoid additional overhead.

In addition to the already mentioned challenges and overheads related to the maintenance of clusters and their cluster-heads (e.g. the cluster-head election process), again, it is clear that even though routing overheads can be reduced, the cluster-head will always have to be part of any routing path, leading to non-optimal paths, and additional interferences in the vicinities of cluster-heads.

2.2.2.2 Layered Cluster-based Routing

The Layered Cluster-based Routing (LCR) protocol [Jemili et al., 2010] is a hierarchical reactive protocol which exploits the main features of the Tiered Based Clustering Algorithm (TBCA) [Jemili et al., 2008] also proposed by the same authors. This clustering scheme is organized into layered stages so that the number of nodes participating in the clustering process, at a given instant, is reduced. By the end of the clustering process a connected dominating set consisting of the elected Cluster-heads and Gateway nodes is formed.

Using an on-demand approach, the LCR protocol restricts its search space to the dominating set retrieved from TBCA. Whenever a new route is required to reach a destination, the initiating or source node broadcasts a RREQ and waits for a specific time interval before issuing a new request. This request is only propagated by dominating nodes, which maintain a table of previous requests (Table_request) in order to refrain a duplicate request, thus avoiding additional overhead. When the destination node receives a RREQ, similarly to the AODV protocol, it sends a RREP and initiates a route maintenance process which periodically exchanges *HELLO* messages between the nodes involved in the route, sending a RERR message if a route failure is detected.

Additional mechanisms used by LCR concern the sensing period of the source and dominating nodes. In fact, the source node's sensing wait time is set to a sensing period equal to Short InterFrame Space (SIFS), where the cluster-head's waiting time is equal to Point coordination InterFrame Space (PIFS) and the Gateway (GW)'s waiting time is equal to Distributed InterFrame Space (DIFS). These specific times are set in order to reduce the probability of collisions during the discovery phase.

Optimized Layered Cluster-based Routing An update to the LCR protocol was provided by its original authors [Jemili et al., 2011b], optimising the MAC-layer mechanisms to avoid collisions and defining a direction mechanism that reduces the number of dominating nodes involved in the routing process. This direction-based mechanism is free from any positioning techniques, such as Global Positioning System (GPS), using information included in the resulting layers from the clustering process and allowing dominating nodes to discard any RREQ when, for instance, it reaches higher layers than the layer where the destination is expected to be. In certain scenarios where this mechanism may not be available, the LCR protocol performs normally without any disadvantages.

Table 2.5: Comparison of Hierarchical Reactive Routing Protocols

Protocol	Cluster-based	Scalability Techniques	Communication	Storage
Hi-AODV	yes	Cluster-heads as Virtual Nodes	$O(C^2)$	$O(N)$
LCR	yes	TBCA	$O(C^2)$	$O(N)$

N : Total number of nodes

C : Total number of dominating nodes or cluster-heads

2.2.2.3 Hierarchical Reactive Routing Protocols – Comparison

In the existing literature there are few Hierarchical Reactive Routing protocols since maintaining a hierarchy typically requires constant updates. Table 2.5 compares the performance of the two protocols which depends entirely on the robustness of the used clustering processes.

2.2.3 Hybrid Routing Protocols

Quite a few Hybrid Routing protocols for Ad-hoc networks can be found in the literature, however, despite the fact that many rely on clusters or well defined zones, not many implement a hierarchical routing scheme. The following protocols propose a hybrid routing scheme capable of retrieving inter-cluster information in a reactive approach, avoiding the necessity of restraining routing information in cluster-heads to reduce the overall overhead. However, on a downside, inter-cluster communication may be subject to route retrieval delay if no previous path has been maintained in cache.

2.2.3.1 Zone-based Hierarchical Link-State Routing Protocol

The Zone-based Hierarchical Link-State (ZHLS) Routing Protocol [Joa-Ng and Lu, 1999], is characterized by dividing the network into non-overlapping zones where two different routing paradigms are used: proactive routing within the zones and reactive between different zones. This proposal alleviates single points of failure and bottlenecks by not being dependent on cluster-head nodes and, at the same time, by maintaining a scalable hierarchy based topology.

One important assumption, and a possible limitation from this protocol is that each node knows its own position (for instance, by using GPS) and consequently its zone

ID which is directly mapped to the node position. With this approach, packets are forwarded by specifying in their header the zone ID and node ID of their destination.

The division of the network into a number of zones depends on factors such as node mobility, network density, transmission power and propagation characteristics. The geographic awareness is much more important in this partitioning process as it facilitates it when compared to radio propagation partitioning [Song and Lutfiyya, 2009].

In addition to the limitation of requiring some positioning system, the ZHLS protocol requires that all nodes exchange inter-zone flooding information when only gateway nodes need this routing information for calculating the shortest path between different zones. Moreover, the ZHLS is susceptible to a route retrieval delay when establishing inter-zone paths, as reactive routing is used for this purpose.

In ZHLS, each node contains an intrazone and interzone routing table to manage routing between nodes from a same zone and from different zones respectively. The update of these tables is performed, by sending two types of Link State Packets (LSPs), node LSP and zone LSP for intrazone and interzone, in that order.

A proposal to enhance the routing, by ZHLS is given in [Hamma et al., 2006], where the ZHLS Gateway Flooding (ZHLSGF) scheme is defined to reduce routing overheads and reduce routing tables' size. This modification is closely related with the nodes that act as a border between different zones, since they are responsible for calculating the shortest path between other gateway nodes, only sending interzone discovery packets between each other, thus avoiding unnecessary packet forwarding to other nodes within the zone.

2.2.3.2 Distributed Dynamic Routing

Another hierarchical hybrid routing protocol, the Distributed Dynamic Routing (DDR) algorithm [Nikaein et al., 2000], for mobile Ad-hoc networks, is a tree based routing protocol which consists of six different stages where an election of the preferred neighbour is made, followed by the forest construction which creates a suitable structure for the wireless network, allowing an improved resource utilisation. Afterwards intra and inter tree clustering is performed, followed by zone naming and partitioning. Zones are responsible for maintaining the protocol scalable and reducing the delay.

While DDR creates and maintains a dynamic logical structure of the wireless network, the Hybrid Ad Hoc Routing Protocol (HARP) [Nikaein et al., 2001] finds and maintains routing paths. The HARP protocol aims at discovering the most suitable end-to-end path from a source to a destination by using a proactive intra-zone routing approach and a reactive inter-zone scheme, by performing an on demand path discovery and by maintaining it while necessary.

Even though the DDR algorithm does not require any sort of cluster-head for cluster maintenance, the possibility of some nodes being chosen as preferred neighbours by other nodes may lead to the creation of bottlenecks, as they would be required to transmit an increased amount of both routing and data packets. It is important that the choice of preferred neighbours is balanced so that the overall performance of the protocol does not get compromised. Moreover maintaining the entire logical structure of the network may be somewhat heavy, depending on how dynamic nodes may be [Kim and Lee, 2011].

2.2.3.3 Cluster Based Routing Protocol

Aiming at a scalable, loop free routing protocol with support for asymmetric links, the Cluster Based Routing Protocol (CBRP) [Jian et al., 1999] proposes a variation of the "Min-Id" [Gonzalez, 1993; Zhu et al., 2010] for cluster formation, in which the purpose is to create a hierarchy consisting of overlapping 2-hop-diameter clusters where a node is elected as cluster head, responsible for maintaining cluster membership information. By exploiting the cluster architecture, flooding traffic used in the routing process is minimized.

As a 2-level hierarchy, this protocol can be scalable to a certain extent, however, the typical cluster formation and cluster-head election overhead still exists. Even though node mobility does not necessarily lead to inaccurate routing table calculations, as it would happen with a purely proactive approach, the inherent route retrieval propagation delay may lead to temporary loops.

In the Routing Process, RREQ packets are flooded from source to destination, but only cluster head nodes are used in this process. When these packets reach the target, a RREP is sent back to the initiator node [Parvathi, 2012]. Even though this process is triggered by an on-demand request, additionally, every node within a cluster zone periodically exchanges with its neighbours routing table information by using *HELLO* packets. This pro-active behaviour in conjunction with the reactive on-demand requests positions the CBRP within the hybrid family of routing protocols.

In addition to the Routing process, the CBRP also defines two other major components which are Cluster Formation and Adjacent Cluster Discovery. The Cluster Formation process consists on the usage of a variation of the "lowest ID" clustering

Table 2.6: Comparison of Hierarchical Hybrid Routing Protocols

Protocol	Cluster-based	Scalability Techniques	Communication	Storage
CBRP	yes	2-level Hierarchy	$O(\frac{N}{Z}^2)$	$O(N)$
DSR	yes (zones)	Preferred Neighbours	$O(N^2)$	$O(N)$
ZHLS	yes (zones)	No Cluster-heads	$O(\frac{N}{Z}^2)$	$O(N)$

N : Total number of nodes
Z : Total number of zones or cluster-heads

algorithm where a set of rules for electing the cluster head are defined. Wrapping the whole protocol, the Adjacent Cluster Discovery process aims at discovering all bi-directionally linked adjacent nodes. The process is executed by broadcasting the summarised Cluster Adjacent Table of each cluster head as Cluster Adjacency Extension to the *HELLO* messages.

2.2.3.4 Hierarchical Hybrid Routing Protocols – Comparison

Hierarchical Hybrid routing protocols provide most of the existing advantages in the previously analysed protocols. Their mechanisms and complexity are presented in Table 2.6, revealing that in a worst case scenario these protocols have similar complexities. The tree-based DSR protocol has a higher communication complexity as it constructs its own forest of connected zones, therefore being more complete than other protocols.

2.3 Open Issues

Regarding routing techniques and different approaches for Mobile Ad-hoc Networks, there is already a vast amount of existing contributions. An important distinction between available routing protocols is how the network is organized. On one hand, hierarchical routing is expected to reduce overhead and improve resilience to mobility [Zhang and Chong, 2009], on the other hand, flat topologies are more simple and flexible [Nikaein et al., 2000].

From the performed analysis of the State of the Art, hierarchical reactive protocols stand out for being scarce. This fact is related with the difficulty in maintaining a hierarchy without periodically exchanging messages. Hierarchical Proactive routing protocols are more common and different approaches already exist. However, to the extent of our knowledge, there is still no hierarchical routing protocol that aggregates cluster information with different granularity levels, similarly to FSR in flat routing, being less prone to disruption when compared with other hierarchical protocols.

There is a lack of a routing concept more effective in supporting node mobility, capable of being oblivious to cluster specific changes that only affect the involved cluster. In addition to avoid changes in the routing tables of nearby clusters, a new routing concept should not relying on expensive on-demand RREQ between clusters, avoiding overhead and inherent delay of RREQ messages. Being able to operate despite the cluster organisation is also an important feature, as many protocols rely on complex clustering algorithms (e.g. fixed number of hops, cluster-heads, different wireless interfaces, among others). Not requiring a specific clustering algorithm would allow, for instance, having no need for the usage of cluster-heads in routing, reducing single failure points and making the entire network more flexible. Therefore, there is a need for a flexible routing protocol able to deal with any clustering scheme.

2.4 Summary

In the existing literature several routing protocols existing for routing in Mobile Adhoc Networks. However not all of these protocols provide a significant new approach for routing, being many times small extensions of the most relevant routing schemes. Moreover, many works rely on complex or even unrealistic assumptions which are not suitable for dynamic networks such as MANETs.

The presented State of the Art highlighted the contributions provided by routing protocols separated into different routing classes, taking also into account improvements made to and provided by these protocols. Nevertheless, several issues still exist, motivating the creation of new routing mechanisms for increasingly larger autonomous networks.

A comparison of the analysed protocols showed that reactive routing protocols are not necessarily more scalable in worst case scenarios where many flows exist. Moreover, it also demonstrated that cluster-based alternatives are able to maintain a smaller communication complexity. Even though the most scalable approach is provided by the DART protocol, regarding the communication complexity, the mechanisms necessary for this scalability to be achieved involve themselves additional overhead which is neither accounted as communication nor storage complexity.

A more global analysis of the routing protocols, one that takes into account most of the aspects related with routing in wireless Ad-hoc networks, must also be considered for instance by using simulation-based or real implementations of the existing protocols. Such an approach will allow a truthful analysis of a protocol's behaviour in challenging environments with node mobility and wireless interferences, validating its performance.

Chapter 3

Deferred Routing

The paramount importance of Scalable Routing in Wireless Multi-hop Networks has been stressed by many recent works in the area of Mobile Ad-hoc Networks (MANETs). In fact, as previously identified, users are expected to be surrounded by thousands of wireless capable devices in a near future [Cimmino and Donadio, 2009], connecting people to their everyday objects, jobs and hobbies.

All these devices and services motivate the need for ubiquitous wireless networks. Even though the number of deployed wireless access points has also increased with time, few are available and organized to support this demand. The concept of Wireless Community Network (WCN) can be seen as an alternative [Frangoudis et al., 2011], where several static Institute of Electrical and Electronics Engineers (IEEE) 802.11-enabled devices are interconnected in urban areas, increasing the connectivity of Wireless Fidelity (Wi-Fi) networks offered to nomadic users. Private hotspot owners play an important role in these networks, as they must share their Wireless Local Area Networks (WLANs) in order to increase coverage. However, without some sort of incentive for sharing private access points, the emergence of these networks is limited.

Mobile Ad-hoc Networks offer a more flexible and ubiquitous network infrastructure, as it consists of all the users interested in being part of such networks. Moreover, Ad-hoc networks can easily be configured to work in parallel with WCNs, creating additional coverage for WLANs. A known limitation of these networks is the scalability of its routing protocols. Since routing in MANETs needs to be resilient to mobility and handle several nodes that individually act as routers, there is a need to improve the performance of these protocols, for instance, in dense urban areas [Arora et al., 2012].

In this chapter, a new routing paradigm is presented in order to guarantee scalable routing in Clustered MANETs. The Deferred Routing (DefeR) scheme is introduced, as well as all the necessary mechanisms to propagate routing information, select the most suitable Gateways (GWs) and finally route and forward data packets.

3.1 Concept and Model Definition

Previous works addressing the topic of scalable multi-hop routing have relied on the usage of a clustered network organisation. By using a clustering protocol, these approaches are able to restrain the propagation of routing messages throughout the entire network and reduce the impact of node mobility within clusters (micro-mobility). However, a major drawback of these solutions is related with inter-cluster routing overhead and poor support of node mobility between different clusters (macro-mobility).

The DefeR approach consists on efficiently handling routing in clustered networks by defining a multiple view network hierarchy, achieved by aggregating clusters into different levels and by postponing routing decisions throughout traversed clusters until the final destination is reached.

This network organisation resembles the cartographic division of the world into continents, countries and cities, assigning identifiers with different granularities to each region. As an example, when travelling through different countries, people only consider their destination in a broader view, setting their goal to it and gradually focusing on their destination as they get closer. Another work with a similar approach, inspired by computational geometry techniques, is the Greedy Distributed Spanning Tree Routing protocol [Leong et al., 2006], which defines convex hull trees using nodes' absolute position information, which may not always be available, in order to optimize the routing process by sending packets to hulls which contain the desired destination's position.

In Deferred Routing, a tree organisation of clusters is considered and, instead of traversing the entire tree looking for the desired cluster destination, the search can be optimized to a complexity of $O(\lfloor \log_2(n+1) \rfloor)$, for n clusters, without using any geographical position information. Therefore, routes are established according to the cluster hierarchy, exploiting the different granularity levels of clusters within clusters. Moreover, the reliability of the links between clusters is taken into account, rather than minimizing the total hop count from source to destination.

One key advantage of using Deferred Routing is that, by keeping its optimised network hierarchy, it is able to limit not only the effects of micro but also macro-mobility, as clusters not involved in the mobility process of nodes are oblivious to changes in other clusters. Moreover, DefeR does not require additional routing messages

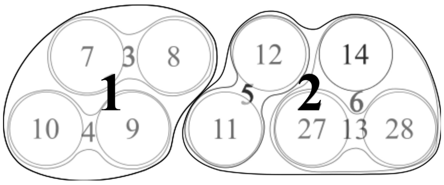


Figure 3.1: Contexts with embedded clusters

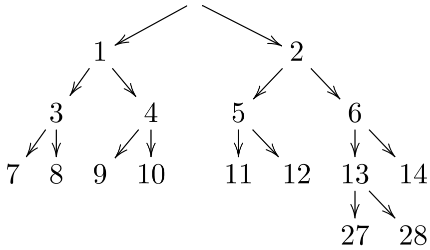


Figure 3.2: Hierarchical tree of clusters

for inter-cluster routing, being adaptable to any available link-state routing protocol with small changes to their own routing messages.

Multiple View Cluster Hierarchy

In order to understand the mechanisms behind DefeR, it is important to firstly be aware of how its hierarchy is defined. Similarly to an Autonomous System (AS) of the Border Gateway Protocol (BGP) in a more challenging environment, groups of nodes are created in wireless networks [Zhou et al., 2009]. These groups of nodes, or clusters, are presented in Figure 3.1. When considering mobile Ad-hoc networks, this organisation can be achieved by using a clustering algorithm such as the generalised max-min clustering algorithm [De Clauzade De Mazieux et al., 2007], where the management of the clusters and their identification is ensured by the DefeR routing scheme. In order to do so, each routing message used by the DefeR protocol includes a Cluster Identifier (CID). Due to mobility, whenever a node changes its cluster, the DefeR routing scheme will update the CID of that node and perform the required adjustments regarding the existing routing tables.

The hierarchy employed by DefeR is based on a binary tree structure, motivated by the bisection that occurs in growing clusters and also by the base-2 logarithmic complexity of balanced binary-search-trees, illustrated by Figure 3.2. Moreover, it also makes it simpler to compute the respective CID of each cluster. This organisation defines different level clusters paired with virtual identifiers for each cluster of nodes

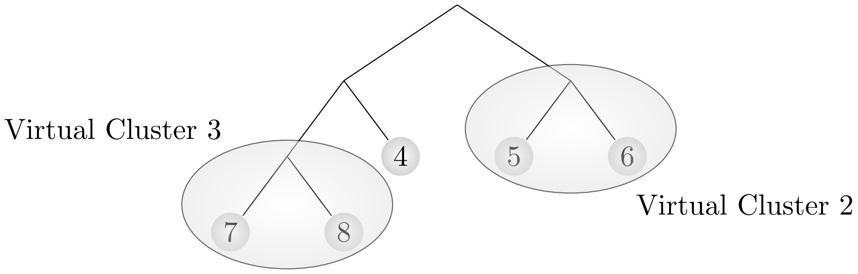


Figure 3.3: Network View by Cluster 4

which correspond to different granularity levels of knowledge. In this example, the higher level clusters, with the CID number 1 and 2, could correspond to two different neighbourhoods. The remaining CIDs represent buildings or common areas within the neighbourhood (CIDs 3, 4, 5 and 6), and finally the leaf clusters of the hierarchy correspond to actual clusters of nodes, where users share similar interests and closely interact (in this hierarchy: CIDs 7, 8, 9, 10, 11, 12, 14, 27 and 28).

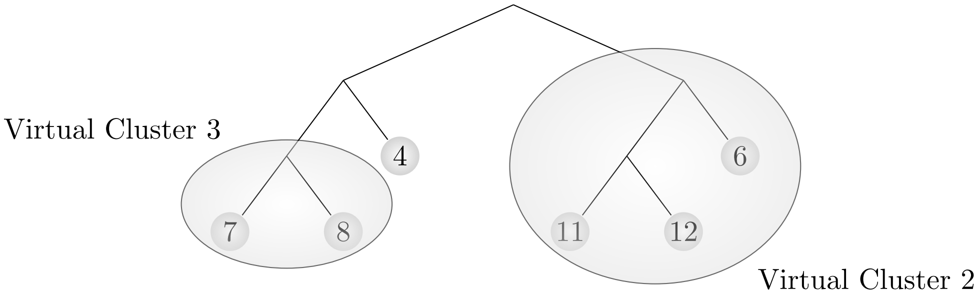
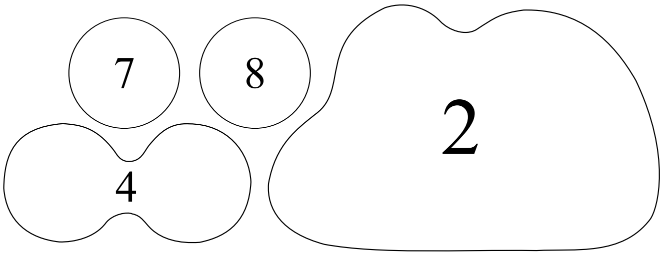


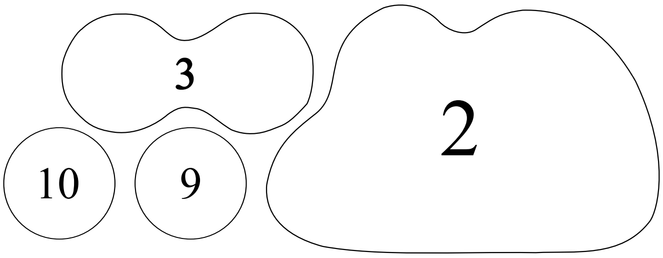
Figure 3.4: Network View by Cluster 4 (updated)

A different perspective on another possible hierarchy is depicted in Figure 3.3, showing how nodes in Cluster 4 perceive the network hierarchy. With this perspective, the routing information propagated within cluster 4 would include only clusters 2 and 3, ignoring completely the existence of clusters 7, 8, 5 and 6. Such aggregation of clusters allows them to change without causing any disruption. For instance, if the density of nodes in cluster 5 increases significantly, a bisection of the cluster, according to the numbering convention of clusters detailed in Appendix A – Algorithm 6, would originate clusters 11 and 12. However, in what concerns nodes in cluster 4, the network perspective would not be changed, as shown in Figure 3.4.

As previously mentioned, the hierarchy defined for DefeR establishes a relation between the virtual clusters and the real clusters, which represent the most detailed level of knowledge about an existing cluster (represented by the leaf clusters). While



(a) Network View for Clusters 7 and 8



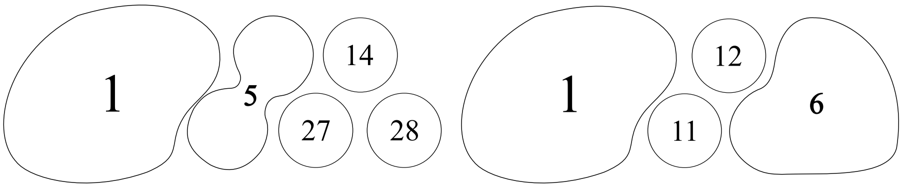
(b) Network View for Clusters 9 and 10

Figure 3.5: Additional Network Views

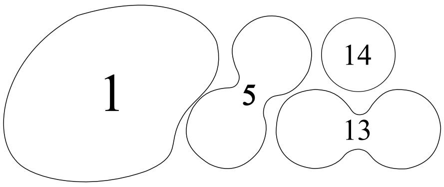
other hierarchies simply take into account existing clusters, the virtual aggregation of clusters allows the DefeR scheme to be more resilient to mobility phenomena, reducing the undesirable effects of micro and macro mobility, thus maintaining routing more scalable. As the movement is not expected to be entirely random [Cacciapuoti et al., 2012], this organisation will reduce the number of registered cluster changes.

A relevant contribution of the presented hierarchy is related with the different perception that nodes have of the entire network and all the existing clusters. In fact, the nodes' membership to each cluster provides them a different network perspective according to their hierarchical position. Figure 3.5a depicts the network organisation as it is perceived by clusters 7 and 8, following the hierarchy previously presented. As sibling clusters, 7 and 8 recognise each other but acknowledge only two other clusters: 4 and 2. As previously explained, clusters 4 and 2 are the result of an aggregated view of the network, being themselves virtual clusters. In a real scenario, clusters 7 and 8 could be for instance two groups of people within a building, whereas CID 4 would correspond to a next door edification, being cluster 2 another infrastructure nearby.

The aggregation is performed according to the hierarchical relationship between clusters, such that hierarchically closer clusters are less aggregated and further away clusters are progressively more aggregated. A similar aggregation level is obtained for clusters 9 and 10, as illustrated by figure 3.5b. By using broader parameters that bring



(a) Network View for Clusters 27 and 28 (b) Network View for Clusters 11 and 12



(c) Network View for Cluster 14

Figure 3.6: Additional Network Views

clusters together, these different granularity perspectives allow the desired organisation for the forwarding of packets through selected GWs. Moreover, as presented later in this work, the GW selection will take into account the reliability of each GW link, avoiding congestion in existing links. In this example, the given hierarchy reveals that clusters 9 and 10 are more likely to interact with clusters 7 and 8 (aggregated into CID 3), rather than with any other node in the remaining clusters.

Another noteworthy example of this aggregation scheme is presented in Figure 3.6a, which represents the view of clusters 27 and 28. These two clusters have an additional hierarchical level, which may have resulted from a more detailed cluster division due to an increasing number of nodes or separation of interests, resulting in an additional cluster in their view. Moreover, in this particular example, even though cluster 14 is the actual cluster, if one or more divisions were to occur, no changes would be noticed by clusters 27 and 28. The remaining views for clusters 11 and 12, as well as for cluster 14 are presented in figures 3.6b and 3.6c, respectively.

The perspective of nodes is generalised in Algorithm 1, where brother clusters are at the same level and directly connected by the same cluster, whereas ascendant clusters are clusters at higher levels from which a cluster was originated. Further details about this aspect are presented in Appendix A – Algorithm 7.

In addition to the already mentioned aspects of the DefeR Protocol Hierarchy, the most important characteristic is how it is able to cope with mobility phenomena and with changes in the clustered network. By using different network perspectives to each cluster, the addition or deletion of clusters, as well as the changes in nodes'

Algorithm 1 Hierarchical Cluster View

```
if received CID is from brother CID then
    processed CID equals received CID
else
    processed CID equals ascendant of received CID when brother of own ascendant
end if
```

cluster association, will only have an impact on hierarchical nearby clusters in which the changes occur.

3.2 Deferred Routing Mechanisms

Having introduced the main characteristics of the DefeR routing protocol, in particular regarding the optimisation of network views, efficient mechanisms for the propagation of routing information must also be defined. This section introduces the approach of packet overhearing in DefeR as well as the steps taken to robustly deliver data packets in a deferred fashion. Moreover, an introduction to the selection of GW nodes is also provided.

3.2.1 Reduced Routing Information Overhead

The DefeR protocol is an entirely proactive routing protocol, periodically exchanging its routing messages with the CID of the cluster to which the nodes belong, included in the header of sent routing messages. This allows routing messages received between neighbour clusters to be processed differently than routing messages from the same cluster.

Considering a link-state routing protocol such as Optimized Link State Routing (OLSR) [Clausen and Jacquet, 2003] for intra-cluster routing, the same routing messages (*HELLO* and Topology Control (*TC*)) are used, requiring only minor changes. No new messages are created and the existing messages are always contained within their originating clusters. Since no routing message is forwarded across different clusters, the most significant overhead introduced in the existing routing messages is a list of Internet Protocol (IP) addresses and their corresponding CIDs (IP-CID mapping).

Nodes on the border of clusters, or GW nodes, often receive routing messages from their neighbour clusters. However, these messages must not be forwarded nor considered for intra-cluster routing and should be discarded. Even though this is the typical approach taken by other cluster-based routing schemes, the DefeR protocol acts differently retaining any routing information in these messages that can be useful for inter-cluster routing (Context Connectivity Information). When a node receives routing information from a foreign cluster it automatically becomes a GW node, sharing its inter-cluster connectivity with the nodes within its cluster. Algorithm 2 explains these procedures, and they are explained in further detail in Appendix A – Algorithms 8, 9 and 10, where the complete specification steps are provided.

Algorithm 2 Sent and Received Routing Messages

```
procedure SEND_ROUTING_MESSAGE(message)
    Include CID in message header
    Include Link-state routing information present in original message
    Include Context Connectivity Information
    Include IP-CID mapping information
    Send message
end procedure

procedure RECEIVED_ROUTING_MESSAGE(message) //Same for HELLO and TC messages
    if received CID not equal to own CID then
        Set node as GW
        Process Context Connectivity Information and discard packet
    else
        Remove and process IP-CID mappings and Context Connectivity information
        Process remaining message for intra-cluster routing
    end if
end procedure
```

An additional improvement on the routing overhead is direct consequence of the used hierarchy. The information added by GW nodes announcing their connectivity is minimal, as it mainly corresponds to their Identifier (ID) and the CID to which they are connected to. Since clusters are aggregated, the number of existing connections is reduced and changes in the hierarchy do not necessarily issue changes in the sent routing information. The information about the association of each node to its cluster behaves in a similar way, avoiding unnecessary updates when nodes move between nearby clusters. This aspect is particularly important since nodes' associativity information is only propagated when changes occur. However, it is important to note that the IP-CID mapping included in routing messages could be completely removed if the clustering protocol was developed for DefeR. Such an approach was considered however, in order to guarantee a generic routing protocol capable of being used in any circumstance, the DefeR protocol was developed so that it can operate in any type of clustered network.

3.2.2 Route Establishment in Deferred Routing

Despite its robust hierarchy, a fundamental feature of the Deferred Routing Protocol is the ability to manage the routing process between the virtual and real clusters, ensuring scalable routing between source and destination nodes. Similarly to other routing protocols, the path establishment within clusters is performed by a link-state routing protocol, such as the OLSR protocol which will be considered from this point forward. However, this alone is not sufficient and additional procedures have to be guaranteed so that packets are correctly forwarded between different clusters, as no additional protocol is used.

The DefeR approach does not require additional routing messages for inter-cluster routing and limits its overhead by inserting Gateway Information in a link-state's specific routing messages, for instance OLSR. Moreover, the presented routing approach maintains a mapping of each node's cluster association, propagating this information in existing routing messages only when changes occur.

As previously presented, the creation of Gateway Information occurs only when a node in the vicinities of a neighbour cluster receives routing messages from other clusters, becoming itself a GW node. While routing messages from foreign clusters are typically discarded, the DefeR Protocol uses these foreign messages to overhear the network topology information as perceived by other clusters. A foreign message is processed when a routing message with different CID is received. This process, which extracts foreign information for use inside the GW's own cluster, is described in more detail later in this section.

After processing this information, border nodes, or GWs, acquire their specific network perspective, as well as their own reliability as GWs, announcing it to every node within the cluster they belong to. This aspect results from the link quality of the GW node, determined by using Kernel Estimators. An additional feature of the used GW nodes is the store-and-forward capability [Nordemann and Tonjes, 2012]. This allows gateways to temporarily store data packets with a maximum delay constraint D, when for some reason a broken link is detected, or when a cluster changes and routes are re-adjusted. By using this mechanism, less packets are lost and the healing process of previous routes is automatically triggered as packets are forwarded.

When defining a routing path, a source node's main concern is to identify where the destination node can be found, taking into account the node's perspective to what concerns their own CID. Assuming that the destination node is within the same cluster as the source node, the shortest path is already known according to the routing table

Algorithm 3 Inter-Cluster Routing

```
procedure DATA_PACKET_HANDLE(packet)

    Retrieve the Packet's destination
    Determine destination's Context ID
    if Context ID not found then
        //Invalid destination
        return
    else
        Find most suitable GW for the destination's CID
        if GW not found then
            //There is currently no connectivity
            Resend packet later
            return
        end if

        Find next-hop towards GW
        Send packet to next-hop

    end if

end procedure
```

defined by the used link-state protocol. However, when a destination is found in a different cluster, the next task of the source node is to find the most suitable GW node. By analysing the provided information by each GW node in each cluster, and its position in the hierarchical tree, the source node will choose the path with less cluster-hops, forwarding packets to it. The processing of a new data packet received by any node that is not the final destination is presented in Algorithm 3.

As previously presented, nodes within a cluster only perceive the network's clusters to a certain extent. This network perspective allows a very straightforward routing decision, which aims at reducing routing complexity, maintaining scalability. However, as nodes choose the shortest path taking into account cluster-hops, the total number of hops may be penalized not only by choosing longer paths, but also by opting for routing stability. This happens because the GW selection process thrives on choosing the GW with the best link-quality, avoiding nodes under congestion or with unreliable links due to mobility or interference, allowing load-balancing between the existing resources.

Even though the total number of hops achieved by DefeR may not always be the lowest possible, as packets travel through clusters, their proximity to the destination cluster unveils a more precise network view, thus shortest paths are more likely to be established near the end. In addition to this, mobility phenomena which might render previously calculated paths impractical, are transparent throughout the packet forwarding amongst different clusters. This straightforward approach allows the Deferred Routing scheme to automatically repair routing paths such that an outdated routing decision does not result in a packet drop. Hence, with this self-healing characteristic, whenever a packet is incorrectly forwarded, the receiving nodes will re-forward it into the correct path as long as it exists.

3.2.3 Efficient Gateway Selection

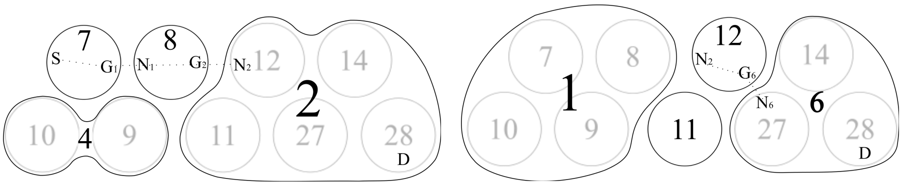
As previously mentioned, Gateway nodes are responsible for connecting different clusters. Through the analysis per link of the time interval between each $HELLO$ message received from a node in a foreign cluster, a parameter for Link Quality is obtained using Kernel Estimators, presented in the next chapter. In the process of choosing the most suitable GWs, the DefeR protocol uses a metric that takes into account not only the current link quality, but also the stored past link quality history, weighing both in order to provide the best possible results. In equation 3.1 the Link Quality Metric (LQM) is defined, where $w_{current}$ and w_{past} represent the weight for the current and past link quality respectively, with the following restriction $w_{current} + w_{past} = 1.0$.

$$LQM(x) = w_{current} \times \widehat{m}(x; 1, h) + w_{past} \times \widehat{m}_{t-1} \tag{3.1}$$

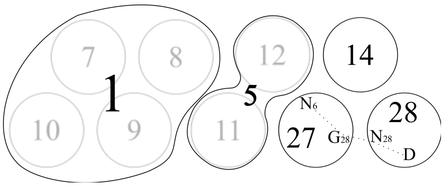
Since this metric analyses the link quality between two nodes in real-time, an efficient GW selection can be achieved. Whenever a GW node is under a significant amount of traffic load, its link quality will decrease and thus an alternative GW, if existing, will be selected. Link disconnections due to mobility are another external influence that can be predicted by this metric, since a departing link will progressively lose its quality, leading to the selection of another GW node. By using this metric with a well defined cluster hierarchy and low routing overhead, the DefeR protocol is able to efficiently forward its packets throughout the network until the final destination is reached.

3.3 Deferred Routing Examples

The concept behind the used hierarchy and the aggregation of cluster views was strongly influenced by the community aspect that usually brings people together. This routing approach takes into account the increased interaction between users within a same context, regardless of the used mobility pattern, using a well defined network hierarchy



(a) Forwarding steps 1 to 4 (b) Forwarding steps 5 and 6



(c) Forwarding steps 7 and 8

Figure 3.7: Forwarding Process – Path Taken

of real and virtual clusters. In fact, it has been proved by comparing real interactions with existing mobility models, that proximity-based interactions are little influenced by the specifics of a mobility pattern and that contact patterns should be considered instead [Panisson et al., 2011].

Regarding the probability of users maintaining more frequent interactions with the same users, or users in the same context, an analysis of the "Infectious SocioPatterns dynamic contact networks" dataset, available from the experimental SocioPatterns[1] framework [Isella et al., 2011], was performed. In this dataset the proximity information of participants of an event has been registered, revealing that repeated interactions between the same users are in fact highly probable. Considering the most frequent interactions per user, it is possible to conclude that on average, 40% of the total interactions occur with the same user.

In order to better illustrate the presented routing protocol, an example of how the routing process occurs is presented next. While having a wireless network organized according to the expected node interactions will allow routing protocols to perform more efficiently, less common interactions between different clusters must also be handled. For instance, referring back to the network hierarchy presented in figure 3.2, the worst case scenario would occur with packets being sent from a source node S within cluster 7 to a destination node D in cluster 28. Even though such a social interaction is not expected to be common, it might occur and the packet forwarding by DefeR is depicted in figure 3.7. The presented GW and path choices are merely illustrative and, despite

---

[1] www.sociopatterns.org

Algorithm 4 Packet Forwarding Step by Step

```
In Node: Source $S$
    Identify the destination's cluster (Cluster 2)
    Forward the packets to the most suitable Gateway (2 hops)
In Node: Gateway $G_1$
    Identify the destination's cluster (Cluster 2)
    Forward the packets to the most reliable neighbour node in the following cluster
In Node: $N_1$
    Identify the destination's cluster (Cluster 2)
    Forward the packets to the most suitable Gateway (1 hop)
In Node: Gateway $G_2$
    Identify the destination's cluster (Cluster 2)
    Forward the packets to the most reliable neighbour node in the destination cluster (direct connectivity)
In Node: $N_2$
    Identify the destination's cluster (Cluster 6)
    Forward the packets to the most suitable Gateway (1 hop)
In Node: Gateway $G_6$
    Identify the destination's cluster (Cluster 6)
    Forward the packets to the most reliable neighbour node in the destination cluster (direct connectivity)
In Node: $N_6$
    Identify the destination's cluster (Cluster 28)
    Forward the packets to the most suitable Gateway (1 hop)
In Node: Gateway $G_{28}$
    Identify the destination's cluster (Cluster 28)
    Forward the packets to the most reliable neighbour node in the destination cluster (direct connectivity)
In Node: $N_{28}$
    Link-state routing procedure
```

aiming at minimizing the number of cluster hops, different paths may exist and will be chosen according to the existing load in each link. All the steps taken by the routing protocol are presented in Algorithm 4.

In the previous example, the destination node D is expected to remain static in cluster 28. However, if this node moved itself to a nearby cluster, only small parts of the forwarding procedure would have to be changed. This transparent way of dealing with mobility results from the usage of virtual clusters which enables a progressive or deferred routing discovery. For instance, assuming that the destination node moves to cluster 27, all the previous steps would be kept unchanged until step 13, which would then be the final step, as shown in Algorithm 5. Moreover, if packets are sent to the destination during the update of routing tables, no problem will be raised as nodes will automatically re-direct the packets to the new destination's cluster.

Algorithm 5 Updated Packet Forwarding

```
In Node: $N_6$
13: Link-state routing procedure
```

The reason for all the previous steps, before step 13, to remain the same is that no change is detected by other clusters. As far as it is perceived by nodes in different clusters other than 27 and 28, the destination node D has always been either in cluster 2, cluster 6 or cluster 13, depending on the hierarchical position of the observing clusters.

In the described packet forwarding scheme, the GW nodes always have to identify the current destination's (D) Cluster. This is required since a GW node may be a GW to several clusters at the same time and it needs to choose the appropriate one. Another important aspect of packet forwarding is the choice of the most suitable GWs. This parameter results directly from the reliability of the link between the nodes in neighbour clusters and the GW.

3.4 Complexity Analysis

Scalable routing in Mobile Ad-hoc Networks has already been addressed by several other works. These propose different schemes that, being more or less effective, introduce an increased complexity and overhead to the routing protocols. The DefeR approach is successful not only in reducing the complexity of the routing procedures, but also in requiring a small amount of resources without introducing assumptions on hardware or other functionalities.

Typically, routing approaches that handle clustered networks, as DefeR does, immediately reduce the complexity and overhead of the routing protocol within these clusters. However, the management of routing information between different clusters is more challenging and the used inter-cluster routing approaches often have problems of their own [Pang and Qin, 2006].

If the OLSR protocol is compared against the Cluster-based OLSR (C-OLSR) protocol and with DefeR, assuming that it also uses OLSR for intra-cluster routing, an improvement on scalability will be registered in the latter two. However, the C-OLSR protocol still has scalability issues, in particular when the cluster structure is changed.

In link-state routing protocols, the forwarding of routing messages is responsible for most of the control traffic overhead. Bearing this in mind, it is important to analyse the impact of the number of TC messages forwarded by the OLSR based protocols, where a lower number of forwards will reflect the protocol's scalability. Therefore, in order to compare the scalability of the three referred protocols, a wireless network shall be represented by using a Poisson Point Process over the plan betoken by S and with intensity γ.

In this network, assuming that the number of nodes N, follows a Poisson Law of intensity $\gamma \times S$, the total number of nodes per unit of area M, is represented by γ ($M = \gamma$). This network layout ensures that each node has on average M neighbour nodes and thus the radius of the network will be $\sqrt{N/M}$, since in a K-hop neighbourhood the number of nodes in a disk radius K is on average K^2M.

The number of forwarded TCs depends on the number of Multipoint Relay (MPR) nodes in a K-hop neighbourhood, since the remaining nodes never forward them. As demonstrated by Adjih et al. [Adjih et al., 2004] and Jacquet et al. [Jacquet et al., 2002], the average number of MPRs selected by a node (M_{MPR}) is defined by Equation 3.2 and further that for an increasingly large number of neighbour nodes ($M \to \infty$), M_{MPR} is represented by Equation 3.3.

$$M_{MPR} \leq \sqrt[3]{9\pi^3 M} \tag{3.2}$$

$$M_{MPR} \sim \beta\sqrt[3]{M} \wedge \beta \approx 5 \tag{3.3}$$

Taking into account the average number of MPRs selected by a node, it follows that the probability of a node to be an MPR is M_{MPR}/M [Canourgues et al., 2008]. Since the number of TC retransmissions corresponds to the number of MPRs times the number of nodes in a K-hop network, the average number of retransmissions is defined in Equation 3.4. Furthermore, the number of nodes that may retransmit a TC message, at precisely K hops of a TC transmitting node, is on average defined by Equation 3.5.

$$TC_{retrans} = \frac{M_{MPR}}{M} \times K^2M = M_{MPR}K^2 \tag{3.4}$$

$$Khop\ TC_{retrans} = \frac{M_{MPR}}{M} \times (K^2 - (K-1)^2)M = M_{MPR}(K^2 - (K-1)^2) \tag{3.5}$$

The previous equations assume an un-clustered network where OLSR is used for routing purposes. However, despite using OLSR for intra-cluster routing, in a clustered network with C clusters the radius of the network will be $\sqrt{N/(M \times C)}$. In fact, the entire network can be considered as C distinct Poisson Point Processes, as DefeR does not forward messages across different clusters. Therefore, the number of retransmissions for the DefeR approach is also defined by Equation 3.4.

In contrast with DefeR, even though other cluster based protocols using OLSR (such as C-OLSR), have a similar perception of the network by grouping nodes, they require routing specific messages to be sent across different clusters. However, in the distributed version of the C-OLSR protocol, *TC* messages may be forwarded among different clusters, such that, for the cluster-based radius, the average number of nodes transmitting a *TC* message is defined by Equation 3.6. Moreover, this protocol uses its own specific messages for maintaining the cluster structure, Cluster Topology Control (C-*TC*) and Cluster *HELLO* (C-*HELLO*) messages, thus having more overhead.

$$C_{OLSR}\ TC_{retrans} = (C-1) \times M_{MPR}(K^2 - (K-1)^2) \tag{3.6}$$

Despite the theoretical performance expected by each protocol, the MPR selection process is NP-Complete [Jacquet et al., 2002] and therefore the actual number of MPR nodes may vary, possibly resulting in additional forwards that will have more impact in less scalable protocols. However, for an identical network organisation, the DefeR scheme forwards less *TC* messages than other cluster-based protocols.

Regarding the overall communication and storage complexity of the DefeR protocol, it is smaller when compared with a typical link-state routing protocol. Considering that each cluster has k_{max} nodes, the number of clusters c, in a network of n nodes, is represented by equation 3.7.

$$c = \left\lceil \frac{n}{k_{max}} \right\rceil \tag{3.7}$$

Knowing the communication complexity of a typical link-state protocol, $O(n^2)$, in DefeR a constant Communication Complexity (CC) value is expected to be achieved (as shown in equation 3.8), since the link-state protocol is restricted inside each cluster. This result can still be further improved if Multipoint Relay nodes are considered.

$$CC = O\left(\frac{n}{c}^2\right) \Leftrightarrow CC = O\left(\frac{n}{\frac{n}{k_{max}}}^2\right) \Leftrightarrow CC = O(k_{max}^2) \tag{3.8}$$

The overhearing approach which avoids DefeR from sending messages throughout all the nodes in the network, allows it to be extremely scalable. Moreover, regarding the storage complexity it does not require more resources than any other routing protocol, maintaining only an entry for each node in each cluster with a complexity of $O(n)$.

3.5 Summary

The definition of a scalable routing scheme for MANETs involves different aspects such as an efficient network organisation, resilience to mobility and robust path establishment. In this chapter, the Deferred Routing Protocol is presented and its mechanisms described. By reducing the number of sent and forwarded routing messages, this protocol is able to guarantee scalable routing while reducing the number of required routing table changes, resorting to the aggregation of network views which masks the movement of nodes across clusters. Moreover, by deferring complete routing decisions to nodes closer to the destination's cluster, less routing information is required and paths are more easily adaptable to changes.

A theoretical analysis of DefeR protocol's complexity was also provided, showing that it requires less routing messages to be forwarded across the network. This results from the protocol's overhearing property in conjunction with a well defined hierarchy.

Resulting from the definition of the Deferred Routing approach an article entitled "Onto scalable Ad-hoc networks: Deferred Routing", 2012, has been published in Elsevier's Journal of Computer Communications, as well as an article entitled "DASH, Deferred aggregated routing for scalable ad-hoc networks" published in the proceedings of the Wireless Days conference.

Chapter 4

Gateway Selection in Deferred Routing

Deferred Routing, as any other routing protocol, relies on the quality of wireless links between nodes participating in the network. However, Gateway nodes have an important role in the Deferred Routing (DefeR) scheme as they are responsible for the forwarding of packets along different clusters. Moreover, since nodes are expected to be mobile, the borders of existing clusters may constantly change and Gateways (GWs) may only be so occasionally. Consequently, finding the most stable GW nodes and their link quality to other clusters is a prevalent task in DefeR.

4.1 Link Quality Estimation

Many works have been proposed for the creation of multi-hop wireless networks with different routing protocols. Despite providing some insights on how to handle these networks and find communication paths between different devices, the existing protocols usually disregard the environment behind wireless communication, ignoring, for instance, link quality or even the network's load. With the purpose of solving this issue, link metric extensions such as the Expected Transmission Time (ETT) among many others [Bezahaf et al., 2012; Borges et al., 2011], have been added to these protocols. Furthermore, works more focused on modelling the wireless link specificities have been proposed, depending on several assumptions in order to correctly operate in this environment. However, many of these works' assumptions render them unrealistic in scenarios such as search and rescue, where information about mobility or position awareness is not available.

Existing Approaches

The estimation of link quality and its availability in wireless networks is an important feature to consider in route establishment and particularly relevant in mobile Ad-hoc networks, where link quality variations are frequent. Several authors have proposed different approaches concerning the analysis of link availability.

The main drawback of most of the existing works is either the usage of unreliable parameters, which are prone to errors and variations, such as signal strength and available energy, or the requirement of unrealistic or complex assumptions such as positioning knowledge (for instance Global Positioning System (GPS) coordinates), specific mobility models and characteristics (i.e. constant speed, known direction, known epochs), among many others parameters. However, a relevant contribution from these works is the proposal of new routing protocols and thorough formal models which allow a better comprehension of link related aspects.

Regarding the analysis of link quality, taking into account mobility aspects, Yu et al. [Yu et al., 2007] rely on the assumption that nodes are able to assess their own mobility parameters. For instance, knowledge of the nodes' average speed, pause time, direction and epoch time is necessary for predicting quality degradation, as well as the assumption of perfectly symmetric links. Moreover, this work's conclusions depend on the used mobility model, which must be the Random Waypoint (RWP) Model with particular specificities such as independent and identically distributed (*i.i.d.*) speeds, epochs and directions.

Link Lifetime (LLT) estimation can be an extremely important feature to consider in route establishment. This aspect is more relevant in network scenarios that consider mobility where link breakages are frequent. The work presented by Huang and Bai [Huang and Bai, 2008] suggests an approach which uses a Markov Chain Model to determine the availability of a link between two nodes, by describing the relative movement of both, knowing the initial distance between them. A comparison of the proposed model with previous works shows that the Markov Chain model outperforms other approaches that use the Rayleigh model [Qin et al., 2005] to predict node distribution, being able to increase stability in the construction of clustered networks [Bai et al., 2009]. However, this work relies on assumptions such as the knowledge of the distance between two nodes (either by using GPS or by analysing signal strength) or even assumptions on link characteristics considering them always bidirectional within a distance of R meters and not considering radio irregularity [Bai and Gong, 2010]. Additionally, assumptions on the mobility model, the Random Walk Mobility (RWM)

Model, are also required, such as a uniform distribution of speed and direction, as well as the same mean epoch length for each node.

A two-state Markov Chain Model is also proposed by Wu et al. [Wu et al., 2009] for the evaluation of a Single-Node Link Lifetime (S-LLT) using the Random Direction Mobility (RDM) Model and assuming that the time duration of each epoch is denoted by a random variable that is exponentially distributed with a known parameter λ_m. Assumptions regarding bidirectional links, known mobility direction and speed are also taken, both being uniformly distributed between $[0, 2\pi]$ and $[v_{min}, v_{max}]$.

Link availability is a parameter often considered as a suitable metric for increasing routing performance in Mobile Ad-hoc Networks. By assuming knowledge about nodes' direction and position while considering a constant speed within a Time Period (T_p) and independent movements, Jiang et al. [Jiang et al., 2005] propose a link availability quantity estimation. This estimation is achieved by exploiting the instantly available velocity, reflecting the dynamic nature of the link status. The authors also propose a T_p estimation, based on a mobility model that follows the assumption that terminal mobility is uncorrelated and that epochs are exponentially distributed with a known mean. In addition to the mobility related assumptions, all nodes are expected to know their positions by using GPS devices.

Shu and Li [Shu and Li, 2007] consider node speed in a wireless network as being responsible for link failure and therefore poor network performance. A link quality estimation is achieved by using a simplified version of the RWP mobility model, where nodes move in an arbitrary large area with no obstacles (i.e. no boundaries). It is also assumed that all nodes constantly move at the same speed with no pause times and, similarly to other works, that every link is bidirectional within a radius of R meters.

Another work regarding link evaluation is presented by Manoj et al. [Manoj et al., 2001], which estimates link lifetime by using a simple linear regression for path choice in a reactive routing protocol. However, this work relies on the used propagation model and link specificities such as the transmit power, channel and frequency, in order to obtain node positioning knowledge, which is required for the performed link estimates.

The work presented by Zhang et al. [Zhang et al., 2010] has a slightly different perspective trying to consider LLT determination taking into account the energy drain rate and relative mobility estimation of wireless nodes. The presented estimation of route lifetime relies on the assumptions of no energy limitations in any of the observed nodes and of nodes moving in the same direction at a constant speed considering a short enough period of time. Moreover, it requires node positioning awareness, either by using GPS or by assuming that transmitted packets are sent with the same power

Table 4.1: Wireless Link Modelling

Existing Approaches	Assumptions on Mobility	Link	Positioning	Validation	Formal Model
Yu et al.	✔	✔	✗	✔	✗
Huang and Bai	✔	✔	✔	✗	✔
Wu et al.	✔	✔	✔	✔	✔
Jiang et al.	✔	✔	✔	✔	✔
Shu and Li	✔	✔	✗	✗	✔
Manoj et al.	✗	✔	✔	✗	✔
Zhang et al.	✔	✔	✔	✔	✗

level as perceived by the receiver, which then can apply the radio propagation model for distance calculation.

A comparison summary of the presented link modelling works is shown in Table 4.1, using as comparison parameters the assumptions of each proposal - which can be related with the mobility model, link characteristics or positioning knowledge. The accuracy of the used validation scheme and contribution with a correct formal model are also taken into account. This table confirms that most of the existing models have strong and unrealistic assumptions and, in addition, some fail to provide a proper validation and formal model.

Not directly related with link quality evaluation, but closely concerned with mobile network modelling, an innovative work is presented by Saeed et al. [Saeed et al., 2008] which consists on using Neuro-Fuzzy Modeling and Neural Network Modeling for analysing the behaviour of different routing protocols, namely the Dynamic Source Routing (DSR) [Johnson et al., 2007], the Ad hoc On-demand Distance Vector (AODV) [Perkins et al., 2003a] and the Optimized Link State Routing (OLSR) [Clausen and Jacquet, 2003] protocols in Mobile Ad-hoc Networks (MANETs) with variable and attainable parameters, such as the number of nodes or mobility. This work provides Empirical Equation Models by analysing quantitative data using polynomial and multiple linear regressions. The authors use the network's context (node number and mobility) as inputs in conjunction with the network's performance (delay, routing delivery rate, routing packets delivery rate and routing load) as outputs for modelling. The modelled results are obtained by using simulation data for each parameter and routing protocol, showing the main differences between the empirical equations, neural networks and neuro-fuzzy models. Despite not presenting validation results compared against the presented models, the proposed models (Neural Networks and Neuro-Fuzzy) are both efficient and accurate in representing wireless networks' features without requiring any simplifications of their complex and dynamic aspects. The main disadvantage of using these methods is that they require previous training.

In the following section a new link quality estimator is presented, using kernel methods in conjunction with existing information obtained from a typical routing protocol. This scheme provides a feasible and robust alternative to the existing models.

4.2 An Accurate Model for Link Quality Estimation

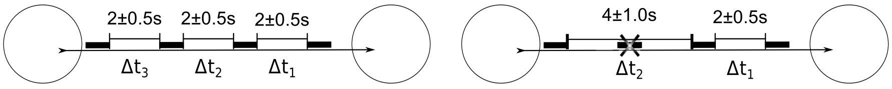
Taking into account the considered assumptions and research questions, local polynomial Kernel Estimators were used for the determination of link quality. In fact, by simply analysing the time interval Δt reception of periodical routing messages, an accurate model for link quality estimation is derived.

Kernel Estimators are applied by Kushki et al. [Kushki et al., 2007] for positioning purposes in Wireless Local Area Networks, by creating "fingerprints" using the Received Signal Strength (RSS). The results presented show that Kernel Regression is an efficient solution for such scenarios, thus motivating further usage of Kernel Methods in wireless modelling. Considering Link Quality, Kernel Methods will allow, the determination of a link quality model estimator by using existing routing or signalling messages. The purpose is to analyse the interval between these periodically received messages and, based upon them, estimate the quality of the used wireless link. These periodic messages can be obtained for instance from the routing protocol or from Layer-2 messages, such as beacons.

Focusing in particular on the OLSR protocol, it periodically sends *HELLO* messages with an interval of $2 \pm d, d \in X \sim U(0, 0.5)$ seconds, width d being an added delay following a uniform distribution between 0 and 0.5 seconds. These messages are sent so that new links and lost links are regularly detected. The random factor is added in order to try to avoid nodes from sending routing packets at the same time, which would cause several collisions in the wireless medium. The expected average interval between *HELLO* messages in a perfect connection would be exactly $E(X = \widehat{\Delta t}) = 2s$. However, since packet collisions and interferences exist, errors may occur, resulting in lost packets. Thus, throughout this work the Quality of a Link will depend on the number of lost packets between two received *HELLO* messages, such that a link without packet losses has perfect link quality. The link quality is defined by equation 4.1.

$$LinkQuality_{\Delta t} = \frac{1}{1 + \text{packets lost}} \ . \tag{4.1}$$

The time interval between a packet being sent and received depends not only on propagation characteristics, but also on the number of required packets sent until one is properly received, as depicted in Figure 4.1. Figure 4.1a represents a link quality of 100% for Δt_1, Δt_2 and Δt_3, while in Figure 4.1b, Δt_1 has a link quality of 100% and in Δt_2 the link quality is only of 50%. These errors are more prone to occur when a poor link quality is registered. Thus, by measuring the interval between consecutive *HELLO* messages, an estimation of the link quality can be retrieved using Kernel Regression Estimation.



(a) No Lost Packets (b) Packet Loss

Figure 4.1: Periodic Routing Message Exchange

As previously mentioned, the estimators used in this work are from the class of kernel-type regression, which allows the estimation of a least-squared weighted regression function $\hat{m}(x; p, h)$, that "locally" fits a pth degree polynomial, for a given data set (x, y) [Wand and Jones, 1994], where h is the smoothing or bandwidth parameter. Kernel methods and, in particular, kernel regression methods are also called *memory-based methods* because they require keeping or storing the entire training set to estimate or compute future data points. Indeed, these methods fit a model separately at each data point x_i and only data points close to x_i are used to fit the above mentioned model. This fitting process is such that the resulting estimated function is smooth in $\Re$.

Other regression functions related with Kernel Regression are the K-Nearest Neighbour (KNN) classification, State Vector Machines (SVM), Neuro-Fuzzy Models and Radial Basis Functions (RBF), which may not be so robust. For instance, on the classification of RSS based fingerprints, Kushki et al. [Kushki et al., 2007] do not consider the KNN approach as it presents a poor performance when training vectors that are nonconvex and multimodal. Also, previously used SVMs and RBFs have shown no resilience in scenarios with highly dynamic wireless settings, where MANETs should be included.

Being m the true regression function of the real link quality observed, the random regression model can be written as $m(x) = E(Q|X = x)$, representing the conditional expectation of variable Q relative to a variable X. From this point forward, q will correspond to the link's quality observed between two nodes and x will be the Δt between measured routing messages (Q and X will be the estimated values).

The Kernel function K is a non-negative real-valued integrable function, defined to be smooth with a maximum at 0 and with the following constraint:

$$\int_{-\infty}^{+\infty} K(x)\,\mathrm{d}x = 1 \text{ and } K(-x) = K(x), \forall x \in \Re \ . \tag{4.2}$$

Two commonly used Kernels are the Epanechnikov Kernel and the Gaussian Kernel [Wand and Jones, 1994], presented in equations 4.3 and 4.4 respectively.

$$K_h(x_i - x) = \frac{3}{4}\left(1 - \frac{|x_i - x|}{h}^2\right)_{\left\{\frac{|x_i - x|}{h} \leq 1\right\}} \ . \tag{4.3}$$

$$K_h(x_i - x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\frac{|x_i - x|}{h}^2} \ . \tag{4.4}$$

As previously mentioned, the Kernel Regression fitting depends on a smoothing parameter h, usually referred to as bandwidth. The choice of a correct bandwidth is extremely important to prevent under or over fitted estimations. A bandwidth selector, as defined by Wand and Jones [Wand and Jones, 1994], is a method that uses the data $X_1, \ldots, X_n$ to produce a bandwidth $\hat{h}$. Typically, bandwidth selectors are divided in two different classes – the *quick and simple* ones, which provide an acceptable bandwidth value without any mathematical guarantees, thus being disregarded in this work. Other selectors that are computationally more complex and aim at finding optimal bandwidth values, belong to a different class referred by Wand and Jones [Wand and Jones, 1994] as *hi-tech*, presented later.

Using Kernel Regression, at point x, the estimator $\widehat{m}(x; p, h)$ is obtained through the fitting of $\beta_0 + \beta_1(\cdot - x) + \ldots + \beta_p(\cdot - x)^p$ to (x_i, Q_i) using the least squares with $K_h(x_i - x)$ such that $\hat{\beta} = (\hat{\beta}_0, \ldots, \hat{\beta}_p)^T$ minimises

$$\sum_{i=1}^{n} \{Q_i - \beta_0 - \ldots - \beta_p(x_i - x)^p\}^2 K_h(x_i - x) \ . \tag{4.5}$$

Considering $\hat{\beta} = (X_x^T W_x X_x)^{-1} X_x^T W_x Q$, as defined by Wand and Jones [Wand and Jones, 1994], it is the solution obtained by the standard weighted least squares theory, assuming that $X_x^T W_x X_x$ is invertible, where $Q = (Q_1, \ldots, Q_n)^T$ is the vector of responses,

$$X_x = \begin{bmatrix} 1 & x_1 - x & \ldots & (x_1 - x)^p \\ \vdots & \ldots & \ddots & \vdots \\ 1 & x_n - x & \ldots & (x_n - x)^p \end{bmatrix} \ . \tag{4.6}$$

is a $n \times (p+1)$ matrix and $W_x = diag\{K_h(x_1 - x), \ldots, K_h(x_n - x)\}$ is a $n \times n$ diagonal matrix of weights.

In this work a 1^{st} degree polynomial estimation was used, thus the local linear estimator $\widehat{m}(x; 1, h)$ is defined by the following equation:

$$n^{-1} \sum_{i=1}^{n} \frac{\{\hat{s_2}(x;h) - \hat{s_1}(x;h)(x_i - x)\} K_h(x_i - x) Q_i}{\hat{s_2}(x;h)\hat{s_0}(x;h) - \hat{s_1}(x;h)^2} \ .$$

where:

$$\hat{s_r}(x;h) = n^{-1} \sum_{i=1}^{n} (x_i - x)^r K_h(x_i - x) \ . \tag{4.7}$$

In order to guarantee the quality of the obtained Kernels, some of the most commonly used optimality *hi-tech* criteria for selecting a bandwidth matrix will be considered. These are for instance the Mean Integrated Squared Error (MISE) and the Averaged Squared Error (ASE), which is used in this work.

$$ASE(h) = \widehat{m}_h = \frac{1}{n} \sum_{j=1}^{n} \{\widehat{m}_h(X_j) - m(X_j)\}^2 w(X_j) \ . \tag{4.8}$$

The ASE is a discrete approximation of the Integrated Squared Error (ISE), which has been shown by [Marron and Härdle, 1986] to lead asymptotically to the same level of smoothing as the ISE and MISE. Thus, without significant loss of performance and knowing that it is the easiest to calculate and handle [Härdle et al., 2004], the ASE is clearly an appropriate bandwidth selector.

4.3 Model Parametrisation

Having defined the main theoretical aspects of the proposed Link Quality Estimator Model, this section aims at presenting the necessary steps to obtain traces generated by both resorting to simulation and realistic traces. These traces allow the definition of distinct estimators, which provide not only the estimation of a wireless link quality, but also a comparison between real and simulation based wireless links. Moreover, an

Table 4.2: Obtained Kernels (simulation)

	Epanechnikov	Gaussian
Bandwidth	0.4000202	0.1157357
ASE	6.165×10^{-5}	5.981997×10^{-5}

implementation of the obtained Kernel Estimator is also presented, using the OPNET Modeler Wireless Suite® [OPNET Technologies, 2012] network simulator with the purpose of validating the estimated link quality values.

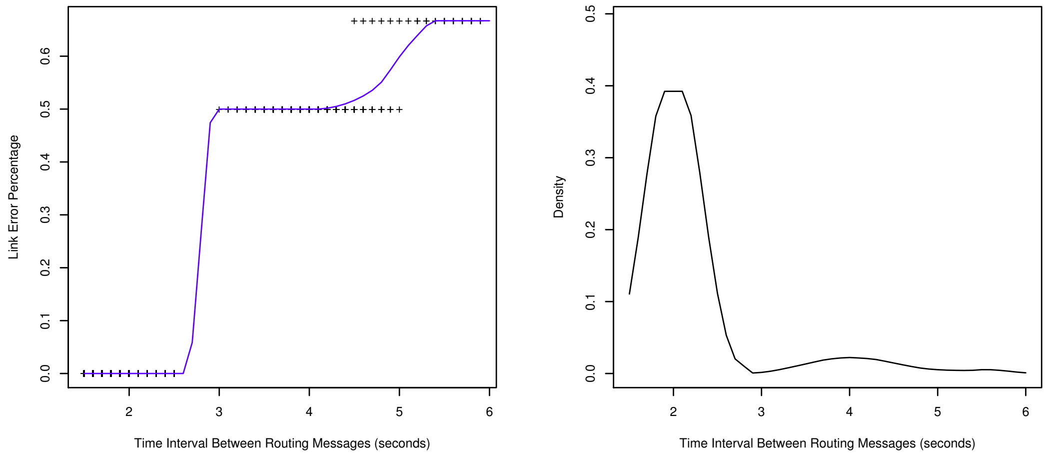
4.3.1 Link Quality Estimator Calculation through Simulation

In order to obtain data traces for the pair (x, q) required by the previously defined model, x being time interval Δt between routing messages and q the link quality, several simulations were performed using the OPNET simulator. These traces were obtained between two nodes placed at several fixed distances ($60m$, $65m$, …, $120m$). The Kernel Based Model will be calculated using the traces and its robustness is presented later in this section.

For each measured distance, 10 hours of routing traffic were simulated using the OLSR Protocol with a total of 50 runs using different seed values. The physical layer of the wireless nodes follows the Institute of Electrical and Electronics Engineers (IEEE) 802.11g ($54Mbit/s$) standard, uses a transmit power of $3.6 \times 10^{-4}W$ and a packet reception-power threshold of -95dBm, which results in a theoretical maximum range of $100m$ [Ortiz, 2009]. The actual range may vary as the OPNET simulator implements by default an accurate radio model, where asymmetric links or even unidirectional links may occur, as well as channel errors and multi-path interferences.

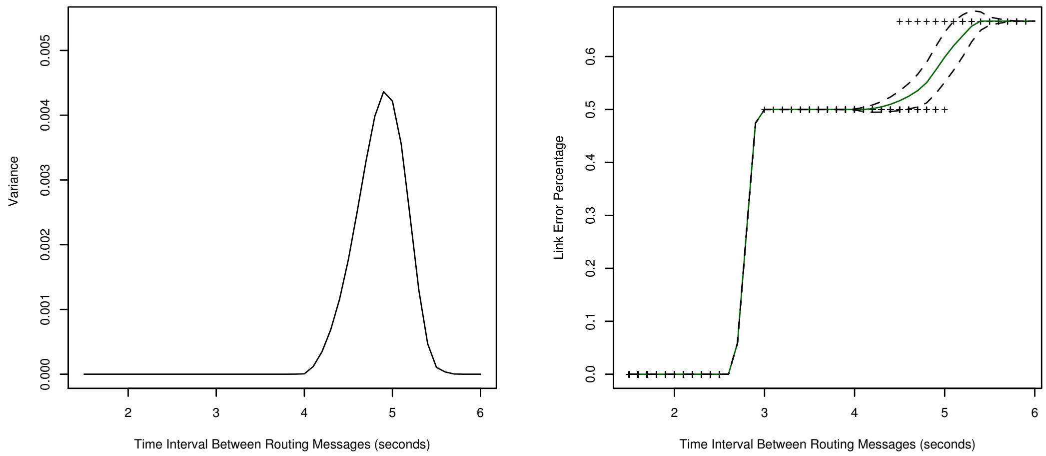
The R statistical language [R Development Core Team, 2010] was used together with the "locpol" package [Cabrera, 2009] in order to perform the required bandwidth computations and regression fitting. The obtained bandwidths and ASE errors are presented in Table 4.2. Both Epanechnikov and Gaussian Kernels were used in order to analyse the main differences between them. Figures 4.2 and 4.3 depict these two kernels and their main characteristics. These two figures include the time intervals up to a maximum of $6s$ which corresponds to the OLSR maximum hold time for a link and thus, not being registered, the link error percentage is always bellow 100%.

Figure 4.2 presents the Regression obtained by using the Epanechnikov Kernel. The density of values obtained for each time interval, x density, is depicted in Figure 4.2b. It is clear that the density is higher for lower time intervals, between 1.5 and 2.5 seconds,



(a) Data and Regression

(b) x Density



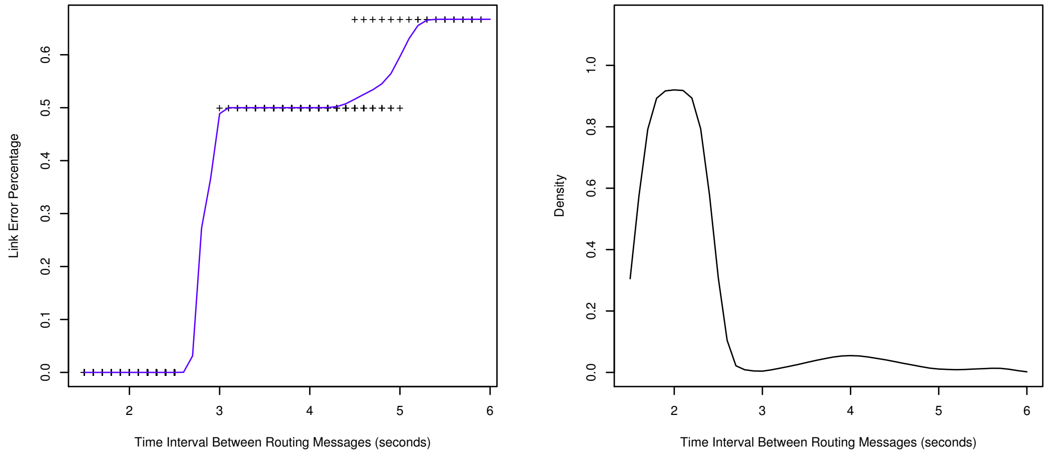
(c) Calculated Variance

(d) 95% Confidence Interval

Figure 4.2: The Epanechnikov Kernel Regression Results (simulation)

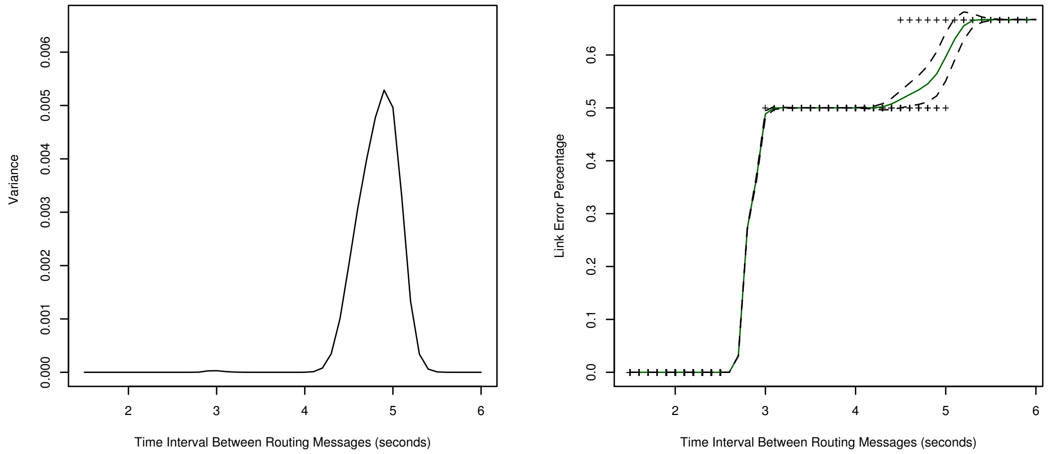
since they correspond to a better link quality with less errors and thus more delivered packets. On the other hand, for higher time intervals, density variations occur due to the physical layer specific operations such as transmission retries, which influence the final packet delivery. As it would be expected, higher time intervals were registered at larger distances [Anastasi et al., 2003; Pham et al., 2005], and there is never a time overlap between no lost packets and one lost packet, which will take at least $3s$, therefore a steep increase of the number of losses is registered in this interval.

In order to better understand the significance of the provided estimation, Fig-



(a) Data and Regression

(b) x Density



(c) Calculated Variance

(d) 95% Confidence Interval

Figure 4.3: The Gaussian Kernel Regression Results (simulation)

ures 4.2c and 4.2d depict, respectively, the calculated variance for each estimated value as well as a 95% confidence interval, showing that the chosen bandwidth value, obtained by minimizing the ASE, provides good final estimations. The results also show that the time difference between no lost packets and one lost packet, the difference between one and two lost packets is more challenging, being kernel regression a suitable tool to estimate these values.

Figure 4.3 depicts the obtained results using a Gaussian Kernel for the same trace values. It is possible to verify that the Gaussian Kernel Estimation provides a slightly

Table 4.3: Obtained Kernels (real)

	Epanechnikov	Gaussian
Bandwidth	0.45	0.14
ASE	1.174×10^{-4}	1.088×10^{-4}

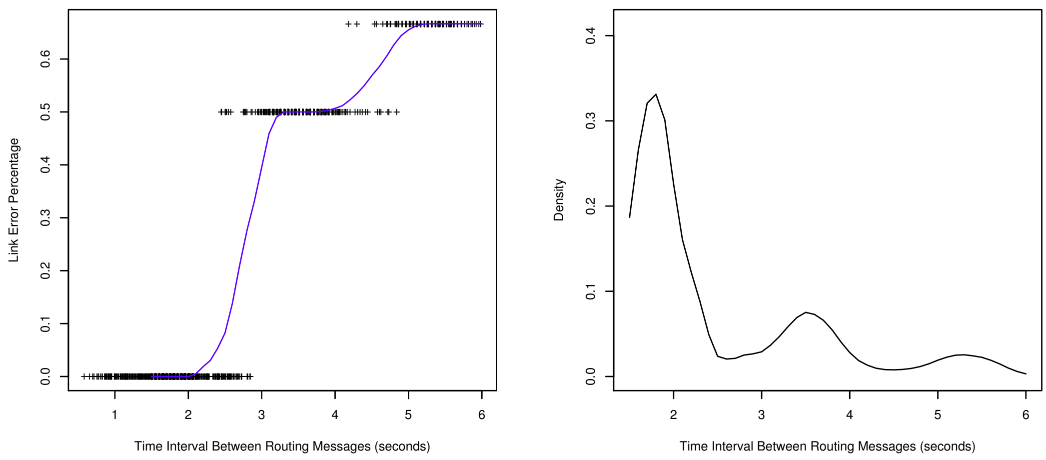
smoother regression while keeping similar estimated values throughout the x-axis. Nonetheless, it is noticeable in Figures 4.3c and 4.3d that the obtained estimation, when compared with the Epanechnikov Kernel, suffers from a minor increase on the variance for higher time intervals. This difference is irrelevant for the desired estimation as it depends on the specific used kernel function. However it is interesting to observe that despite both kernels have used the same method for bandwidth calculation, the Gaussian Estimator has the smallest calculated ASE.

4.3.2 Link Quality Estimator Calculation Using Real Traces

Similarly to the simulation-based traces, data traces for the pair (x, q), where x is the time interval Δt between routing messages and q is the link quality, were obtained by placing two nodes at several fixed distances ($60m$, $65m$, …, $90m$) with a clear Line-of-Sight (LOS). Beyond $70m$ of distance a degradation of the wireless link was registered. The maximum range for the used wireless equipment was found to be set at $85m$, after which no further packets were received.

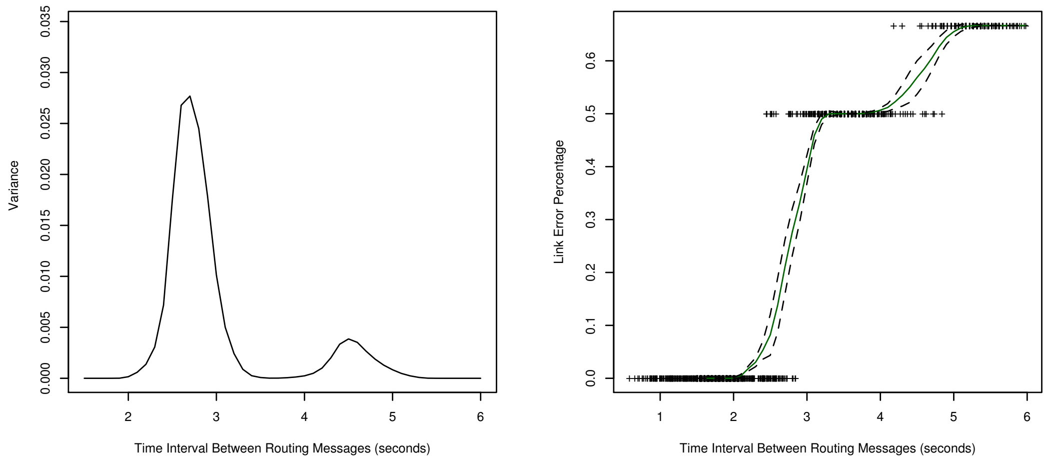
In order to obtain valid and useful traces, without requiring any specific hardware, two off-the-shelf Asus *EEE* netbooks were used. In particular, the used *EEE* model was the 1001 PX with an Intel Atom 450, 1GB Ram and an Atheros Communication Wireless Card (model AR9285, ath9k driver), running the Linux distribution Ubuntu 11.10 (using kernel 3.0.0-16-generic-pae). No modifications were performed to the hardware, using the default embedded antennas.

For each used distance, 2 hours of routing traffic were obtained using an implementation of OLSR Protocol, the OLSRd implementation by Tnnesen et al. [Tnnesen et al., 2012], with only minor modifications in order to register the time interval between received *HELLO* messages. These modifications were compliant with the protocol's specification, having no impact in the performance or interoperability of the protocol. The physical layer of the netbooks was configured similarly to the simulation scenarios, using the IEEE 802.11g ($54Mbit/s$) standard, configured to use channel 13, which was clear from interferences from other equipments. The height from the ground of the two nodes was set at $50cm$ approximately, as the wireless link is known to be strongly



(a) Data and Regression

(b) x Density



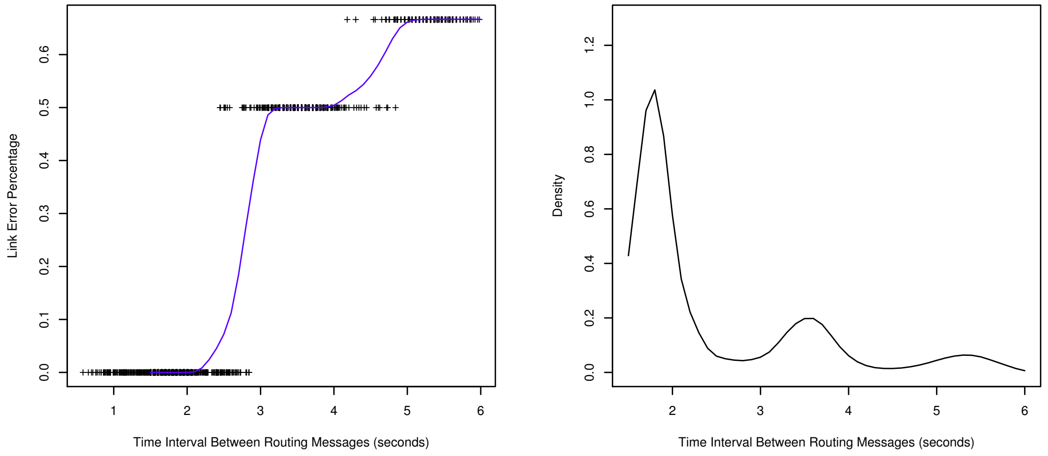
(c) Calculated Variance

(d) 95% Confidence Interval

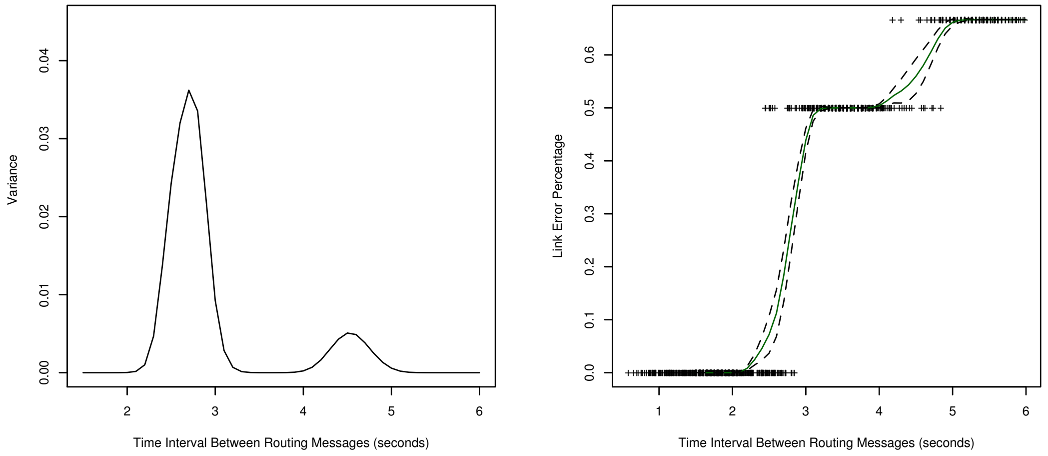
Figure 4.4: The Epanechnikov Kernel Regression Results (real)

influenced by height [Gaertner and Cahill, 2004], and therefore the ground was avoided.

The R statistical language was once again used in conjunction with the "locpol" package. Just like in the results obtained through simulation-based traces, the used bandwidths and ASE errors are presented in Table 4.3. Figures 4.4 and 4.5 depict the estimators obtained for the Epanechnikov and Gaussian Kernels, along with their main characteristics. The same link hold time characteristic for the OLSR protocol is maintained, resulting in a link error percentage always bellow 100%, with a maximum time interval of $6s$.



(a) Data and Regression (b) x Density



(c) Calculated Variance (d) 95% Confidence Interval

Figure 4.5: The Gaussian Kernel Regression Results (real)

The obtained Regression using the Epanechnikov Kernel is presented in Figure 4.4 and shows that it is comparable to the simulation-based regression. The most obvious difference is that the obtained data traces, depicted by the "+" symbols, are more erratic and provide a wider range of values. For instance, since the minimum time interval between two consecutive *HELLO* messages is $1.5s$, there should never be 0.5 link-error percentage in less than $3s$ as a message must have been lost first. However, due to propagation delays inherent to the wireless link and also retransmissions, in these situations, the first valid message has been received with a considerable delay,

Table 4.4: Obtained Kernels for 3 Packets

	Epanechnikov	Gaussian
Bandwidth	0.2999571	0.1331059
ASE	7.909×10^{-4}	7.877×10^{-4}

therefore reducing the total interval time until the next valid message is received.

The density of values obtained for each time interval, x density, is depicted in Figure 4.4b, where clear density peaks are registered at 2, 3.5 and 5.5 seconds, being the first more prominent since it corresponds to a better link quality and therefore more packets are received.

The significance of the presented estimation can be perceived by analysing calculated variance for each estimated value, as well as a 95% confidence interval for the regression, shown in Figures 4.4c and 4.4d respectively. When compared with the simulation results it is clear that a higher uncertainty exists. However, the obtained estimator is still very accurate and reflects the true dynamic nature of wireless Ad-hoc networks which can not be provided by any simulator.

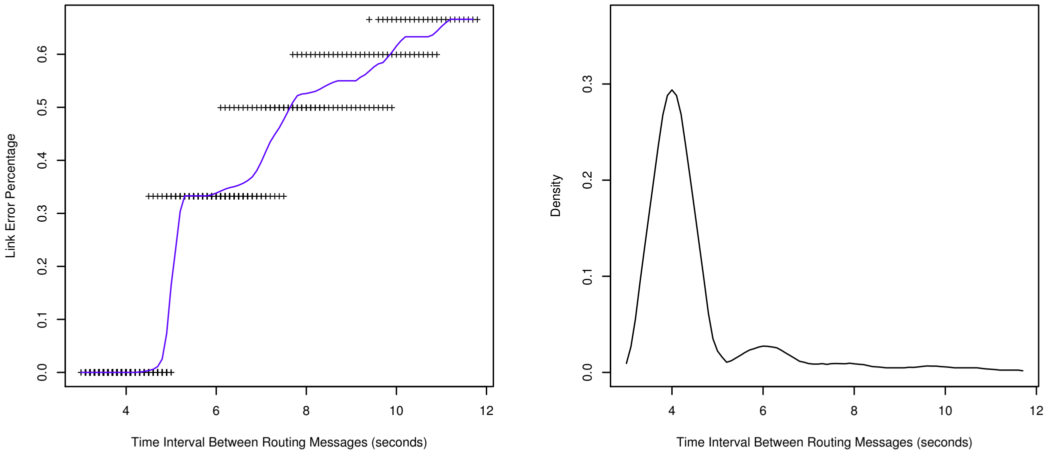
The usage of a Gaussian Kernel for the regression calculation, with the obtained real traces, is presented in Figure 4.5. Resembling what was observed with the simulation-based traces, the Gaussian Kernel Estimation seems to provide a more subtle fitting, without being over-smoothed, with a considerably lower bandwidth. Moreover, similarly to the simulation-based estimator, the Gaussian Kernel has a slightly higher variance comparing with the Epanechnikov Kernel, as shown in Figures 4.5c and 4.5d, even though it has a smaller ASE.

4.3.3 Additional Results

Using the same simulation-based traces, a different analysis could be performed, taking into account more than the interval between two received packets. For instance, if three *HELLOs* were used, a more detailed link quality estimation could be obtained. However, waiting for a third packet adds more delay in the estimation, such that the time interval between the first and the third message could be up to 12 seconds.

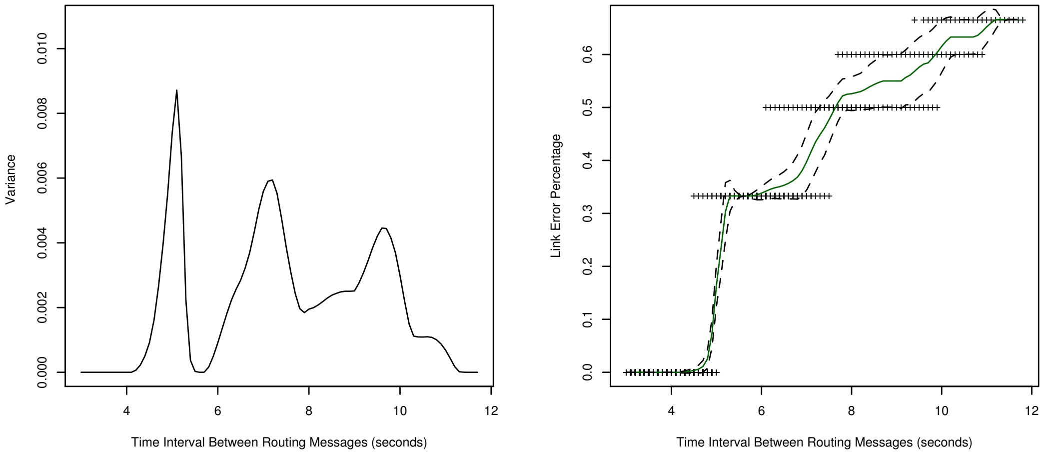
By using 3 packets for the determination of a wireless link's quality, the link quality is defined by the following equation 4.9:

$$LinkQuality_{\Delta t} = \frac{2}{2 + \text{packets lost}} \quad . \tag{4.9}$$



(a) Data and Regression

(b) x Density



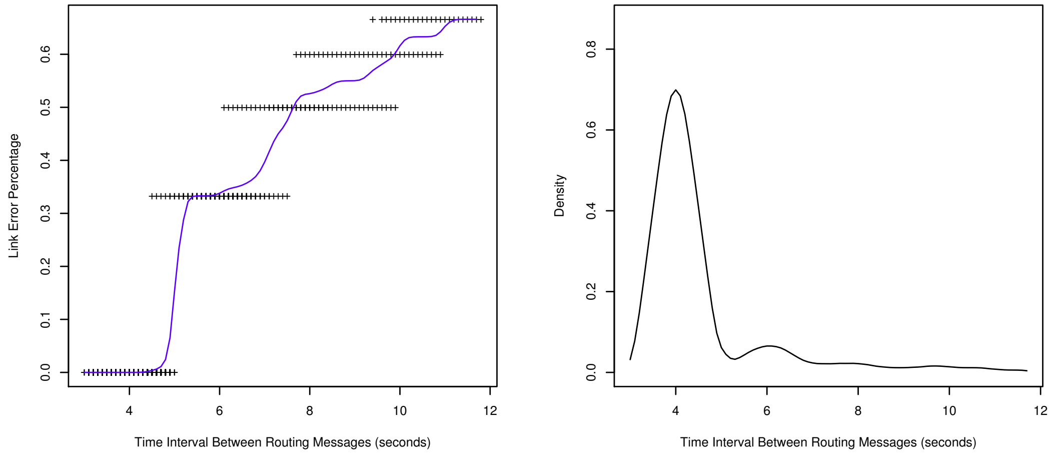
(c) Calculated Variance

(d) 95% Confidence Interval

Figure 4.6: The Epanechnikov Kernel Regression Results for 3 Packets

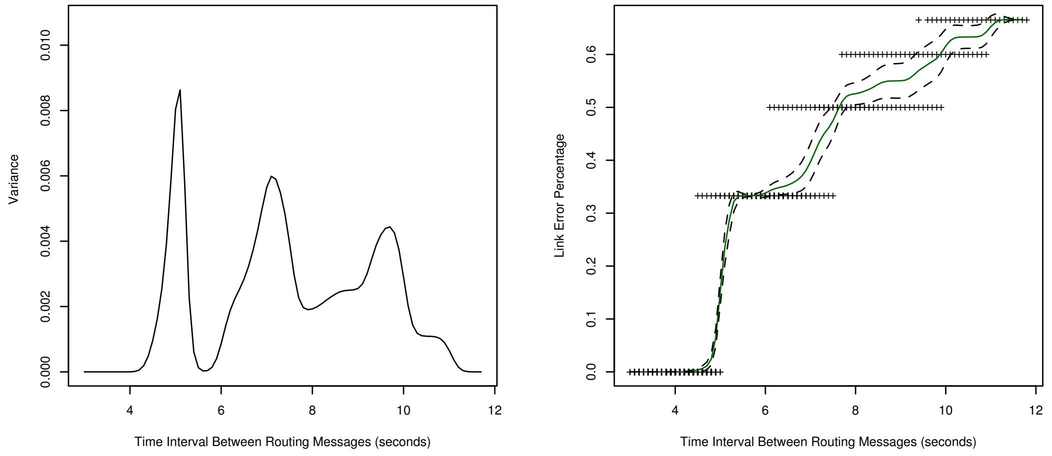
Figures 4.6 and 4.7 show the obtained regression for the previously analysed traces, allowing a more finely grained assessment of a link's quality. These results show that the confidence interval is not as good as for 2 packets, since they require a higher number of traces. Nevertheless, it is interesting to conclude that by having only a slightly less updated assessment of the wireless link still allows a good estimator with more detail.

The analysis of the time interval of the traces for different link errors shows an overlap for the same time interval. This is due to the addition of an extra routing packet in the analysis, reflecting the dynamic nature of the wireless medium. However, the



(a) Data and Regression

(b) x Density



(c) Calculated Variance

(d) 95% Confidence Interval

Figure 4.7: The Gaussian Kernel Regression Results for 3 Packets

obtained regressions resulted into reliable models, capable of providing precise estimates for the different time intervals. Comparing the results obtained through the usage of Epanechnikov and Gaussian kernels, the latter provided a regression with a smaller error for the same confidence interval of 95%, being therefore more accurate.

4.3.4 Model Validation

While a formal model is itself a contribution for the representation of a reality, its final application in a realistic scenario is also important, along with its precision. The proposed Link Estimation model has no unrealistic assumptions, requiring only the measurement of the time elapsed between receiving two consecutive routing messages, disregarding the protocol itself. Having this in mind, the obtained Kernel Estimators were implemented in the OPNET simulator and link quality estimations were performed in real time in two different scenarios with mobility, presented next. It is important to note, though, that any other scenario and mobility models could be used without requiring any recalculation of the obtained models, as long as the wireless physical layer specifications are maintained.

A simulation based study was performed to evaluate the obtained estimators, because it would not be feasible to have several real nodes available for a significant amount of time, nor would the scenarios be sufficiently manageable in order to allow the repetition of measurements and their validation.

Despite having used traces from static nodes to perform the Kernels Calculation, they are still suitable for any scenarios with or without mobility. An exhaustive evaluation was performed, where each simulation-based Link Quality Kernel Estimation was ran 50 times, using different seed values, with a total time of 10 simulated hours per run for each scenario.

4.3.4.1 High Density Scenario

For the validation process, a scenario was created with ten nodes moving freely in an area that exceeds more than twice the area covered by the nodes' maximum range (240×240 meters). The nodes follow the RWP Mobility Model [Shakeri and Hosseini, 2010], with a uniform speed between $3\ and\ 30km/h$, corresponding to pedestrians' walking speed or moderate driving [3GPP, 2008], with a pause time of 100 seconds. This scenario represents a significantly dense area which is prone to more packet collisions and errors. While it is a more academic scenario, it allows a thorough validation of the obtained Kernel Estimators.

4.3.4.2 Oulu Scenario

A more complex but realistic scenario was also created using Synthetic Map-based Mobility Traces [Aschenbruck and Schwamborn, 2010], which allows the definition of mobility traces according to real-world locations. This was obtained using the bonnmotion tool [Aschenbruck et al., 2010] and the Random Street Model, specifying a total area of 3000×3500 meters with 30 moving nodes in Oulu, Finland. Again, the used speed corresponds to pedestrian walking (uniform speed between $3\ and\ 30km/h$ with

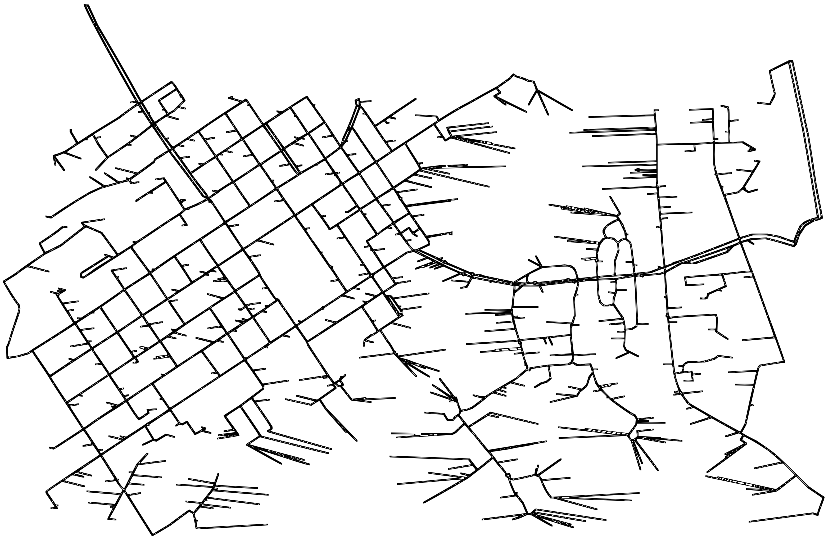


Figure 4.8: Oulu Mobility Trajectories

a pause time of $100s$), creating a fairly realistic scenario. The obtained trajectories are depicted in Figure 4.8.

4.3.4.3 Validation Results

As previously mentioned, wireless link quality is influenced by the amount of errors that may occur when transmitting a packet. These errors result from many aspects such as the transmission power, interferences and collisions which may vary from packet to packet. For each simulated run, all the generated packets have been registered as well as all the link quality estimations made by the used models. By comparing the amount of packets with errors (i.e. not received) per link and the determined link quality for each pair of received *HELLO* messages, it is possible to determine the performance of the obtained models.

Comprising all the validation results, Table 4.5 presents the overall link error percentage for both Kernel Based Estimations and real values. This table presents the actual average link quality and standard deviation, registered by analysing all the generated packets next to the results obtained by both the presented link quality estimator models for both scenarios. As it can be seen, both Kernel Estimators performed extremely well in a realistic implementation in two different scenarios with different mobility models and characteristics. In particular, it is worth noting that the difference between real and estimated errors is small, proving that Kernel Methods are good estimators and provide a good generalization. Moreover, it has been shown that the

Table 4.5: Combined Results (50 runs)

	Random Waypoint			Random Street		
	Average Link Errors	Standard Deviation	Total Difference	Average Link Errors	Standard Deviation	Total Difference
Real Values	0.0198374	0.00254325	-	0.0190814	0.0211598	-
Epanechnikov	0.0193507	0.00234906	0.0004867	0.0184662	0.0196664	0.0006152
Gaussian	0.0195183	0.00234445	0.0003191	0.0186297	0.0195614	0.0004517

calculation of the proposed model does not require traces with mobility in order to obtain a good performance, further proving the quality of the provided models and their applicability.

Being able to determine the amount of link errors in real-time is an important feature of link quality estimators. All the presented results were obtained instantaneously during the performed simulations, since no recalculation is needed for any scenario. For example, in the high density scenario, the average difference between the real link errors and the estimated errors, obtained for one run, is represented in Figure 4.9. This figure shows the performance of both the Epanechnikov and Gaussian Kernel Estimators and, despite being quite small even in the beginning of the simulation (less than 0.12%), the difference between the estimated and true error percentage gets even smaller throughout the time for both Kernels.

Similar results are presented in Figure 4.10, showing that for a more realistic scenario the performance is maintained. However, despite very small, a fluctuation is registered suggesting that the dynamic characteristic of this scenario may present different link behaviours that the used model estimators were still able to cope with.

The proposed link model using Kernel Estimators proved that by taking into account the feedback from the protocols' own routing messages, which are periodically sent and received, the protocol could improve its own routing decisions by incorporating link quality in the used path calculation algorithms.

4.4 Summary

A new model for link quality estimation, which requires no unrealistic assumptions, has been proposed. This model was derived from Kernel Regression Estimation techniques, resulting in an accurate estimation of the wireless link quality by analysing routing packets' inter-arrival times. Such results are extremely relevant for future wireless communications, allowing routing protocols to choose the best available links without

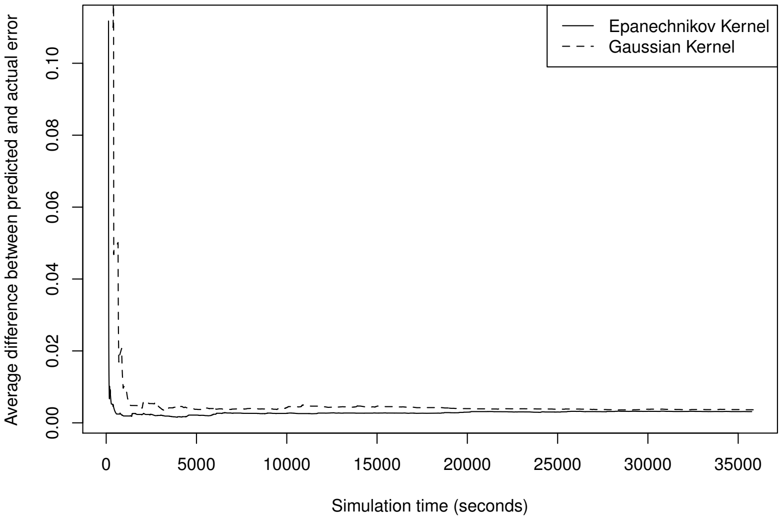


Figure 4.9: High Density Scenario - Real-time Performance

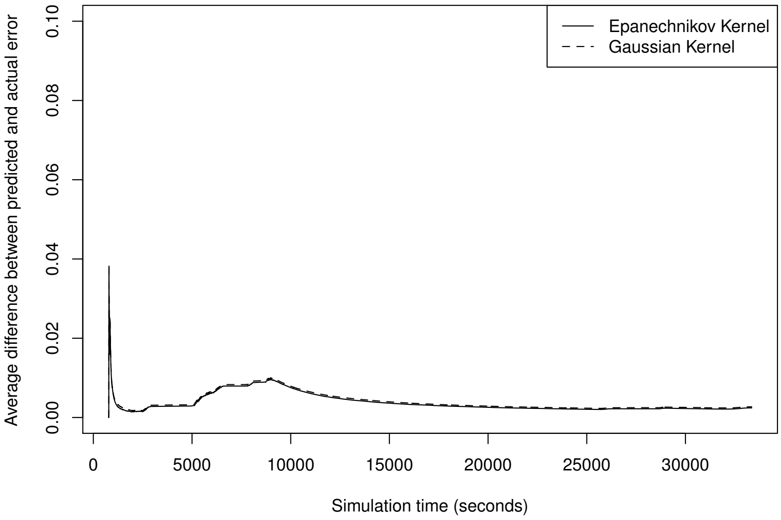


Figure 4.10: Oulu Scenario - Real-time Performance

additional messages overhead or imposed limitations.

The presented results were obtained by using the OLSR protocol routing messages. However, any routing protocol or periodically sent messages (e.g. Layer-2 messages) can be used. Taking these messages into account, two different Kernel Regression Estimators were used to determine in real-time the quality of a link. Moreover, it has been shown that it is possible, using only simple traces (i.e. fixed distances and no mobility), to calculate such estimators for any given scenario.

An important conclusion is that both the resulting kernel estimators were able to provide realistic estimates in scenarios with different types of mobility and node density. This suggests that in a real scenario the required traces can easily be obtained and used in a myriad of situations. In addition to this, kernel methods do not require any training (supervised or not), being more efficient than other techniques such as neural networks.

Most importantly, the obtained link quality estimator can be directly employed in Deferred Routing by using it in the "metric" parameter, defined by each GW when creating new Context Connectivity Information. This will allow the best GWs to be chosen, ensuring that poor wireless links are less likely to be used.

The work presented in this chapter has been published in Elsevier's Computer Networks Journal, with the title "Link quality estimation in wireless multi-hop networks using Kernel based methods", 2012.

Chapter 5

Deferred Routing Performance

The performance of a routing protocol can be assessed through several parameters. Typically, a protocol's competency to deliver data packets successfully, paired with the end-to-end delay of the chosen path, determines whether a protocol has a good performance or not. However, in Mobile Ad-hoc Networks there are several other metrics and characteristics that must be analysed, such as scalability and resilience to mobility. Bearing this in mind, the performance of the Deferred Routing (DefeR) protocol was evaluated by using seven distinct metrics that are indicative of the protocol's behaviour.

In order to evaluate the protocol with a greater degree, a set of different scenarios sustaining distinguishable features was used to portray the protocol's abilities. In particular, the scalable properties of the protocol will be assessed, as well as the validity of routes when subject to different mobility models, exploring the robustness of the proposed hierarchy and routing scheme.

Regarding all the provided results, they were obtained using the OPNET Modeler Wireless Simulator [OPNET Technologies, 2012], where the considered wireless nodes follow the Institute of Electrical and Electronics Engineers (IEEE) 802.11g standard [Ortiz, 2009] at 2.4Ghz, and have a maximum range of 100 meters (Transmit Power of $3.7e^{-4}W$), which corresponds to the maximum obtainable range of common wireless cards [Anastasi et al., 2004; Xing et al., 2009], unless stated otherwise. Nonetheless, due to the accurate radio model implemented by default in the OPNET Simulator, asymmetric links or even unidirectional links may occur, as well as channel errors and multi-path interferences respectively. Moreover, the Consultative Committee on International Radio (CCIR) propagation model was used, configured to represent a small to medium city with a building coverage of 15.8 percent, as it is considered as an appropriate propagation model for Mobile Ad-hoc Networks (MANETs) [Myers and Richardson, 1999]. The usage of this simulation environment strives for being more realistic when compared with other works, which use the outdated 802.11b with non-standard Medium Access Control (MAC) layers and unlikely ranges (for instance, 250m). Each evaluated scenario has specific variations of several simulation parameters since they independently assess different characteristics. Simulation parameters not mentioned here or in the definition of the scenarios are defined with the values used by default in the OPNET Modeler Wireless Suite Simulator, version 16.0.A PL1.

All the different parameters varied in each of the defined scenarios were obtained after 30 runs per parameter, always using different seed values and the Linear-Congruential Random Number Generator Algorithm, for a total simulated time of 15 minutes (900 seconds per run), which allows routing protocols to be appropriately evaluated by guaranteeing enough mobility [Sagar et al., 2012].

Taking into account the defined objectives of this evaluation and their statistical validity, all the presented results have a 95% confidence interval obtained from the central limit theorem, which states that regardless of a random variable's actual distribution, as the number of samples (i.e. runs) grows larger, the random variable has a distribution that approaches that of a Normal random variable of mean m, corresponding to the same mean as the random variable itself.

5.1 Evaluation Metrics

As previously mentioned, in order to provide a thorough evaluation of the DefeR routing protocol and its behaviour in large scale networks, it is important to assess the performance of different routing aspects, choosing appropriate comparison metrics. For this purpose the following items were considered in provided evaluation:

- Traffic Delivery Performance
  - Losses
  - End-to-end Delay.
- Routing Performance
  - Path Length
  - Routing Stability
  - Control Traffic Overhead.

Taking these different aspects into consideration, this performance assessment must involve the evaluation of a large scale network, measuring the stability and overhead of this concept, as well as its overall traffic delivery performance. Moreover, in order to allow a more exhaustive evaluation it is important to determine the protocol's ability to handle mobility phenomena, introducing dynamic scenarios with different mobility models.

The average percentage of losses and end-to-end delay reflect a protocol's competency to choose suitable paths and are taken into account in this evaluation in all the presented scenarios. The percentage of losses strongly influences the applicability of a routing protocol in different scenarios. However, in Mobile Ad-hoc Networks a high number of losses is expected due to its inherent nature, where nodes are intermittently connected and where interferences and collisions are frequent [Cao and Sun, 2012]. Moreover, the delay metric is also subject to these interferences, limiting the usage of real-time applications in some scenarios. Nonetheless, in an extreme outlook, where only MANETs may be available, the registered losses may not be significant and retransmission mechanisms can be used to successfully deliver the required data.

A different routing metric considers the path length (hop count), from source to destination, which typically is minimized by routing protocols in order to reduce the number of nodes that intervene in the data delivery process. By reducing the number of hops, protocols are expected to be more energy efficient. However, this is not always the best option, as bottlenecks may arise and collisions will not only originate more losses but also a faster energy depletion on nodes in "popular" paths. Regarding this aspect, the DefeR protocol follows a different approach, choosing paths that minimize the total number of cluster-hops, selecting the most suitable Gateway (GW) nodes according to their quality.

In addition to these metrics, it is also important to measure the required resources and, therefore, routing traffic overhead, as well as the stability of the existing routes. Regarding the latter aspect, mobility of nodes is responsible for most of the topology changes and it is the protocol's task to efficiently handle these changes.

The topology awareness of a routing protocol is a metric representative of a routing protocol's stability and knowledge about the network's structure, registering topology changes during the simulation. A topology change occurs whenever a new Topology Control (TC) or a TC with a higher sequence number is received and also when a TC entry is deleted after expiry. Each topology change triggers a routing table recalculation, however, in order to reduce computational overhead, the routing table is only recalculated by default at most every 1 second, processing all the received topology changes between each recalculation. Such technique is compliant with the Optimized Link State Routing (OLSR) specification and used in existing implementations [Golubenco, 2012; Tnnesen et al., 2012]. Moreover, all the analysed protocols use this improvement in order ensure a fair comparison between them.

The amount of processed topology changes in routing table calculation reflects a protocol's stability and will also be analysed, referred as Average Topology Changes per Routing Table calculation (AToCRT) and defined by equation 5.1.

$$AToCRT = \frac{\text{Number of Topology Changes}}{\text{Number of Routing Table Calculations}} \tag{5.1}$$

The number of routing table calculations possible in a T seconds simulation is defined in equation 5.2, with i being the simulation instant where n Topology Changes occur. Since the number of topology changes is influenced by the mobility of nodes, the different speeds used in an evaluation will be reflected in the AToCRT metric and also on the total number of routing table calculations. In particular, with higher speeds, an increased number of Topology Changes throughout the time will trigger a higher number of routing table calculations, with a maximum of 1 per second, as defined by $f(n)$.

$$\text{Routing Table Calcs} = \sum_{i=1}^{T} f(TopologyChanges_i),\ f(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n > 0 \end{cases} \tag{5.2}$$

Topology Changes are propagated by *TC* messages sent by Multipoint Relay (MPR) nodes. These messages are forwarded to all the elected MPR nodes and represent most of the routing overhead, as they are the only forwarded messages sent throughout the network. The number of forwards per *TC* messages must then be analysed, in order to correctly assess the scalability of a protocol.

The overall routing overhead must also be considered taking into account the periodically sent and received routing traffic from both *HELLO* and *TC* messages. This will also reflect the protocol's ability to handle a large number of nodes.

For comparison purposes, all the presented scenarios have been used to evaluate the DefeR, OLSR and Cluster-based OLSR (C-OLSR) protocols. These comprise respectively the different approaches available for proactive protocols, employing hierarchical clustered routing, flat un-clustered routing and flat clustered routing. By analysing the three approaches it is easier to understand which one is more suitable for large scale networks. The OLSR protocol was chosen as a control subject, providing a basis for comparison due to its popularity in MANETs, being a standard protocol currently under improvement by the MANETs Internet Engineering Task Force (IETF) working group in its second version [Clausen et al., 2012].

Regarding the creation of clusters used by both the DefeR and C-OLSR protocols, a static definition of the areas comprised by each cluster was used and a mechanism for the nodes to automatically update their Cluster Identifier (CID) was implemented. However, this approach does not guarantee a constant density of nodes within each cluster. Such limitation impacts the performance of both protocols, since, in a worst case scenario, all the nodes might move into one single cluster. Nonetheless, in a realistic scenario clustering algorithms may not be able to guarantee constant density unless they introduce limitations of their own (such a single-hop cluster coverage) [Yang et al., 2007].

5.2 Scalability Assessment

In order to assess how scalable the mechanisms of a protocol are, a set of results where the total number of nodes increases should be obtained. By increasing the number of traffic flows, it is also possible to understand how the protocol handles not only the size of the network, but also how it copes with a demanding network where several routes must be established.

5.2.1 Description

Following an approach where a growing size network is used, the total number of node clusters is incremented presenting a scalability evaluation of the DefeR scheme. This evaluation depicts the behaviour of this protocol with both small and large-scale networks. It is a straightforward assessment which somewhat disregards the nature of MANETs, as it does not take into account the natural behaviour of moving people, being entirely random regarding both mobility and traffic flows.

A set of results from 1 cluster up to 10 clusters is provided, where each cluster has 49 nodes (which is the best number of nodes handled by OLSR [Palma and Curado, 2009]). The dimension of each cluster is of $500 \times 500m$, ensuring an initial constant density of the network. Figure 5.1 depicts the configuration of the network used in this scenario.

Regarding the smaller simulated networks with one cluster, both C-OLSR and DefeR behave exactly like OLSR, as both use it for intra-clustering and no inter-cluster

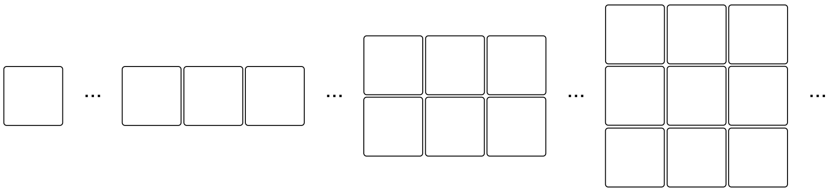


Figure 5.1: Increasing Number of Clusters

operations are required. The provided results for smaller networks are important as other protocols designed for large-scale networks, such as Dynamic Address Routing, are known not to perform well in smaller networks [Eriksson et al., 2007].

In order to assess how the DefeR protocol handles networks with a different number of traffic flows, the different size networks were also evaluated with 1, 4, 8 and 16 traffic flows. Each flow begins randomly after an interval between 50 and 250 seconds of simulation time, uniformly distributed, being concluded by the end of the simulation. The destination of each flow was randomly chosen, using a User Datagram Protocol (UDP) traffic type, with a constant bit rate of 8 packets of 4kbit per second, representative of typical interactive gaming, simple file transfers or information exchange [ITU-T, 2003].

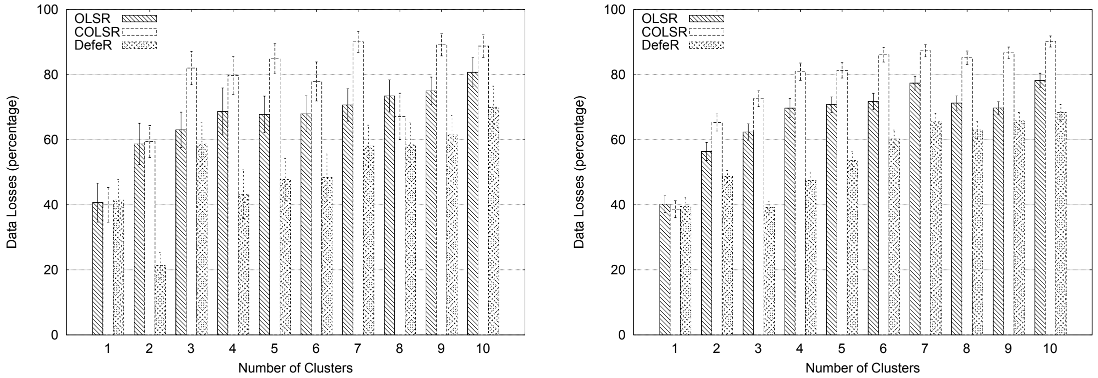
As previously mentioned, in this scenario the DefeR protocol's ability to maintain a reduced overhead in scenarios where nodes are likely to move within nearby contexts is disregarded. All nodes randomly start their movement after an initial warm-up time, between 100 and 250 seconds (following an uniform distribution). The used mobility model is the Random Waypoint with a pause time of 60 seconds, without any distance or cluster restrictions, such that nodes are able to move freely across the entire network. The nodes' speed is uniform between $2\ and\ 6km/h$, corresponding to pedestrians' walking speed [3GPP, 2008].

5.2.2 Obtained Results

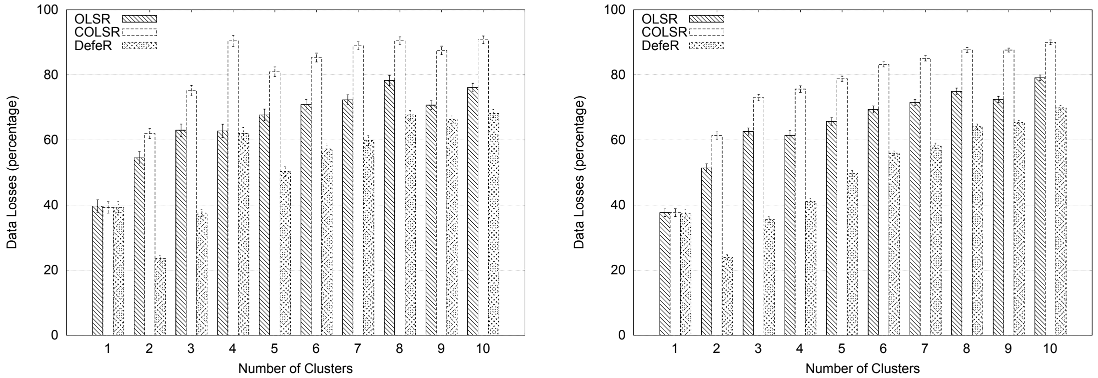
Taking into account the discussed evaluation metrics, the obtained results in the defined scenario are presented next. Each metric is presented with the four different number of flows, side-by-side, in order to allow a better comparison.

5.2.2.1 Percentage of Losses

In any routing evaluation, the percentage of registered losses can be considered as an indicator of how a routing protocol performs. This is presented in Figure 5.2, where the obtained percentage of losses is clearly influenced by the number of clusters in the



(a) 1 Flow (b) 4 Flows

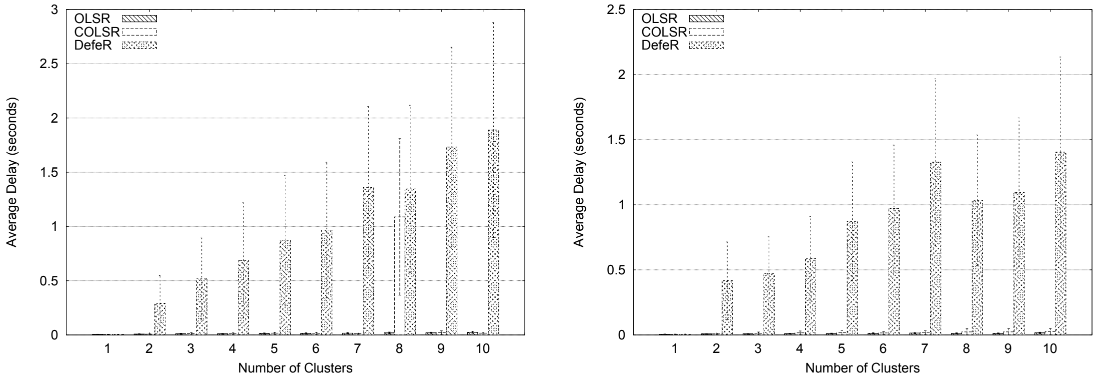


(c) 8 Flows (d) 16 Flows

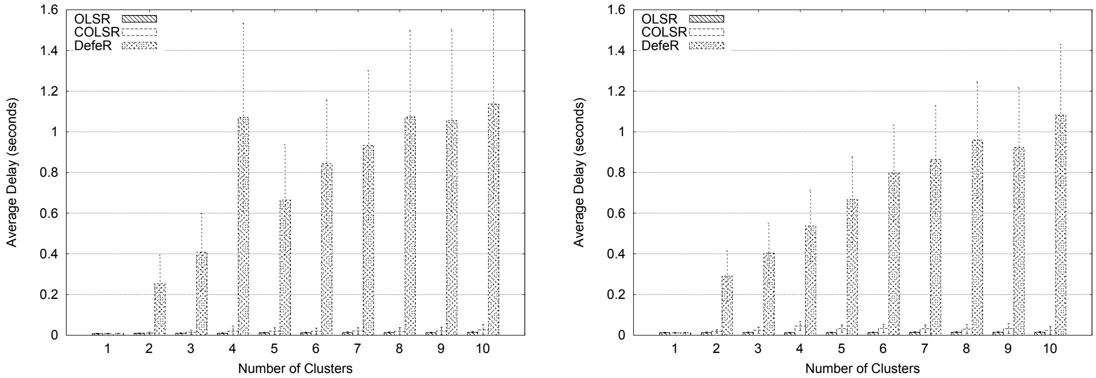
Figure 5.2: Average Percentage of Losses

network. In a 1-cluster network, the three protocols have a similar performance, as all of them simply use the OLSR protocol for maintaining routing paths. While the growing number of flows varies only slightly the data traffic delivery performance, the increasing number of clusters has a higher impact, such that the C-OLSR protocol registers more than 80% of losses in networks with 4 or more clusters.

Regarding the overall percentage of losses, the DefeR protocol registers the best performance being able to constantly deliver more data packets than its competitors. However, the DefeR protocol still has a significant amount of losses in larger networks, which is consistent with the performance of other protocols such as the Destination-Sequenced Distance-Vector (DSDV) or the Dynamic MANET On-demand (DYMO) routing protocols [Nordemann and Tonjes, 2012]. Though many losses are not desirable, this results from the intrinsic nature of MANETs. It is important to take into account that the proposed scenario is extremely demanding, where a path from source



(a) 1 Flow (b) 4 Flows



(c) 8 Flows (d) 16 Flows

Figure 5.3: Average End-to-end Delay

to destination may often not exist. Despite this fact, the proposed routing approach managed to perform two times better than the C-OLSR protocol in some network configurations.

5.2.2.2 End-to-end Delay

In realistic multi-hop wireless networks, as previously discussed, the constraint of an existing path between any two nodes cannot be guaranteed. As a result Delay-Tolerant Networks have been proposed [Nordemann and Tonjes, 2012], focusing on the delivery of data packets, regardless of the time interval it might take between source and destination. While the OLSR and C-OLSR protocols simply discard packets when a route is not found, the DefeR gateways are able to re-route packets if alternative paths exist. As a result of an improved traffic delivery, the DefeR protocol has a higher end-to-end delay, as seen in Figure 5.3. A similar delay is found in the C-OLSR protocol for an eight cluster scenario with solely 1 traffic flow, where this protocol has an abnormal improvement in traffic delivery (see Figure 5.2a).

Considering the class of reactive routing protocols, the path discovery process is responsible for initial delays even higher than the ones registered by any of the three analysed protocols [Trindade and Vazao, 2011b].

Even though the DefeR scheme is outperformed by the other two protocols, when delay is considered, its increased traffic delivery must not be disregarded as it helps to understand its origin. In fact, after a closer analysis of the obtained results, the high standard deviation reveals that the registered delay is only introduced by some flows, which are likely to be failed by the other protocols. This is the only reason for such a standard deviation, as the three protocols were equally simulated 30 times and only DefeR was this dynamic.

5.2.2.3 Path Length

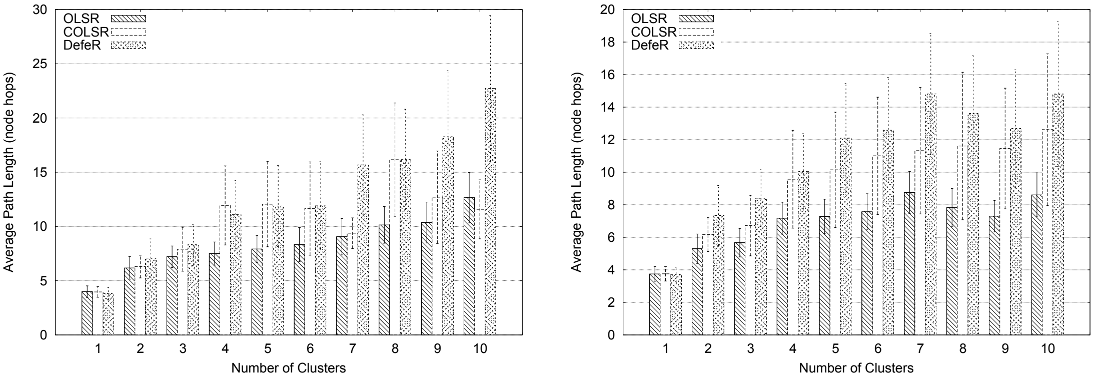
A different evaluation metric that confirms the DefeR ability to deliver packets to more challenging destinations is the average path length. The number of hops from source to destination is presented in Figure 5.4, where the OLSR protocol stands out for being able to achieve shorter routes. Regarding the cluster-based routing protocols, the DefeR protocol is able to keep up or even surpass the C-OLSR protocol's performance, while always delivering more data packets.

Once again, the increasing network size proportionally affects the metric results. However, while the average path length increases with the number of nodes, it decreases with a higher number of traffic flows. A similar behaviour was found with the delay metric, as it is also influenced by the number of intervening nodes in the deliver of data packets.

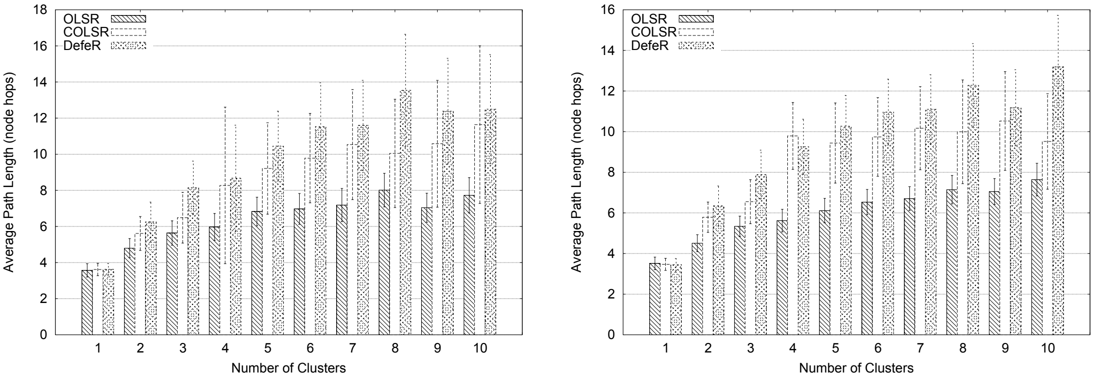
5.2.2.4 Topology Changes Per Routing Table Calculation

In MANETs, topology changes are likely to occur very often, not only due to interferences but mainly due to the mobility of nodes. It is the routing protocol's responsibility to detect existing topology changes and reflect them when updating its routing table. However, too many topology changes have a strong impact on the overhead introduced by a routing protocol and may reveal that the protocol suffers from instability.

In Figure 5.5 the lack of scalability of the OLSR protocol becomes clear, resulting in a growing number of registered topology changes in networks with a higher number



(a) 1 Flow (b) 4 Flows



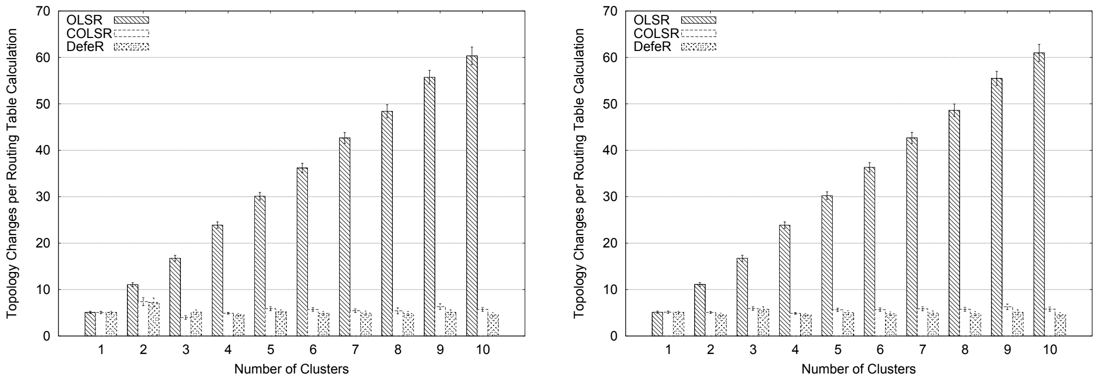
(c) 8 Flows (d) 16 Flows

Figure 5.4: Average Number of Hops

of nodes. On the other hand, the use of clusters by the DefeR and C-OLSR protocols allows them to achieve a more stable routing performance, keeping a fairly constant number of topology changes per routing table calculation. However, important topology changes cannot be disregarded by routing protocols. Regarding the overall routing performance of the C-OLSR protocol when compared with its unclustered version, even though it is more stable, it fails to achieve a similar traffic delivery, suggesting that its handling of topology changes does not have the same efficacy.

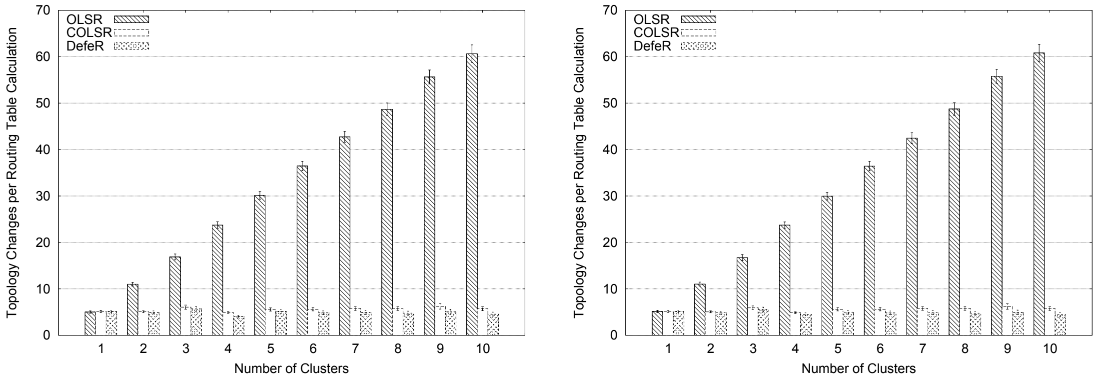
5.2.2.5 Number of Forwards Per *TC* message

Closely related with the detected number of topology changes, the ratio between sent and forwarded Topology Control messages is also a token of a protocol's ability to scale. The forwarding of *TC* messages deals with a large amount of overhead in the network



(a) 1 Flow

(b) 4 Flows



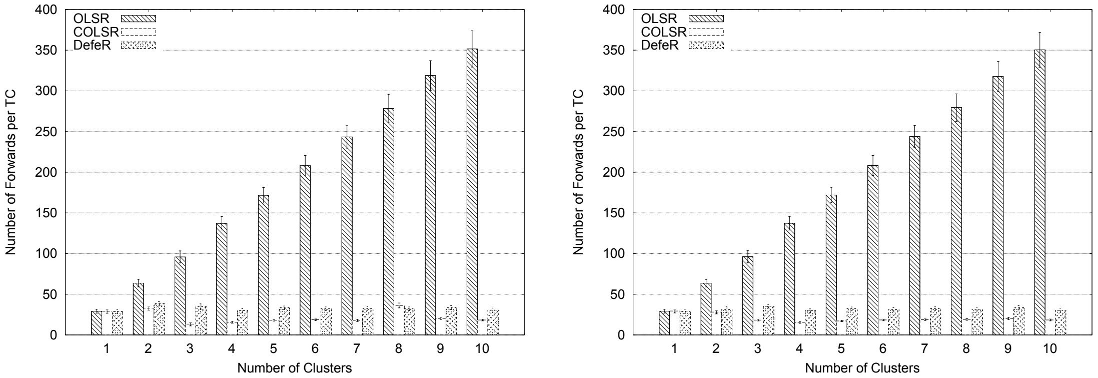
(c) 8 Flows

(d) 16 Flows

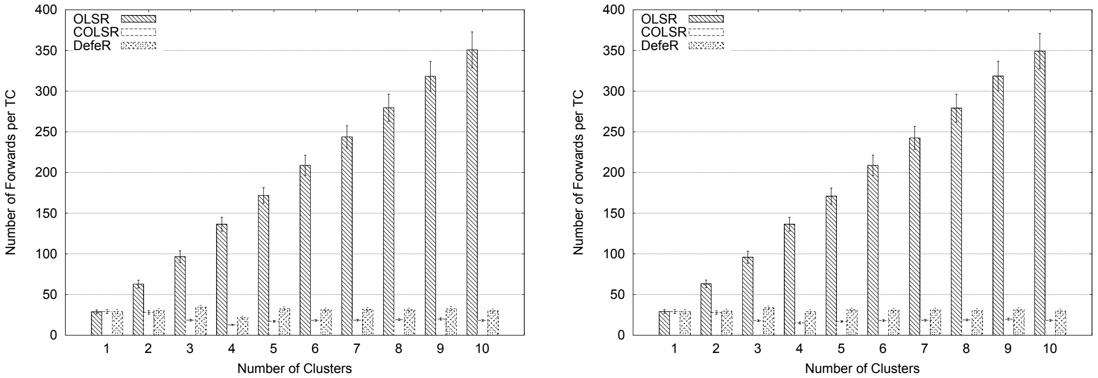
Figure 5.5: Topology Changes per Routing Table Calculation (AToCRT)

and should be kept to a minimum. Due to containment of routing information within clusters, the DefeR and C-OLSR protocols require a rather small number of forwards in order to disseminate their routing information – though the C-OLSR protocol requires the smallest amount of forwards. However, once more, an excessively low number of updates may indicate that existing routes are not entirely valid.

Regarding the number of traffic flows, there is no obvious impact on this metric, as it only depends on the existing number of nodes and topology changes. The latter aspect is clearer in the OLSR protocol, as seen in Figure 5.6 which shows that it requires its *TC* messages to be forwarded to most of the nodes in the network.



(a) 1 Flow (b) 4 Flows



(c) 8 Flows (d) 16 Flows

Figure 5.6: Number of Forwards per *TC* message

5.2.2.6 Control Traffic Overhead

Since only purely proactive routing protocols are being considered in this evaluation, the number of traffic flows does not influence significantly the number of required routing messages. Figure 5.7 shows the total overhead of routing control traffic issued by each protocol in the scenario with 16 Flows. As the number of nodes increases, the amount of existing routing information also increases for any proactive protocol. However, the DefeR protocol increases its overhead slower than its competitors, since it requires less routing messages. Moreover, the performance of the proposed protocol can be further improved by using a clustering algorithm that provides a table with the mappings of each node to its CID, as they usually use such a table for cluster maintenance purposes.

The overhead felt by the sent routing messages is more clearly noticed by the received routing information in the entire network. While *HELLO* messages are only

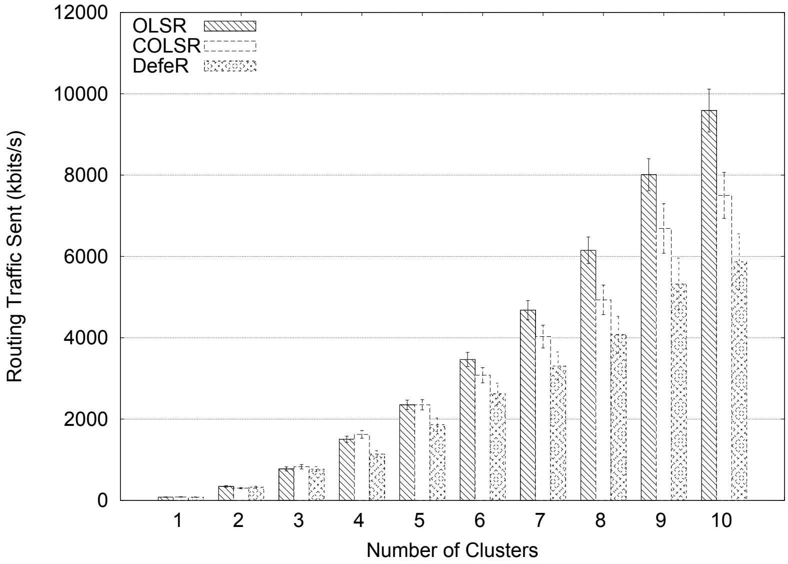


Figure 5.7: Sent Routing Traffic Overhead

sent locally, the previously analysed ratio between sent and forwarded *TC* messages determines how much more overhead is propagated through the network. Even though the C-OLSR protocol has a slightly lower ratio of forwarded *TC*s, when compared with DefeR, it has a higher received routing traffic overhead, as it sends more routing data per message. The received control traffic overhead for each protocol is presented in Figure 5.8.

5.3 Resilience to Mobility

The considered performance assessment must involve not only the evaluation of a large scale network, measuring the stability, overhead and overall traffic delivery performance, but also its ability to handle mobility phenomena, introducing dynamic scenarios with different mobility models.

Regarding this last aspect, even though many mobility models have been proposed in previous works, each one of them has unique characteristics, therefore not replacing one other.

5.3.1 Description

In this evaluation, several mobility patterns will be taken into consideration. In order to do so, the BonnMotion tool [Aschenbruck et al., 2010] has been used to generate

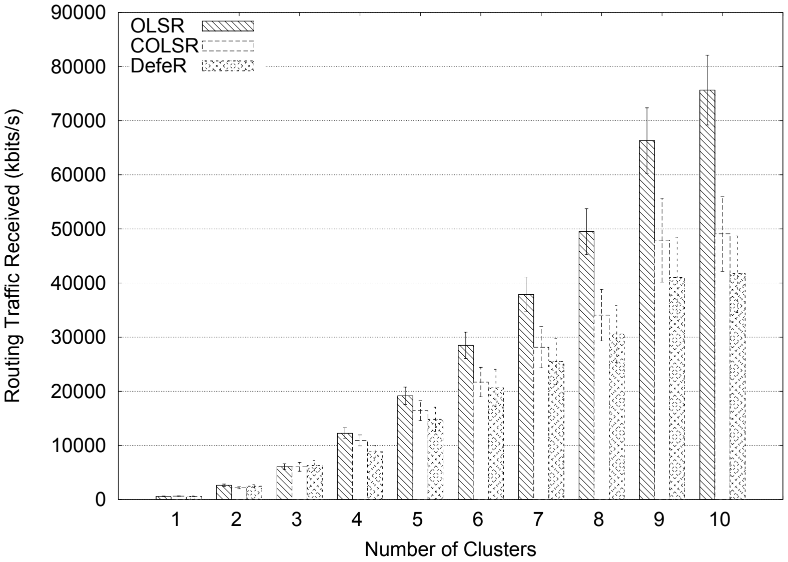
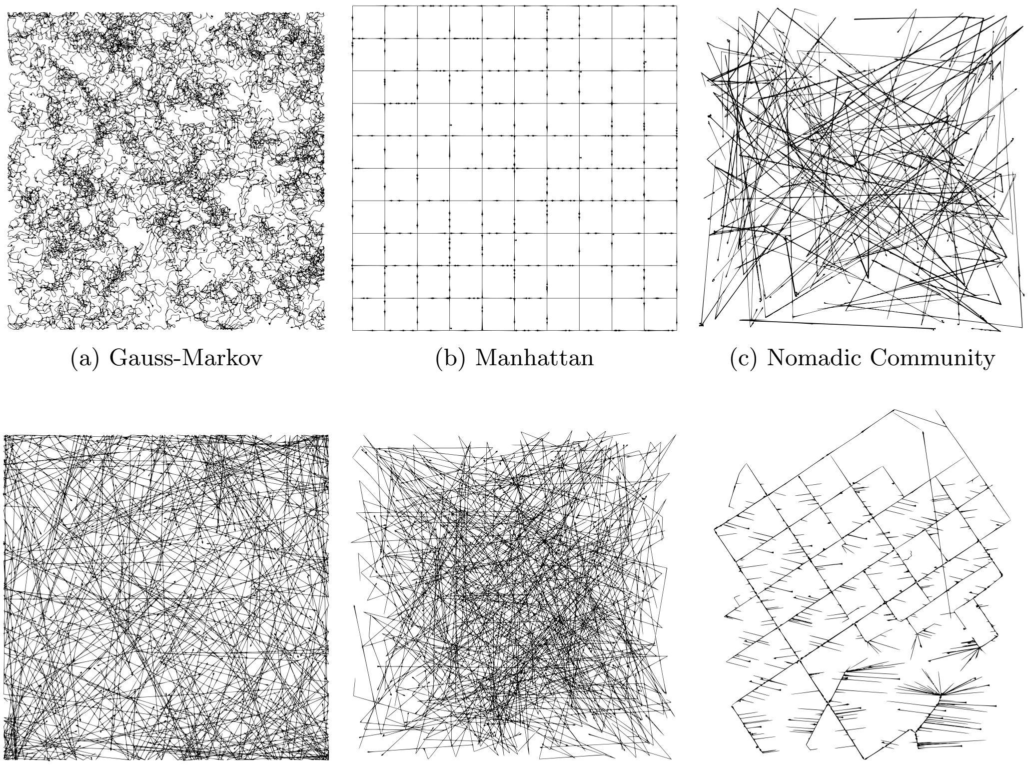


Figure 5.8: Received Routing Traffic Overhead

different node trajectories, later employed in conjunction with the OPNET Modeler Wireless Simulator. These trajectories were created assuming a plausible speed for a person walking [3GPP, 2008], between 0.5 and 1.5 m/s and a pause time of 60 seconds, when applicable. The mobility generation disregarded the first 3600 seconds, solely using the follows 900 seconds of path randomization, avoiding the initial warm-up from the random number generations, thus achieving a more stable scenario. Moreover, the area of motion was of 1500 by 1500 meters, for a total number of 541 nodes. Higher speeds were not considered, as the sense of clusters would be faded away and the realm of vehicular Ad-hoc networks would be entered. Even though new mobility models already present similarities with human mobility, the used mobility patterns were chosen for the sake of comparison with existing works on this subject. Moreover, being the DefeR protocol designed to explore spatial locality, it would benefit from non-random mobility models, rendering this comparison unfair.

For illustration purposes, after being imported to the simulator, the resulting trajectories were then converted to image files and are depicted in figure 5.9, representing the Gauss-Markov (figure 5.9a), Manhattan (figure 5.9b), Nomadic Community (figure 5.9c), Random Direction (figure 5.9d), Random Waypoint (figure 5.9e) and Random Street (figure 5.9f) Mobility Models. These different mobility models are entirely random, but each one has its own specificities. By using them the intent is to demonstrate that the DefeR paradigm is suitable in the most diverse scenarios.



(a) Gauss-Markov (b) Manhattan (c) Nomadic Community

(d) Random Direction (e) Random Waypoint (f) Random Street Map

Figure 5.9: Mobile Models' Trajectories

In order to evaluate the performance of the presented routing paradigm (DefeR in the presented figures), six scenarios incorporating different mobility models and an additional one with static nodes have been used. All these scenarios have the same area and number of nodes, using the trajectories defined by the BonnMotion tool, as previously detailed.

Since the nodes move freely across the entire scenario, their cluster association has to be changed. These changes are handled by the DefeR protocol which considers a total of 9 clusters, divided across the scenario, updating the nodes' CID when they move to a different cluster. As a result of nodes not being constrained to specific clusters, different node densities per cluster exist throughout the simulation time, while nodes follow their trajectories. These different densities impact the DefeR protocol negatively, since clusters are expected to be equally balanced throughout the simulation. However, the same conditions are maintained to all the used protocols.

Another important aspect that motivates and influences wireless multi-hop networks is the establishment of data flows between nodes. In the defined scenarios, 24 traffic flows with different destinations were generated in each run. From these flows, 50% were randomly chosen throughout the network, while the remaining traffic destinations were set to nodes within the cluster of the source node. By using this approach, both interactions within and outside clusters were assessed, providing a complete evaluation of the protocol's performance.

Each flow was defined with a constant bit rate of 8 packets of 4kbit per second (using UDP), representative of typical interactive gaming, simple file transfers or information exchange [ITU-T, 2003], which are all well suited applications for mobile Ad-hoc networks. The start time of each flow is randomly determined following a uniform distribution between 50 and 250 seconds of simulation time, being concluded by the end of the simulation.

5.3.2 Obtained Results

The purpose of this scenario is to clearly understand the impact of different mobility models on DefeR routing. Even though the presented concept handles inter-cluster routes in a completely different fashion from what other protocols propose, the following results show its efficiency in dealing with several distinct patterns of mobility.

5.3.2.1 Percentage of Losses

Figure 5.10 illustrates the percentage of losses registered by the routing protocols in all the defined mobility variations. In these, the DefeR protocol stands out by dint of having almost less than half of the losses than the remaining protocols. Conversely, the C-OLSR protocol registers the worst performance, having always more lost packets than the remaining protocols.

Regarding the Static scenario, the OLSR and C-OLSR protocols unexpectedly show worse delivery performance than in some mobile scenarios. This is a consequence of their inability to scale, as in the Static scenario more paths exist, whereas in the Manhattan scenario, for example, nodes are separated by the arrangement of the streets. However, the DefeR scheme is oblivious to the nodes' placement and has a similar performance in all the scenarios.

5.3.2.2 End-to-end Delay

The average end-to-end delay is presented in figure 5.11 for all the proposed mobility models. Being the static scenario the only exception, in the remaining scenarios the

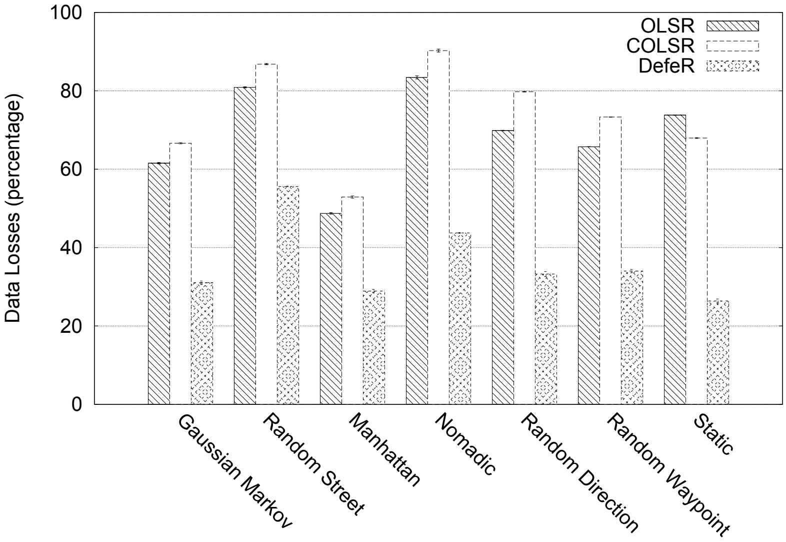


Figure 5.10: Average Percentage of Losses

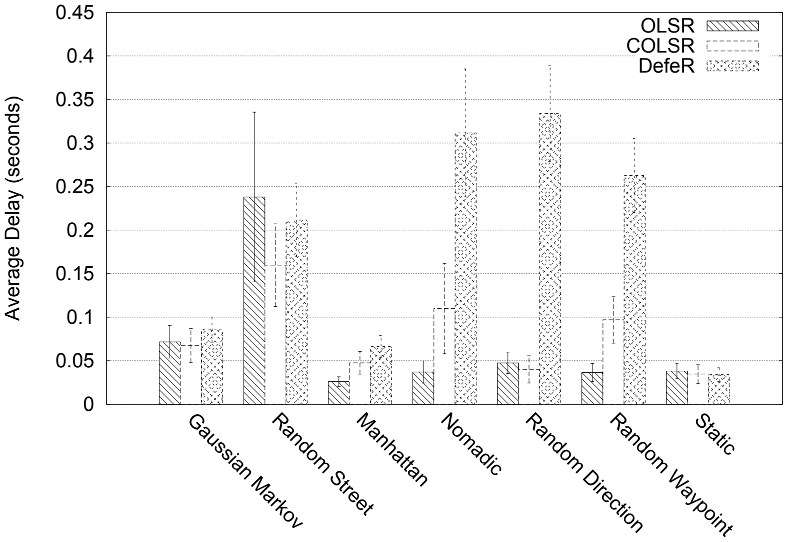


Figure 5.11: Average End-to-end Delay

DefeR protocol presents a higher delay. This aspect may not be desirable for certain types of traffic, such as voice, which are not well suited for Ad-hoc networks. The explanation for the higher delay registered by the DefeR protocol repeats itself – as a consequence of the additional traffic delivery achieved, an increased load of traffic is forwarded instead of being dropped.

In fact, while the end-to-end delay is typically a result of a higher path length, the used metrics will show that this is not the case. Specifically, when analysing the Manhattan scenario, where the highest hop count of the all mobile scenarios is registered for DefeR (see Figure 5.12), it has at the same time the lowest delay of all the mobile scenarios. This confirms that the approach taken by DefeR, which sometimes uses longer but more stable paths, registers less losses and is efficient, not introducing any delay by itself. The higher delay times are not registered in the Manhattan model, as the nodes follow well defined trajectories, where the additional delay overhead in the other mobile scenarios is due only to the repairing of broken paths, allowing the increased performance in traffic delivery registered by DefeR.

The self-restoring property of the DefeR protocol may occur in demanding situations where, due to the mobility phenomena, instead of dropping packets while routing tables change, packets are held and re-forwarded to the appropriate route. Thus, as previously concluded, a higher total delay average is expectable. Moreover, when bottlenecks are avoided due to load-balancing, the re-routing process may also introduce a slight delay. However, as the DefeR scheme is able to reach more challenging destinations than its competitors, the additional delay overhead is justifiable and still suitable for many different applications.

5.3.2.3 Path Length

Minimizing the path length is a typical target of routing protocols, with the purpose of reducing the network load and optimising packet delivery. However, due to network dynamics strongly influenced by node mobility, such a routing approach may reduce the protocols' traffic delivery as it disregards the stability of the chosen routes.

In most scenarios, the DefeR scheme is able to achieve a better path length than the remaining protocols while maintaining lower losses, as depicted in figure 5.12. Nevertheless, for the Manhattan, Random Waypoint and Static mobility models, the Deferred Routing Protocol has a slightly higher path length. This is a consequence of the scenarios' specificities and increased traffic performance of the DefeR, as it reaches more demanding destination nodes. The trade-off between path length and traffic efficiency,

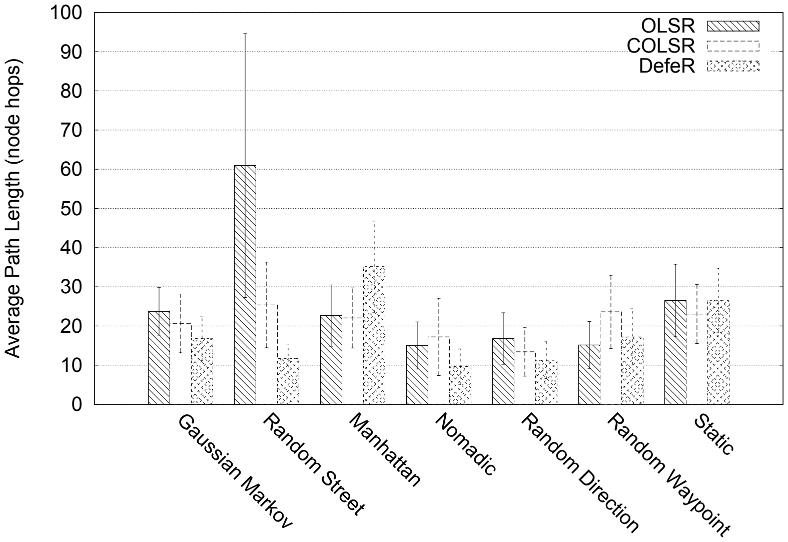


Figure 5.12: Average Number of Hops

in order to achieve an increased traffic performance, should be therefore regarded as an important feature in DefeR.

As a result of the randomly chosen destinations and of the wireless medium interactions, the confidence interval registered for the path length is higher than for other parameters. However, this interval is still similar to all the analysed routing protocols, validating the outcome of the parameter. The only observed exception worth of taking note occurs with the OLSR protocol in the Random Street Mobility Model. This mobility pattern is highly complex and it is clear that the OLSR protocol is not capable of dealing with the constant and close interactions between the moving nodes. In particular, the obtained standard deviation suggests that in certain occasions routing loops occur, drastically increasing the total number of hops.

5.3.2.4 Topology Changes Per Routing Table Calculation

When considering the scalability of a routing protocol, the stability of its routing tables is a key aspect on how it performs. The update of a routing table may be a costly procedure in terms of processing power and required energy, possibly leading to the creation and dissemination of additional routing messages, depleting the batteries of mobile devices faster than desirable.

Regarding this aspect, the OLSR protocol is clearly less scalable than the C-OLSR and DefeR protocols, which register a significantly smaller number of topology changes

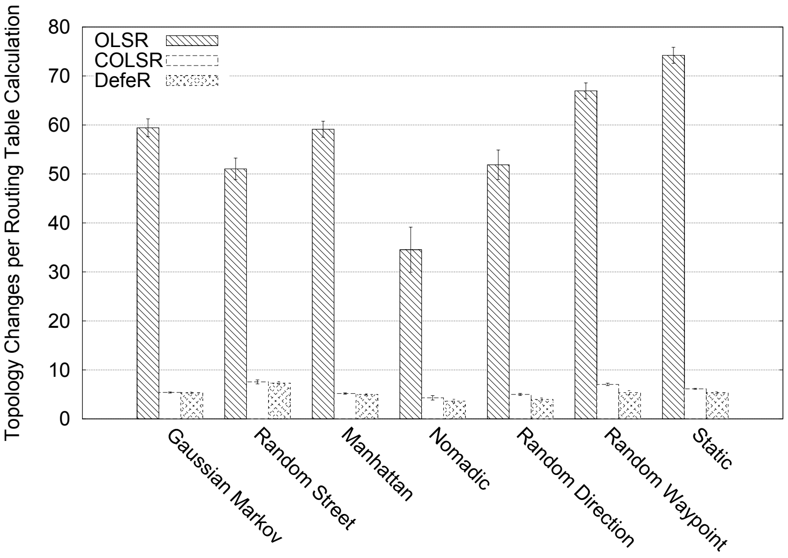


Figure 5.13: Topology Changes per Routing Table Calculation (AToCRT)

per routing table calculation, as shown in figure 5.13. In particular, the OLSR protocol has its worse performance in the static scenario. Such behaviour is a direct consequence of the wireless medium interactions of the nodes which are strongly connected in this scenario. In fact, in the mobile scenarios, where connectivity is often scarce, there is a clear reduction of the number of topology changes, suggesting once more that the OLSR protocol does not scale appropriately.

Considering the C-OLSR protocol, which benefits from the usage of clusters such as DefeR, it achieves a greater stability when compared with the standard OLSR. The number of topology changes per routing table calculation registered by this protocol is only slightly higher than the ones obtained from Deferred Routing. However, the overall performance of the C-OLSR protocol regarding traffic delivery suggests that its ability to timely register important topology changes is not appropriate, resulting in wrong or outdated routing paths. On the other hand, the DefeR awareness of the network is entirely different, detecting only the required amount of topology changes thus being more stable, leading to an increased traffic delivery performance, lower routing overhead and better energy efficiency.

5.3.2.5 Number of Forwards Per *TC* message

The three considered routing protocols rely on Topology Control routing messages to propagate the required information. These messages are issued periodically and

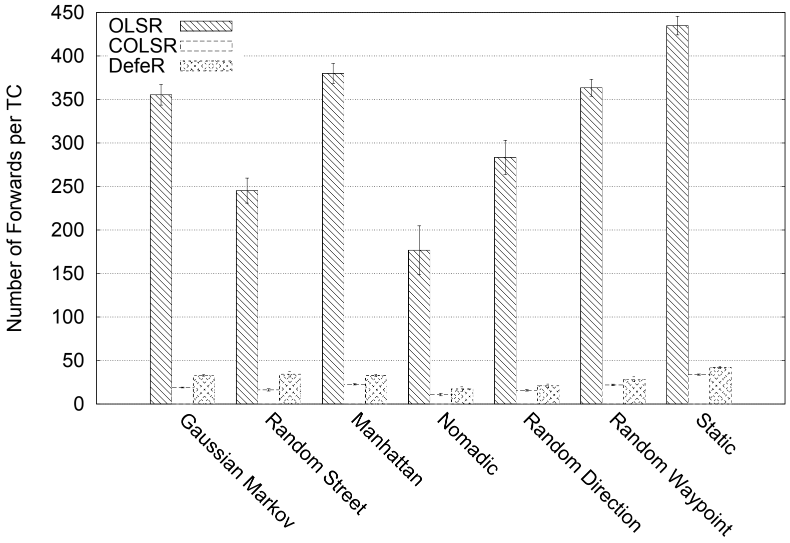


Figure 5.14: Number of Forwards per *TC* message

whenever a topology change is detected. Similarly to the previously analysed metric, the OLSR protocol is the worst performer, being at its lowest in the static scenario (Figure 5.14). The way that the OLSR routing protocol handles its routing information leads to an expensive propagation of its *TC* messages throughout the network.

On the other hand, the C-OLSR protocol requires less forwards per *TC* message than any of the other two protocols. Even though both C-OLSR and DefeR routing use the same clusters, it is clear that the usage of Cluster *HELLO* (C-*HELLO*) and Cluster Topology Control (C-*TC*) messages by the C-OLSR protocol is able to reduce the ratio between forwarded and sent *TC* messages. However, the amount of information and validity contained in these messages, also needs to be considered, as the previously analysed metrics reveal.

5.3.2.6 Control Traffic Overhead

Figure 5.15 shows the total amount of routing traffic sent by each routing protocol using the different mobility models. The OLSR protocol once again stands out for having the worst performance. The lack of a well defined network structure, which can be more easily obtained by using clusters, originates an increased overhead. While in the Static scenario this protocol has a bad performance, it is in the Random Street model that more routing traffic is sent.

While the clustered version of the OLSR protocol is able to provide an improve-

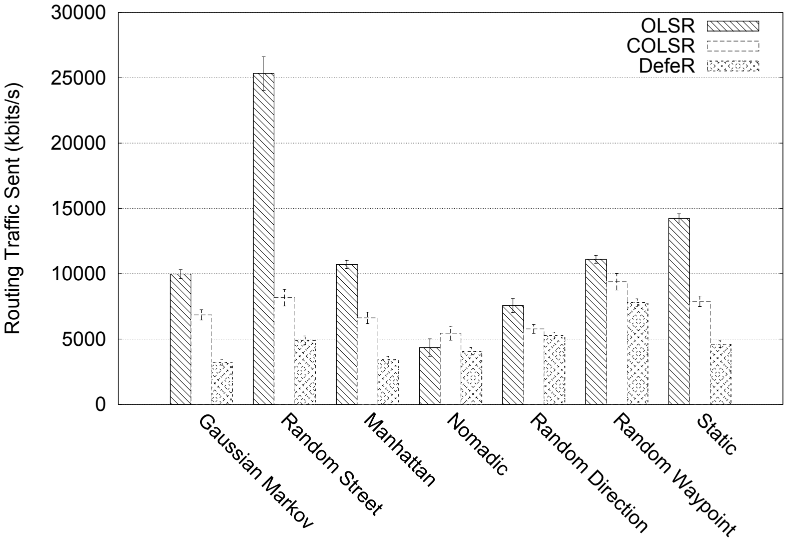


Figure 5.15: Sent Routing Traffic Overhead

ment regarding sent routing traffic, as seen before, it is not capable of maintaining this improvement in terms of data traffic delivery. On the other hand, the proposed DefeR protocol not only outperforms the C-OLSR by having less overhead, but it also outperforms the OLSR protocol in traffic delivery, registering less losses.

Since the sent routing messages may be forwarded through several nodes, Figure 5.16 presents the control traffic overhead received throughout the network. These results confirm the superiority of Deferred Routing in the handling of different mobility models, being in accordance with the verified ratio between sent and forwarded *TC* messages. Moreover, these results are obtained without guaranteeing a uniform density of nodes within clusters, which would benefit the performance of the DefeR protocol even further, as presented in the following scenarios.

5.4 Hierarchy Robustness

Flat un-clustered protocols such as OLSR, do not usually scale and even protocols with flat but clustered views of the network, such as C-OLSR, may suffer from costly overheads when handling routes between clusters, usually relying on cluster-heads. On the other hand, routing protocols that manage a network using a hierarchy for clustered nodes require a lower communication overhead in order to maintain their routes.

While hierarchical organisations may reduce the overall routing overhead, keeping

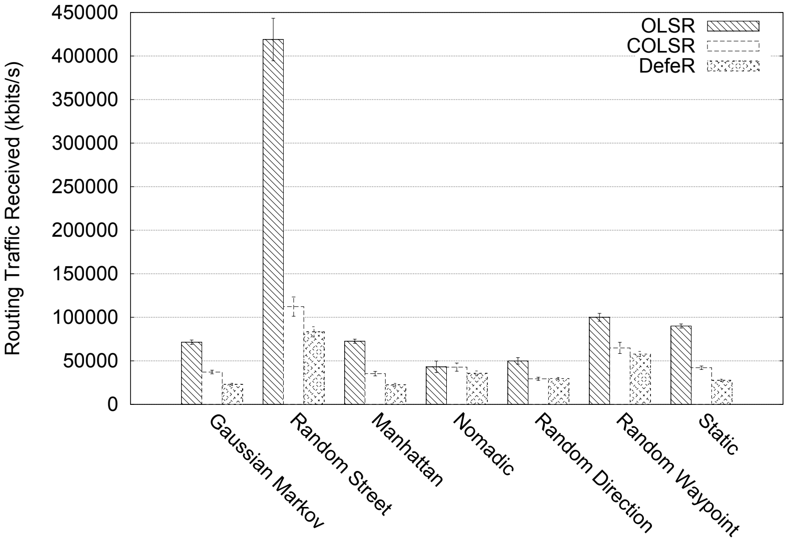


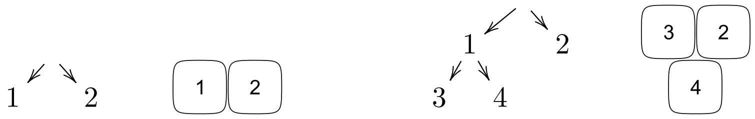
Figure 5.16: Received Routing Traffic Overhead

a hierarchy updated may introduce additional costs, resulting from required complex mechanisms such as dynamic addressing [Eriksson et al., 2007]. The hierarchy presented by DefeR aims at avoiding similar overheads, resorting to a virtual aggregation of the existing clusters. However, it is important to define a proper evaluation of this approach in order to validate it. For this purpose, different self-contained scenarios will be defined so that several hierarchies and hierarchical transitions are assessed in DefeR. By analysing in detail the overhead involved in each hierarchical transition, as well as the remaining defined metrics, the robustness of the DefeR hierarchy and its virtual views can be verified.

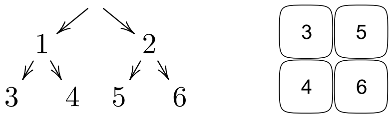
5.4.1 Description

Bearing in mind that the DefeR protocol is cluster-based and that it uses the OLSR protocol for intra-cluster routing, the differences between these two protocols will only be noticeable in a network with at least two clusters. Therefore, three different scenarios with 2, 3 and 4 clusters are presented. These scenarios will allow the evaluation of the impact of node mobility between clusters on the routing performance. In particular, since the DefeR protocol has a well defined hierarchy, a node moving between two different clusters will trigger a hierarchical transition, also allowing an assessment of the impact rendered by different level transitions.

In each of the defined scenarios a single node moves between two different clusters,



(a) 2 Clusters (b) 3 Clusters



(c) 4 Clusters

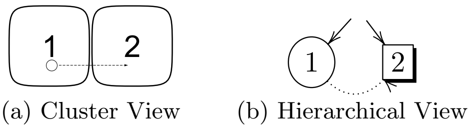
Figure 5.17: Network Hierarchy for a Different Number of Clusters

where each cluster has a total of 49 nodes distributed using a Poisson Point Process over the plan betoken by S and with intensity γ, along a square area of $500 \times 500m^2$. It starts off, by being stationary for 250 seconds and after that it moves in the direction of a destination cluster at a speed of $12km/h$, similarly to travelling by bicycle or walking [3GPP, 2008], travelling a total distance of 600 meters. Since the purpose of this work is to evaluate the performance of the DefeR protocol, the moving node was also the destination for a constant bit rate using UDP flow of 32 $kbit/s$ (8 packets per second) and all the remaining nodes were static. Once again, this type of traffic flows is representative of typical interactive gaming, simple file transfers or information exchange [ITU-T, 2003], which are appropriate applications for mobile Ad-hoc networks.

By specifying a moving node which is part of a traffic flow, while keeping all the other nodes static, a more accurate understanding of the impact of different level transitions will be obtained. This will reveal how efficient a routing protocol is when updating its existing routes, allowing not only the analysis of its scalability, but also overall routing performance regarding delivered traffic. Moreover, it is important not to introduce any other mobile nodes, as it would likely reduce the connectivity between nodes, influencing the intended scalability analysis.

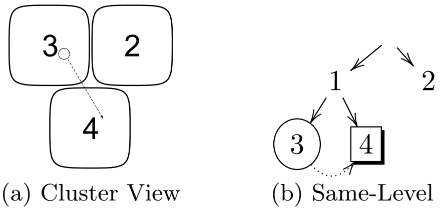
5.4.1.1 Two-Cluster Network

The most straightforward hierarchy in DefeR is found in a network with two clusters. In this hierarchy the only possible transitions will occur in the same hierarchical level (0 Level Transition), when nodes move from the cluster with CID 1 to CID 2 and vice-versa. Figure 5.18 shows the configuration of such network, where the fully circled CID



(a) Cluster View (b) Hierarchical View

Figure 5.18: Same-Level Transition Example for 2 Clusters



(a) Cluster View (b) Same-Level

Figure 5.19: Same-Level Transition Example for 3 Clusters

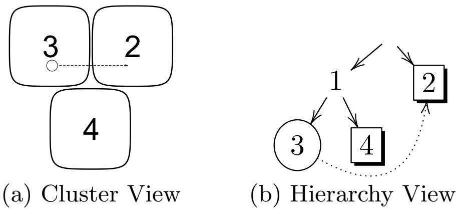
and the end of the arrow respectively correspond to the origin and destination clusters. Since there are two possible transitions, this scenario was evaluated twice, one where the node moves from cluster 1 to 2 and the other from cluster 2 to 1.

In this scenario all the clusters are affected by any occurring transition, since they are sibling clusters. However, in a scenario with more clusters this will not always occur, as shown in the three-cluster network.

5.4.1.2 Three-Cluster Network

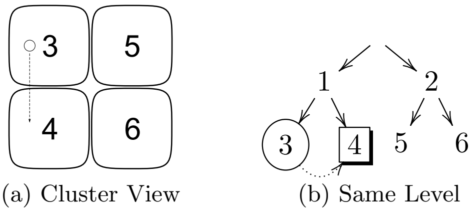
As the number of clusters increases in a network, so does the number of possible transitions in the DefeR hierarchy. In a network with three clusters, in addition to Same-Level transitions between clusters 3 and 4, there is also a One-Level transition between CIDs 3 or 4 and 2. Figures 5.19 and 5.20 depict some of these transitions, when a node moves from cluster 3 to 4 and from cluster 3 to 2. Moreover, in order to better illustrate the protocol's behaviour, in these figures the clusters which are affected by each transition, in addition to the source and destination, are depicted in a shaded box. This highlights the existing aggregated views used by DefeR, such that for Same-Level transitions nothing is changed for nodes in cluster 2.

Since there are three clusters in this scenario, six different transitions may occur - from cluster 3 to 4 and 2, from cluster 4 to 3 and 2 and finally from cluster 2 to 3 and 4. Similarly to the previous scenario, all these transitions were individually simulated, leading to four One-Level transitions and two Same-Level transitions.



(a) Cluster View (b) Hierarchy View

Figure 5.20: One-Level Transition Example for 3 Clusters



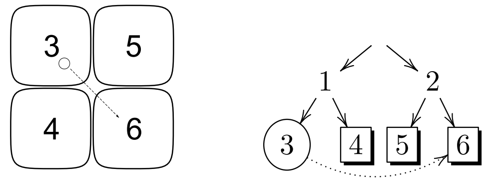
(a) Cluster View (b) Same Level

Figure 5.21: Same-Level Transition Example for 4 Clusters

5.4.1.3 Four-Cluster Network

In a network with a total of 4 clusters, Two-Level transitions may occur when a node changes its cluster association to a cluster in a different branch of the network. Even though Same-Level transitions still exist (Figure 5.21), One-Level transitions will never occur, since a node moving to a non-sibling cluster will have to go one level higher into the hierarchy and then lower to a leaf cluster. In Figure 5.22 a Two-Level transition is presented, where a node from cluster 3 moves to cluster 6, affecting not only the source and destination clusters, but also their sibling brothers. This transition represents the worst case scenario, since Same-Level transitions only affect 2 clusters. This reduced impact is related with the adoption of the Deferred Routing concept, where in a network with 4 clusters, each node perceives only 2 clusters. In fact, for a network with C clusters, at any given point a node recognizes at most $\lceil \log_2 C \rceil$, which also corresponds to the number of levels in the hierarchy. Thus, for an l-level transition in a network with C clusters, knowing that $l \leq \log_2 C$, the maximum number of clusters affected by a transition is $2 + l$.

Once again, since several transitions among the four different clusters exist (12 possibilities), this scenario was evaluated individually for each transition, leading to a total of 4 Same-Level transitions and 8 Two-Level transitions.



(a) Cluster View (b) Hierarchy View

Figure 5.22: Two-Level Transition Example for 4 Clusters

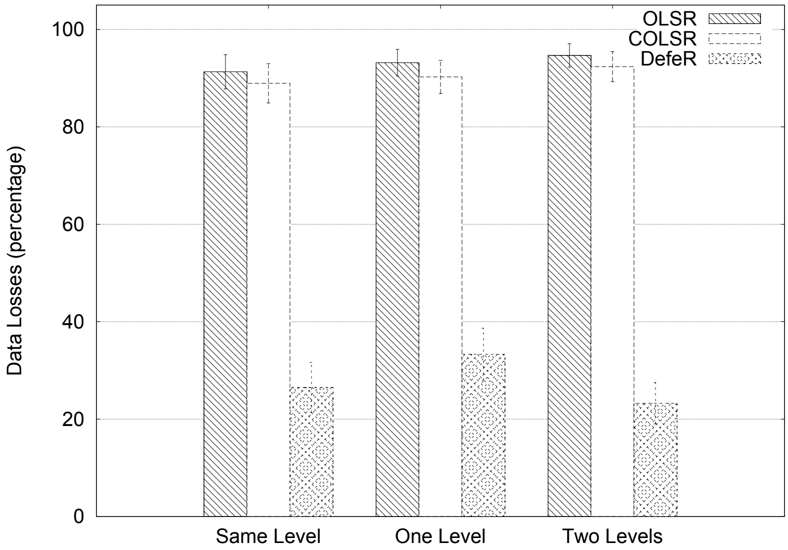


Figure 5.23: Average Percentage of Losses

5.4.2 Obtained Results

This scenario offers a more controlled environment, where most of the randomness introduced by the mobility of nodes was removed, allowing a better understanding of how routing protocols handle their paths. In particular, a more detailed analysis of the innovative hierarchy presented by Deferred Routing can be achieved.

The presented results comprise the three different hierarchical transitions that occur with 2, 3 and 4 clusters. The Same-Level transition results aggregate the information obtained for the three network sizes, while One-Level and Two-Level transitions represent respectively networks with tree and four clusters.

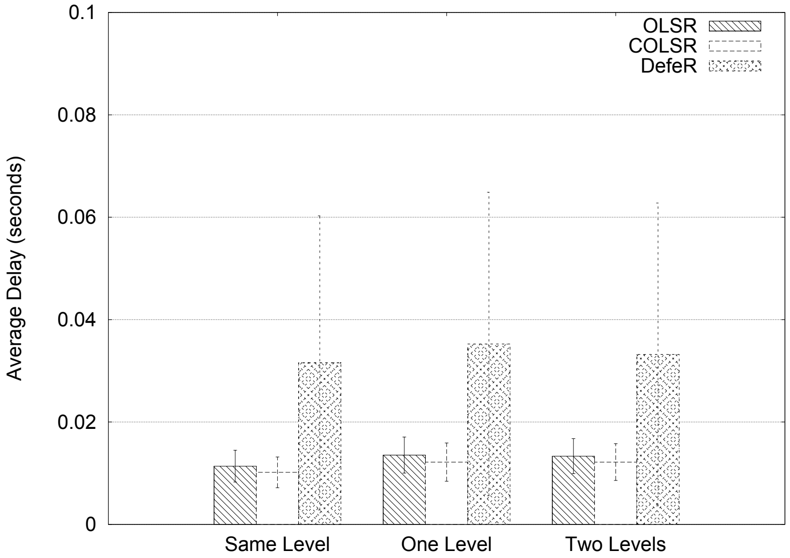


Figure 5.24: Average End-to-end Delay

5.4.2.1 Percentage of Losses

In some hierarchical routing protocols the route update of a moving node can be more problematic if a large "distance" is travelled within the used hierarchy. Due to the Deferred paradigm used by the proposed protocol, this does not occur. Figure 5.23 depicts the percentage of losses of a node moving between two clusters. In the defined scenario this cluster change may issue different hierarchical transitions, which are distinguished by the DefeR protocol. This protocol clearly outperforms the remaining two protocols, which do not have any hierarchy but fail to efficiently deliver most of the data packets to a moving destination.

Focusing more on the DefeR performance, less packets were delivered to the One Level transition. This results from the fact that the specific transition only occurs in the network with 3 clusters, which has a slightly different configuration. Even though, the usage of square clusters limits the contact area between the three clusters, the same conditions were kept to all the evaluated protocols.

5.4.2.2 End-to-end Delay

In Figure 5.24, the end-to-end delay registered by each protocol is presented. While the DefeR scheme presents a higher delay, as observed in other scenarios, it results yet again from the improved traffic delivery performed and the self-healing property between different clusters.

Regarding the two other protocols, the C-OLSR protocol not only has a smaller delay than its non-clustered version, but it also registers less losses. Comparing with the other scenarios analysed so far, this is an important result for the C-OLSR protocol, as it is the first time that it outperforms the OLSR protocol. A routing protocol must not only be scalable but also capable of efficiently delivering the required routing packets. Such results indicate that the C-OLSR protocol has issues with handling mobility, since this scenario is considerably static.

5.4.2.3 Path Length

The characteristics of the defined scenario should render a fairly constant path length, as the moving node follows a pre-defined movement. However, due to the different management of routing information, as well the dynamics of the wireless links, this is not always the case. In Figure 5.25 the number of hops between source and destination, with the exception of the C-OLSR protocol in the One-Level transition, is similar for the three protocols. A possible explanation for this abnormal behaviour is related with the different topology of the 3 cluster network. However, the particular behaviour of the C-OLSR protocol is related with its distributed mechanisms to propagate routing changes among different clusters, leading to the momentary creation routing loops. Despite this fact, the C-OLSR still manages to deliver more data packets than the OLSR protocol.

5.4.2.4 Topology Changes per Routing Table Calculation

The number of topology changes in each routing table calculation reflects the stability of the overall routing process of a protocol. Figure 5.26 depicts this parameter, showing that the DefeR protocol is more stable regardless of the taken transition. Such results further prove the advantages of the used hierarchy, being more stable than its competitors.

In the particular case of One-Level transitions, for the tree cluster scenario, the C-OLSR protocol repeats its unexpected behaviour and has a significant drop in the number of topology changes. This observation helps to better understand the previously analysed increase in the number of hops, re-enforcing the likeliness of temporary routing loops.

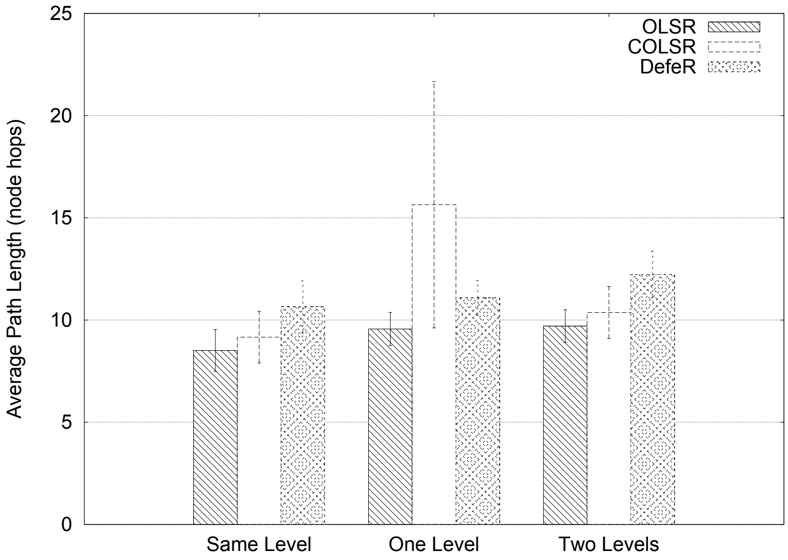


Figure 5.25: Average Number of Hops

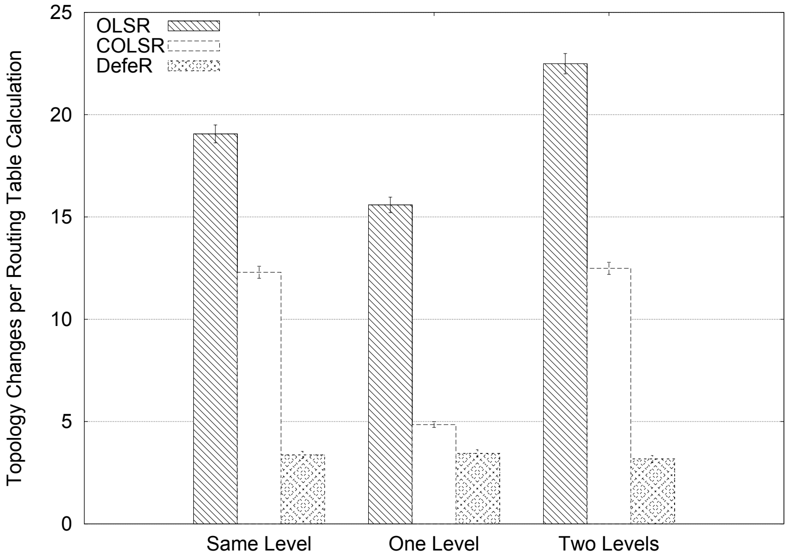


Figure 5.26: Topology Changes per Routing Table Calculation (AToCRT)

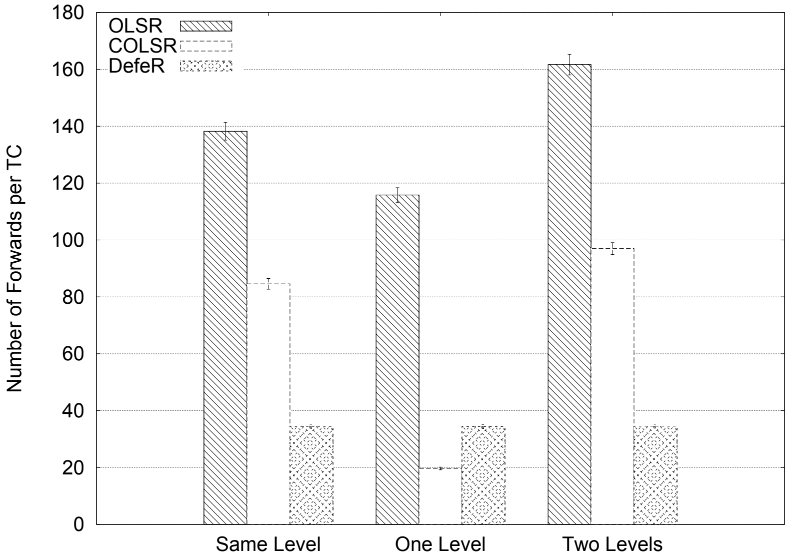


Figure 5.27: Number of Forwards per *TC* message

5.4.2.5 Number of Forwards Per *TC* message

As previously stated, a protocol using OLSR should minimize the average number of forwards per *TC* message, avoiding an expensive flooding of routing data. As it is shown in Figure 5.27, the pure OLSR performs worse than the other two protocols. In particular, the DefeR protocol has a constant number of forwards for all the transitions, revealing yet again the scalable properties of the DefeR protocol.

Same-Level transitions occur in 2, 3 and 4 cluster networks, while One-Level transitions are only possible in the 3 cluster network. As a result, it is expectable that the OLSR protocol has less forwards in the One-Level transition. However, regarding the C-OLSR protocol, an anomaly occurs in these transitions, as the protocol should, similarly to the DefeR scheme, maintain a constant number of forwards per sent *TC*.

After analysing in detail the behaviour of the C-OLSR protocol, it was found that due its instability and for cluster organization purposes, additional *TC* messages were created, lowering the average number of forwards. This reflects an issue with the usage of Cluster MPRs by C-OLSR. While a certain number of MPRs are expected for a well behaved network, in reality, and even through simulation, the volatility of the wireless links renders the task extremely complex. This dependency on the MPR election algorithm makes C-OLSR vulnerable to the typical changes in Mobile Ad-hoc Networks.

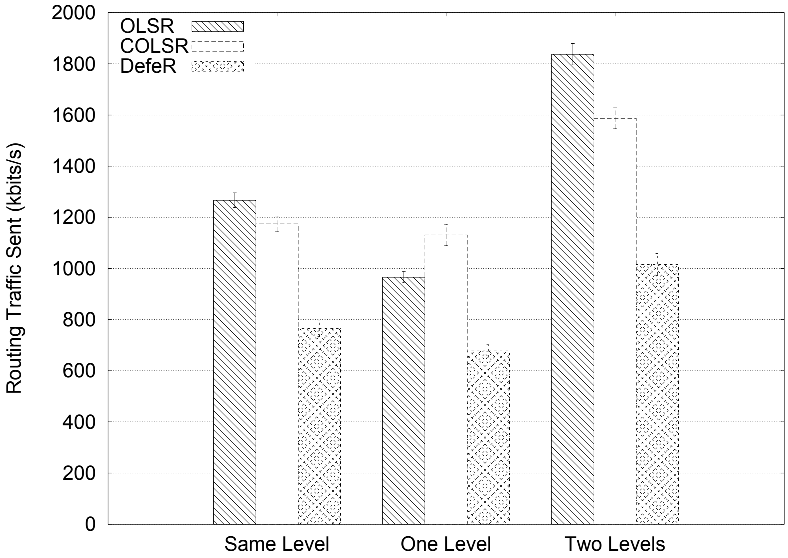


Figure 5.28: Sent Routing Traffic Overhead

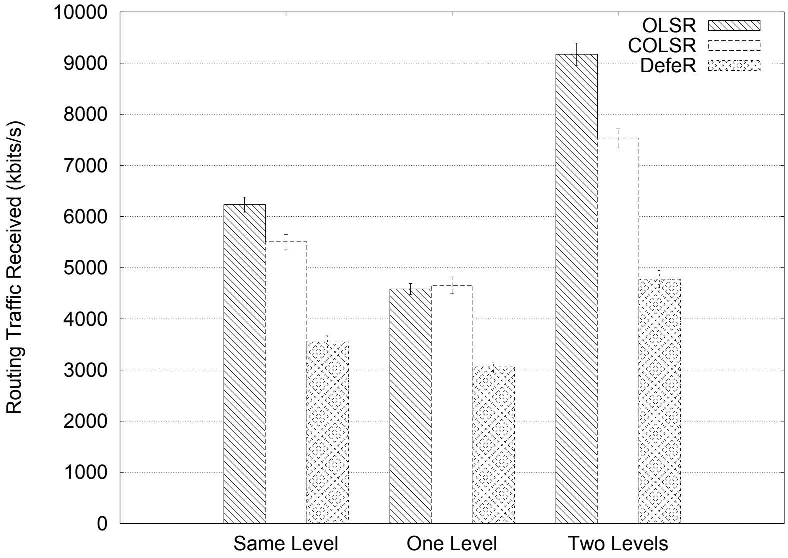


Figure 5.29: Received Routing Traffic Overhead

5.4.2.6 Control Traffic Overhead

After showing that the hierarchy used by the DefeR routing protocol is not influenced by the different hierarchical transitions that might occur, the analysis of the control traffic will reinforce the scalable properties of the protocol. Figures 5.28 and 5.29

corroborate what has been found so far. The DefeR is clearly more lightweight than any of its competitors.

When comparing the C-OLSR and OLSR protocols it is not so obvious which one performs best, as in certain situations the C-OLSR protocol reveals unexpected results. Nonetheless, in this scenario the clustered version of the OLSR protocol generally has a smaller overhead and it manages to have a better traffic delivery performance.

5.5 Realistic Scenario

Regarding the defined Deferred Routing characteristics, a thorough evaluation of its performance should consider aspects such as the existence of communities. The protocol should take advantage of existing contexts and maintain itself scalable. The performed assessment must involve the evaluation of a large scale network with dynamic characteristics, determining the concept's ability to handle different mobility patterns and network characteristics, without increasing the routing overhead.

5.5.1 Description

In order to evaluate the performance of the presented Deferred Routing paradigm, a complex scenario with 3 distinct mobility behaviours, following the Random Waypoint Mobility Model, and another one without mobility, was defined. The Random Waypoint (RWP) mobility model has been extensively used in literature and, while some works have shown some disadvantages in using it [Shakeri and Hosseini, 2010], it is still widely used in recent works [Zhou and Ying, 2010; Qin and Zimmermann, 2010], as it provides a generic form of mobility without being tied to particular applications. Moreover, the random waypoint implemented in OPNET Modeler Wireless Simulator guarantees a uniform distribution of the x and y coordinates within the boundaries of the scenario, as well as different initial states (Pause or Moving) for each node, ensuring a "steady-state" distribution of the random waypoint model [Navidi and Camp, 2004]. Different node densities per cluster, as well as different traffic flows throughout the simulation time, were defined, as presented in this description.

Similarly to the other presented scenarios, in this scenario the considered wireless nodes follow the IEEE 802.11g standard [Ortiz, 2009]. However, since an indoor-based simulation is being defined, the wireless nodes have a maximum range of 25 meters (Transmit Power of $2.27e^{-5}W$), which should correspond to a realistic range within a building. Moreover, due to the accurate radio model implemented by default in the

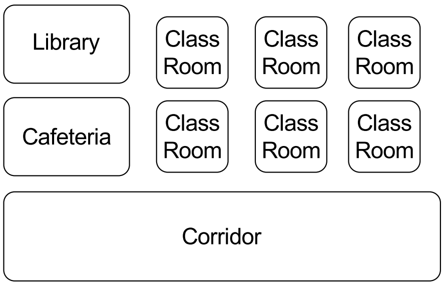


Figure 5.30: Defined Scenarios Representation

Table 5.1: Clusters' Description

Room Type	Area (m x m)	Mobility Characteristics	
		Speed (ms^{-1})	Pause Time (s)
Class Room	20 x 20	static	static
Cafeteria	22 x 35	0.2 - 1.0	60 - 120
Library	22 x 35	0.2 - 1.0	180 - 600
Corridor	25 x 120	0.2 - 1.0	10 - 60

Table 5.2: Flows' Characteristics

Flow	Simulation Time	Packet Size	Inter-arrival Time
1 (6x)	start:160;stop:280 sec	4kb	2 - 6 sec
2 (6x)	start:280;stop:520 sec	4kb	2 - 6 sec
3 (6x)	start:520;stop:760 sec	exponential(1kb)	exponential(1)

OPNET Simulator, asymmetric links or even unidirectional links may occur, as well as channel errors and multi-path interferences, among others.

The chosen scenario is intended to be dynamic and it has been inspired in a University Department, representing one floor, where different clusters exist due to different rooms (such as class rooms, a library, a cafeteria and a big corridor connecting all the rooms). All the clusters have 49 nodes and their specifications are described in Table 5.1, where the speed and pause time intervals follow a uniform distribution. There are 6 Class Rooms with the same characteristics, summing up, with the other rooms, to a total of 9 clusters. Traffic flows are the same for all the rooms, with 6 UDP source nodes which send data to nodes in their own cluster, and an additional source node which randomly chooses a destination node in the network, exploring the sense of community [Kim and Yoo, 2007]. Every source node creates 3 different flows during the simulation, representing simple file transfers, interactive gaming and information exchange [ITU-T, 2003], as described in Table 5.2. All the speed and pause values

Table 5.3: Traffic Delivery Perfomance

Metrics	OLSR	C-OLSR	DefeR
Percentage of Losses	6.29	5.82	1.26
End-to-end Delay (s)	0.023	0.002	0.002

presented within an interval are randomly chosen following a uniform distribution.

5.5.2 Obtained Results

In this scenario, the most significant contribution is the attempt to recreate a realistic scenario with dynamic traffic flows and variations of the same mobility pattern.

5.5.2.1 Percentage of Losses

Due to the proximity between the source and destination nodes in the defined traffic flows, the amount of registered losses is very small for any of the three protocols, as seen in Table 5.3. The DefeR protocol still registers the best performance with nearly no losses, though the remaining protocols also have a satisfying traffic delivery. It is important to note though, that no node moves from its originating cluster, reducing the number of topology changes that strongly influences the OLSR and C-OLSR protocols in the previously analysed scenarios.

5.5.2.2 End-to-end Delay

Similarly to the obtained percentage of losses, and for the same reasons, a low end-to-end delay was registered for the three protocols, presented in the same table. However, the OLSR protocol shows how its lack of scalability influences traffic paths, suggesting that less up-to-date paths are used.

5.5.2.3 Path Length

The properties of the defined scenario are once again clear, portrayed the Routing Performance results, presented in Table 5.4, highlighting the locality of the used traffic flows with a low number of hops. Despite the increased delay, the OLSR protocol performs as well as any of the other routing protocols, proving that not always does the path length directly influence the delay found in a route.

Table 5.4: Routing Performance

Metrics	OLSR	C-OLSR	DefeR
Path Length (hops)	1.424	1.427	1.424
Topology Changes per Routing Table Calculation (AToCRT)	47.74	1.99	1.06
Number of Forwards per *TC*	195.29	6.66	14.84

5.5.2.4 Topology Changes per Routing Table Calculation

The fairly stable behaviour of the wireless nodes is also registered in Table 5.4, which shows that only a small number of topology changes are required in each routing table recalculation. The DefeR protocol is slightly more stable than the C-OLSR protocol, but it is the OLSR protocol that requires a higher number of topology changes. The lack of containment of the routing information leads to a higher instability of the protocol as otherwise seen in other scenarios.

5.5.2.5 Number of Forwards Per *TC* message

The average of number times that each *TC* is forwarded by each protocol is also present in the Routing Performance table. The lack of scalability of the OLSR protocol is evident, requiring at least 13 times more forwards than its competitors. Regarding this aspect, the C-OLSR protocol was the most efficient, outperforming the DefeR scheme. However, the amount of information contained in each routing message has a significant importance in the overall routing traffic overhead, as seen next.

5.5.2.6 Control Traffic Overhead

The amount of control traffic overhead, introduced in a network by a routing protocol, strongly influences the overall network's performance. In particular, when too much routing traffic exists, interferences may lead to an increased number of losses and total energy consumption by each node.

By not requiring additional routing messages, only but some minor changes to a proactive routing protocol, the Deferred Routing approach is able to send significantly less control traffic overhead than other routing protocols. This is clear in Figure 5.31, which also reveals that the C-OLSR protocol is not as scalable as expected. While this

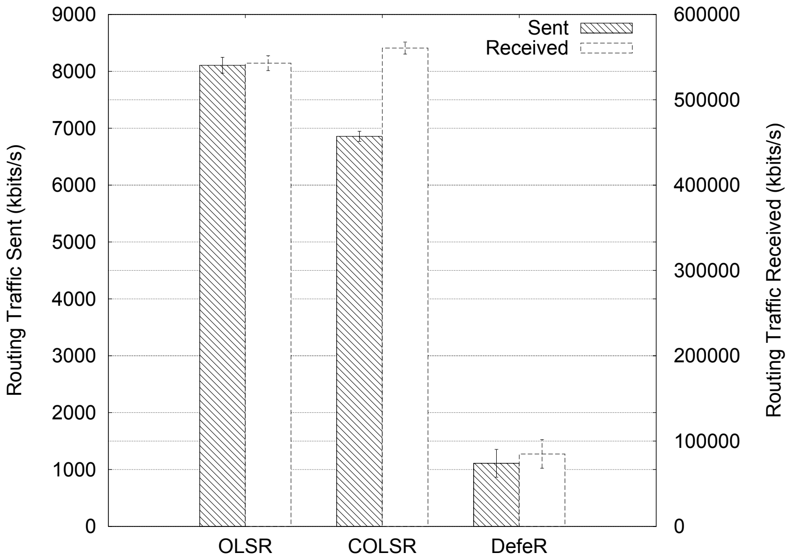


Figure 5.31: Sent and Received Routing Traffic Overhead

protocol issues less forwards per *TC* message, it does require the exchange of cluster *HELLO* and cluster *TC* messages.

The impact of the specific routing messages used by the C-OLSR protocol is even more pronounced in received routing traffic. In Figure 5.31 it is also possible to see that the overall received control traffic is even higher than the OLSR protocol. On the other hand, a network using the DefeR protocol receives 5 times less routing traffic, showing that it can be more easily used by wireless mobile nodes with lower resources.

5.6 Summary

The versatility of Mobile Ad-hoc Networks makes them suitable for a wide range of scenarios. Moreover, the dynamic nature of the wireless medium involves a large set of variables which influence the behaviour of these networks. In this Chapter, several parameters were considered for the assessment of the proposed routing protocol, as well as 4 different scenarios with different characteristics. The protocols' scalability was tested by using a scenario with different size networks, while the effects of different mobile patters were assessed in a scenario using seven different mobility models. In addition to the more conventional scenarios, the hierarchical nature of the proposed protocol was evaluated in a scenario with controlled hierarchical transitions and a useful scenario where MANETs can be helpful for ubiquitous communication, were considered.

In the performance evaluation of the DefeR protocol, the OLSR and C-OLSR protocols were also considered for comparison purposes. Between these protocols, the DefeR scheme revealed that it is able to deliver more traffic than its competitors, even though it introduces some delay with the path repairing mechanism. Despite having a slightly higher delay, this protocol is still useful for many possible applications, being more stable regarding the number of required routing operations and having an overall smaller overhead.

The thorough evaluation obtained from all the defined scenarios and complete simulation environment, provided a complete understanding of the protocols' performance, which could not be acquired from simplistic wireless models. Moreover, the empirical results from the simulated scenarios also revealed that the DefeR is more lightweight than the remaining protocols, always completing the performed simulations faster and with less resources than the remaining protocols.

Resulting from the analysis of the impact of moving nodes in hierarchy-based protocols, two publications entitled "Impact Analysis of Hierarchical Transitions in Multi-hop Clustered Networks" and "Scalability and Routing Performance of Future Autonomous Networks" have been respectively included in the proceedings of the Network Operations and Management Symposium (NOMS), 2012, and of the 3rd IEEE International Workshop on SmArt COmmunications in NEtwork Technologies (SaCoNeT-III), 2012.

Chapter 6

Conclusion and Future Work

The usage of wireless multi-hop networks is undeniably important for a future world where thousands of wireless capable devices are expected to be connected. Despite the existing work on this topic, open issues such as routing scalability still exist. Throughout this thesis, improved routing mechanisms, as well as scalability techniques for Mobile Ad-hoc Networks, have been described. In this chapter the addressed problems are revisited, highlighting the major contributions of this work, setting the direction for future improvements and additional contributions.

6.1 Synthesis of the Thesis

The subject of routing in Mobile Ad-hoc Networks has long been addressed by several authors, who proposed many different approaches. There is in fact an Internet Engineering Task Force (IETF) working group entirely dedicated to this topic, the Mobile Ad-hoc Network (MANET) working group, guaranteeing two main routing classes of routing – proactive and reactive. These and other approaches are analysed in Chapter 2, where the most relevant features and open issues are identified. Such analysis motivated the definition of an alternative routing paradigm.

Taking into consideration the nature of MANETs, the Deferred Routing protocol is presented in Chapter 3. Aiming at handling the increasing demand of wireless capable devices, this protocol offers highly scalable mechanisms that are also able to cope with the mobility of such devices. These new mechanisms are explained in detail and all the required changes to adapt any link-state routing protocol are also specified, tackling Research Question number 1, identified in the introduction chapter.

Awareness of link quality in a wireless network is another aspect capable of improving the performance of a routing protocol in MANETs. In particular, due to usage of clusters, the Deferred Routing (DefeR) protocol takes advantage of a link quality estimator for gateway selection, modelled in Chapter 4. The defined estimator resulted from traces obtained from both real and simulation-based environments, using kernel-based methods for its calculation. The issues raised by Research Question number 2 are handled in this Chapter.

In order to validate the contribution of the presented routing protocol an extensive set of different scenarios is defined in Chapter 5, where the protocol's performance is assessed regarding its scalability, stability and traffic delivery capabilities –answering Research Question number 3. The provided results were obtained from several simulations, taking into account the dynamic characteristics of the wireless link and different mobility patterns, which significantly influence MANETs.

6.2 Contributions

The contributions provided by this work have already been briefly discussed in the introduction chapter, laying the foundations for the remaining of the thesis. In this section these contributions are revisited and further detailed, taking into consideration the analysed aspects of existing protocols regarding scalable routing, and also the performance obtained by the DefeR protocol.

A new routing paradigm The presented Deferred Routing Protocol introduces an entirely different perspective towards scalable routing in highly dynamic wireless networks, handling existing routing information in different perspectives of the network, without requiring additional routing messages. Furthermore, the DefeR protocol is able to efficiently deal with mobility disruptions in real-time, holding and re-routing packets towards the most suitable Gateways as data packets reach their destination. Moreover, by resorting to a hierarchical organisation of clusters, the DefeR scheme is able to provide different network perspectives to each node, according to its own position in the hierarchy. This approach allows routing information to be appropriately aggregated, limiting the effect of macro-mobility which strongly affects other cluster-based protocols. In fact, the performed evaluation shows that this hierarchy is able to efficiently handle different cluster transitions without disrupting the protocol's performance. Typically, cluster-based routing protocols discard packets arriving from foreign clusters, however the proposing routing scheme handles these routing packets differently, overhearing existing information and reducing propagation overheads. The packet is firstly analysed and all the relevant information is retrieved before discarding the packet, with no issued forwards. This allows routing information to autonomously be propagated through the network without the need for additional routing messages, increasing the performance of the DefeR protocol, which may only suffer from a slight increase on delay when changes occur.

Deferred Routing The name of the presented routing protocol comes from the approach by which data packets are forwarded. Instead of maintaining complete end-to-end routing paths, which require more routing information and are more prone to disruption, the DefeR protocol is only concerned with the forwarding of packets across the appropriate clusters. Knowing the destination's containing cluster, and by consulting the maintained hierarchy, the DefeR protocol focuses its forwarding decisions on the choice of the most suitable Gateways (GWs) that minimize the total number of traversed clusters.

Efficient Gateway Selection Link quality estimation provides the DefeR protocol an optimised way to select the most suitable GWs, involved in the packet forwarding process. Instead of minimising only the number of cluster hops from source to destination, the overall quality of a gateway's link is considered. This parameter is obtained from a Link-quality Model, defined by using Kernel-based Methods for Regression which, through the analysis of the time interval between periodically sent *HELLO* messages, estimates the performance of a wireless link. The defined model is completely assumption free and the obtained results show its robustness with both simulation-based and realistic traces.

Performance Evaluation Even though the main objective of the presented performance evaluation is the assessment of the Deferred Routing characteristics, this work provides a comprehensive and thorough evaluation that can be used to assess any routing protocol for MANETs. Several scenarios with different purposes are defined, scrutinising different aspects of the performance of a routing protocol, such as its scalability, regarding both the number of nodes and an increasing of flows, as well as its resilience to several distinct mobility patterns. Moreover, the presented evaluation further analyses the impact of the transition of a mobile node between different hierarchical levels of a protocol. Finally, a less academic scenario is also defined, using a topology likely to be found on a daily basis, motivating the usage of MANETs in alternative, but realistic, scenarios.

6.3 Future Work

The importance of Mobile Ad-hoc Networks in a near future has been discussed throughout this thesis, unveiling a new routing paradigm capable of handling the dynamic nature of these networks in a scalable fashion. The provided evaluation revealed that the Deferred Routings protocol is capable of outperforming existing solutions, motivating its usage in a wide range of scenarios. However, despite the optimistic results, there are still unresolved aspects that should be addressed in the future.

Regarding the performance of the DefeR approach, as with other cluster-based protocols, it results from the usage of clustered wireless networks, which allows routing information to be contained within limited contexts. Even though several other routing protocols also rely on this aspect, and considering that many clustering algorithms have already been proposed, there is no clustering scheme that truly reflects the concept behind DefeR. The modification of an existing clustering approach or even the definition of a new one should be addressed in a future work, taking advantage of the increasing availability of contextual information provided by sensors, databases, mobility and traffic patterns or even by the users themselves. An analysis of the user's own experience can also be considered in order to further improve the performance of the Deferred Routing scheme.

Following the purpose of better exploring the paradigm proposed by Deferred Routing, a further improvement should consider the real interactions between users, combining the routing process with the social context of users and their favourite applications. This would allow a more focused and even more lightweight implementation of DefeR, which could be developed for standard wireless capable devices such as smart-phones, gaming or media stations. Moreover, in the future Internet of Things, where increased Machine-to-Machine interactions will exist [Attwood et al., 2011], overlays of robust and scalable mobile and mesh Ad-hoc networks will further motivate the continuation of the presented work.

Future work for the DefeR protocol should also consider a more environmentally friendly approach. Bearing this in mind, works have already been proposed towards "green computing", defining incentives and mechanisms for energy accounting [Siekkinen et al., 2012]. The improvement of the overall energy efficiency of wireless Ad-hoc networks can be achieved by enhancing routing decisions, for instance with the selection of the most appropriate GW nodes in DefeR, according to their wireless characteristics [Chorppath and Alpcan, 2011]. Nowadays people are more and more focused on the creation of a sustainable society, where energy efficiency is becoming an increasing concern and where wireless technologies play an important part.

References

[3GPP, 2008] 3GPP. Spatial channel model for multiple input multiple output (mimo) simulations. technical specification group radio access network. Technical Report 7, 3rd Generation Partnership Project, 2008. 76, 86, 94, 104

[Adjih et al., 2004] Cédric Adjih, Emmanuel Baccelli, Thomas Heide Clausen, Philippe Jacquet, and Georgios Rodolakis. Fish eye olsr scaling properties. *IEEE Journal of Communications and Networks*, dec 2004. 55

[AlAamri et al., 2012] Huda AlAamri, Mehran Abolhasan, Daniel Franklin, and Justin Lipman. Optimised relay selection for route discovery in reactive routing. *Ad Hoc Networks*, (0):–, 2012. ISSN 1570-8705. doi: 10.1016/j.adhoc.2012.04.008. 20

[Alotaibi and Mukherjee, 2012] Eiman Alotaibi and Biswanath Mukherjee. A survey on routing algorithms for wireless ad-hoc and mesh networks. *Computer Networks*, 56(2):940 – 965, 2012. ISSN 1389-1286. doi: 10.1016/j.comnet.2011.10.011. 13

[Anastasi et al., 2003] Giuseppe Anastasi, Eleonora Borgia, Marco Conti, and Enrico Gregori. Ieee 802.11 ad hoc networks: performance measurements. In *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on*, pages 758 – 763, May 2003. doi: 10.1109/ICDCSW.2003.1203643. 68

[Anastasi et al., 2004] Giuseppe Anastasi, Eleonora Borgia, Marco Conti, and Enrico Gregori. Wi-fi in ad hoc mode: A measurement study. *Pervasive Computing and Communications, IEEE International Conference on*, page 145, 2004. doi: doi.ieeecomputersociety.org/10.1109/PERCOM.2004.1276853. 81

[Aponte and Bohacek, 2012] Carlos R. Aponte and Stephan Bohacek. OLSR and approximate distance routing: Loops, black holes, and path stretch. In *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*, pages 1 –10, jan. 2012. doi: 10.1109/COMSNETS.2012.6151347. 16

[Arora et al., 2012] Deepali Arora, Eamon Millman, and Stephen W. Neville. Assessing the expected performance of the olsr routing protocol for denser urban core ad hoc network deployments. In *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*, pages 406 –414, march 2012. doi: 10.1109/AINA.2012.93. 41

[Aschenbruck and Schwamborn, 2010] Nils Aschenbruck and Matthias Schwamborn. Synthetic map-based mobility traces for the performance evaluation in opportunistic networks. In *Proceedings of the Second International Workshop on Mobile Opportunistic Networking*, MobiOpp '10, pages 143–146, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-925-1. doi: http://doi.acm.org/10.1145/1755743.1755769. 76

[Aschenbruck et al., 2010] Nils Aschenbruck, Raphael Ernst, Elmar Gerhards-Padilla, and Matthias Schwamborn. Bonnmotion: a mobility scenario generation and analysis tool. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, SIMUTools '10, pages 51:1–51:10, ICST, Brussels, Belgium, Belgium, 2010. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). ISBN 978-963-9799-87-5. doi: http://dx.doi.org/10.4108/ICST.SIMUTOOLS2010.8684. 76, 93

[Attwood et al., 2011] Andrew Attwood, Madjid Merabti, and Omar Abuelmaatti. Iomanets: Mobility architecture for wireless m2m networks. In *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, pages 399 –404, dec. 2011. doi: 10.1109/GLOCOMW.2011.6162479. 122

[Bai and Gong, 2010] Yuebin Bai and Lei Gong. Link availability prediction with radio irregularity coverage for mobile multi-hop networks. *Communications Letters, IEEE*, 14(6):518 –520, june 2010. ISSN 1089-7798. doi: 10.1109/LCOMM.2010.06.100238. 60

[Bai et al., 2009] Yuebin Bai, Jun Huang, Qingmian Han, and Depei Qian. Link Availability Based Mobility-Aware Max-Min Multi-Hop Clustering (M^4C) for Mobile Ad Hoc Networks. *IEICE Transactions on Communications*, 92:3132–3142, 2009. doi: 10.1587/transcom.E92.B.3132. 60

[Beigh and Peer, 2012] Bilal Maqbool Beigh and M. A. Peer. Performance evaluation of geographical routing protocols: An empirical study. In *Computer Communication and Informatics (ICCCI), 2012 International Conference on*, pages 1 –6, jan. 2012. doi: 10.1109/ICCCI.2012.6158920. 3

[Bernardo et al., 2011] Vitor Bernardo, Marilia Curado, Thomas Staub, and Torsten Braun. Towards energy consumption measurement in a cloud computing wireless testbed. In *Network Cloud Computing and Applications (NCCA), 2011 First International Symposium on*, pages 91 –98, nov. 2011. doi: 10.1109/NCCA.2011.22. 5

[Bezahaf et al., 2012] Mehdi Bezahaf, Luigi Iannone, Marcelo Dias de Amorim, and Serge Fdida. An experimental evaluation of cross-layer routing in a wireless mesh backbone. *Computer Networks*, 56(5):1568 – 1583, 2012. ISSN 1389-1286. doi: 10.1016/j.comnet.2012.01.004. 59

[Borges et al., 2011] Vinicius C.M. Borges, Marilia Curado, and Edmundo Monteiro. Cross-layer routing metrics for mesh networks: Current status and research directions. *Computer Communications*, 34(6):681 – 703, 2011. ISSN 0140-3664. doi: 10.1016/j.comcom.2010.12.001. 59

[Boukerche et al., 2011] Azzedine Boukerche, Begumhan Turgut, Nevin Aydin, Mohammad Z. Ahmad, Ladislau Bölöni, and Damla Turgut. Routing protocols in ad hoc networks: A survey. *Computer Networks*, 55(13):3032 – 3080, 2011. ISSN 1389-1286. doi: 10.1016/j.comnet.2011.05.010. 13

[Bulut and Szymanski, 2010] Eyuphan Bulut and Boleslaw K. Szymanski. Friendship based routing in delay tolerant mobile social networks. In *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*, pages 1 –5, dec. 2010. doi: 10.1109/GLOCOM.2010.5683082. 4

[Cabrera, 2009] Jorge Luis Ojeda Cabrera. *locpol: Kernel local polynomial regression*, 2009. URL `http://CRAN.R-project.org/package=locpol`. R package version 0.4-0. 67

[Cacciapuoti et al., 2012] Angela Sara Cacciapuoti, Francesco Calabrese, Marcello Caleffi, Giusy Di Lorenzo, and Luigi Paura. Human-mobility enabled networks in urban environments: Is there any (mobile wireless) small world out there? *Ad Hoc Networks*, 10(8):1520 – 1531, 2012. ISSN 1570-8705. doi: 10.1016/j.adhoc.2011.07.006. ¡ce:title¿Special Issue on Social-Based Routing in Mobile and Delay-Tolerant Networks¡/ce:title¿. 45

[Canourgues et al., 2008] Lucile Canourgues, Jerome Lephay, Laurent Soyer, and Andre-Luc Beylot. A scalable adaptation of the OLSR protocol for large clustered mobile ad hoc networks. *Advances in Ad Hoc Networking*, 2008. 55

[Cao and Sun, 2012] Y. Cao and Z. Sun. Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. *Communications Surveys Tutorials, IEEE*, PP(99):1 –24, 2012. ISSN 1553-877X. doi: 10.1109/SURV.2012.042512.00053. 83

[Chakeres and Perkins, 2012] Ian Chakeres and Charles Perkins. Dynamic manet on-demand (dymo) routing. *Internet Engineering Task Force, Internet-Draft*, March 2012. 22

[Chen and Wang, 2012] Shyr-Kuen Chen and Pi-Chung Wang. Shortcut anycast tree routing in manets. In *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*, pages 635 –640, march 2012. doi: 10.1109/AINA.2012.20. 13

[Ching-Chuan et al., 1997] Chiang Ching-Chuan, Wu Hsiao-Kuang, Liu Winston, and Gerla Mario. Routing in clustered multihop mobile wireless networks with fading channel. In *Proceedings of IEEE SICON*, pages 197–211, April 1997. 29

[Chorppath and Alpcan, 2011] Anil Kumar. Chorppath and Tansu Alpcan. Mechanism design for energy efficiency in wireless networks. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2011 International Symposium on*, pages 389 –394, may 2011. doi: 10.1109/WIOPT.2011.5930050. 122

[Chuang et al., 2012] Po-Jen Chuang, Po-Hsun Yen, and Ting-Yi Chu. Efficient route discovery and repair in mobile ad-hoc networks. In *Advanced Informa-*

tion Networking and Applications (AINA), 2012 IEEE 26th International Conference on, pages 391 –398, march 2012. doi: 10.1109/AINA.2012.120. 5

[Cimmino and Donadio, 2009] Antonio Cimmino and Pasquale Donadio. Overall requirements for global information multimedia communication village 10th strategic workshop. *Wireless Personal Communications*, 49(3): 311–319, May 2009. doi: 10.1007/s11277-009-9686-3. URL `http://dx.doi.org/10.1007/s11277-009-9686-3`. 1, 2, 6, 41

[Clausen and Jacquet, 2003] Thomas Clausen and Philippe Jacquet. Optimized link state routing protocol (olsr). RFC 3626, Internet Engineering Task Force, October 2003. 16, 47, 62

[Clausen et al., 2012] Thomas Clausen, Christopher Dearlove, and Philippe Jacquet. The optimized link state routing protocol version 2. *Internet Engineering Task Force, Internet-Draft*, May 2012. URL `http://tools.ietf.org/id/draft-ietf-manet-olsrv2-15.txt`. 18, 85

[Dargahi et al., 2008] Tooska Dargahi, Amir M. Rahmani, and Ahmad Khademzadeh. A hybrid of reactive and proactive approaches based on neighbors' conditions of nodes for mobile ad hoc networks. In *Information and Automation for Sustainability, 2008. ICIAFS 2008. 4th International Conference on*, pages 425 –430, dec. 2008. doi: 10.1109/ICIAFS.2008.4784004. 26

[De Clauzade De Mazieux et al., 2007] Alexandre Delye De Clauzade De Mazieux, Michel Marot, and Monique Becker. Correction, generalisation and validation of the "max-min d-cluster formation heuristic". In *Proceedings of the 6th international IFIP-TC6 conference on Ad Hoc and sensor networks, wireless networks, next generation internet*, NETWORKING'07, pages 1149–1152, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-72605-0. 43

[Eriksson et al., 2007] Jakob Eriksson, Michalis Faloutsos, and Srikanth V. Krishnamurthy. Dart: Dynamic address routing for scalable ad hoc and mesh networks. *Networking, IEEE/ACM Transactions on*, 15:119–132, 2007. ISSN 1063-6692. doi: 10.1109/TNET.2006.890092. 30, 86, 103

[Frangoudis et al., 2011] Pantelis A. Frangoudis, George C. Polyzos, and Vasileios P. Kemerlis. Wireless community networks: an alternative approach for nomadic broadband network access. *Communications Magazine, IEEE*, 49(5):206 –213, may 2011. ISSN 0163-6804. doi: 10.1109/MCOM.2011.5762819. 41

[Gaertner and Cahill, 2004] Gregor Gaertner and Vinny Cahill. Understanding link quality in 802.11 mobile ad hoc networks. *Internet Computing, IEEE*, 8(1):55 – 60, jan-feb 2004. ISSN 1089-7801. doi: 10.1109/MIC.2004.1260704. 71

[Gandhi et al., 2012] Savita Gandhi, Nirbhay Chaubey, Naren Tada, and Srushti Trivedi. Scenario-based performance comparison of reactive, proactive & hybrid protocols in manet. In *Computer Communication and Informatics (ICCCI), 2012 International Conference on*, pages 1 –5, jan. 2012. doi: 10.1109/ICCCI.2012.6158842. 20

[Garcia-Luna-Aceves and Roy, 2005] Jose J. Garcia-Luna-Aceves and Soumya Roy. On-demand loop-free routing with link vectors. *Selected Areas in Communications, IEEE Journal on*, 23(3):533 – 546, march 2005. ISSN 0733-8716. doi: 10.1109/JSAC.2004.842564. 28

[Garcia-Luna-Aceves and Spohn, 1999] Jose J. Garcia-Luna-Aceves and Marco Spohn. Source-tree routing in wireless networks. In *Network Protocols, 1999. (ICNP '99) Proceedings. Seventh International Conference on*, pages 273 – 282, 31 1999. 28

[Golubenco, 2012] Tudor Golubenco. Olsr daemon for quagga, 2012. URL `http://olsrdq.sourceforge.net`. http://olsrdq.sourceforge.net. 84

[Gonzalez, 1993] Teofilo F. Gonzalez. A simple lp-free approximation algorithm for the minimum weight vertex cover problem. Technical report, Santa Barbara, CA, USA, 1993. 36

[Grzonka et al., 2012] S. Grzonka, A. Karwath, F. Dijoux, and W. Burgard. Activity-based estimation of human trajectories. *Robotics, IEEE Transactions on*, 28(1):234 –245, feb. 2012. ISSN 1552-3098. doi: 10.1109/TRO.2011.2165372. 5

[Gupta and Gupta, 2010] N. Gupta and R. Gupta. Routing protocols in mobile adhoc networks: An overview. In *Emerging Trends in Robotics and Communication Technologies (INTERACT), 2010 International Conference on*, pages 173 –177, dec. 2010. doi: 10.1109/INTERACT.2010.5706220. 13

[Haas et al., 2002] Zygmunt J. Haas, Marc R. Pearlman, and Prince Samar. The zone routing protocol (zrp) for ad hoc networksmarc r. pearlman. *Internet Engineering Task Force, MANET Working Group Internet Draft (draft-ietf-manet-zone-zrp-04)*, 2002. 25

[Hamma et al., 2006] Takashi Hamma, Takashi Katoh, Bhed Bahadur Bista, and Toyoo Takata. An efficient zhls routing protocol for mobile ad hoc networks. In *Database and Expert Systems Applications, 2006. DEXA '06. 17th International Conference on*, pages 66–70, 2006. ISBN 1529-4188. doi: 10.1109/DEXA.2006.24. 35

[Härdle et al., 2004] W. Härdle, M. Müller, S. Sperlich, and A. Welatz. *Nonparametric and Semiparametric Models.* Springer Verlag, 2004. ISBN 3540207228. 66

[Herberg and Clausen, 2011] Ulrich Herberg and Thomas Clausen. Delay tolerant networking with olsrv2. In *Embedded and Ubiquitous Computing (EUC), 2011 IFIP 9th International Conference on*, pages 144 –149, oct. 2011. doi: 10.1109/EUC.2011.27. 18

[Herberg et al., 2012] Ulrich Herberg, Robert Cole, and Ian Chakeres. Mobile ad hoc network (manet) neighborhood discovery protocol (nhdp). *Internet Engineering Task Force, Internet-Draft*, August 2012. URL `http://tools.ietf.org/id/draft-ietf-manet-nhdp-17.txt`. 18

[Huang and Bai, 2008] Jun Huang and Yuebin Bai. A novel approach of link availability estimation for mobile ad hoc networks. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 26–30, 2008. ISBN 1550-2252. doi: 10.1109/VETECS.2008.18. 60, 62

[Hutson et al., 2007] Kevin R. Hutson, Terri L. Schlosser, and Douglas R. Shier. On the distributed bellman-ford algorithm and the looping problem. *INFORMS J. on Computing*, 19(4):542–551, October 2007. ISSN 1526-

5528. doi: 10.1287/ijoc.1060.0195. URL `http://dx.doi.org/10.1287/ijoc.1060.0195`. 15

[Isella et al., 2011] Lorenzo Isella, Juliette Stehlé, Alain Barrat, Ciro Cattuto, Jean-François Pinton, and Wouter Van den Broeck. What's in a crowd? analysis of face-to-face behavioral networks. *Journal of Theoretical Biology*, 271(1):166 – 180, 2011. ISSN 0022-5193. doi: 10.1016/j.jtbi.2010.11.033. 52

[ITU-T, 2003] ITU-T. Performance and quality of service requirements for international mobile telecommunications-2000 (imt-2000) access networks, August 2003. 86, 96, 104, 114

[Jacquet et al., 2002] Philippe Jacquet, Anis Laouiti, Pascale Minet, and Laurent Viennot. Performance of Multipoint Relaying in Ad Hoc Mobile Routing Protocols. In *Proceedings of the Second International IFIP-TC6 Networking Conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; and Mobile and Wireless Communications (NETWORKING '02)*, pages 387–398, Pisa, Italy, May 2002. 55, 56

[Jemili et al., 2011a] Ian Jemili, Nader Chaabouni, Abdelfattah Belghith, and Mohamed Mosbah. An optimized layered cluster-based routing protocol for ad hoc networks. In *Wireless Advanced (WiAd), 2011*, pages 156 –163, june 2011a. doi: 10.1109/WiAd.2011.5983304. 3

[Jemili et al., 2008] Imen Jemili, Abdelfattah Belghith, and Mohamed Mosbah. A synchronous tiered based clustering algorithm for large-scale ad hoc networks. In Zoubir Mammeri, editor, *Wireless and Mobile Networking*, volume 284 of *IFIP International Federation for Information Processing*, pages 41–55. Springer Boston, 2008. ISBN 978-0-387-84838-9. 10.1007/978-0-387-84839-6-4. 33

[Jemili et al., 2010] Imen Jemili, Abdelfattah Belghith, and Mohamed Mosbah. A layered cluster based routing for an ad hoc environment. In *Computer Systems and Applications (AICCSA), 2010 IEEE/ACS International Conference on*, pages 1 –8, may 2010. doi: 10.1109/AICCSA.2010.5587016. 33

[Jemili et al., 2011b] Imen Jemili, Naren Chaabouni, Abdelfattah Belghith, and Mohamed Mosbah. An optimized layered cluster-based routing protocol for ad hoc networks. In *Wireless Advanced (WiAd), 2011*, pages 156 –163, june 2011b. doi: 10.1109/WiAd.2011.5983304. 33

[Jian et al., 1999] Mingliand Jian, Jinyang Li, and Y.C. Tay. Cluster based routing protocol (cbrp). *Internet Engineering Task Force, MANET Working Group, Internet Draft: draft-ietf-manet-cbrp-spec-01*, August 1999. 36

[Jiang et al., 2005] Shengming Jiang, Dajiang He, and Jianqiang Rao. A prediction-based link availability estimation for routing metrics in MANETs. *Networking, IEEE/ACM Transactions on*, 13(6):1302–1312, 2005. ISSN 1063-6692. doi: 10.1109/TNET.2005.860094. 61, 62

[Joa-Ng and Lu, 1999] Mario Joa-Ng and I-Tai Lu. A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 17:1415–1425, 1999. ISSN 0733-8716. 34

[Johnson et al., 2007] David B. Johnson, David A. Maltz, and Yih C. Hu. The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4. *Internet Engineering Task Force, MANET Working Group, Request For Comments 4728*, 2007. 21, 62

[Kang et al., 2012] Mi-Seon Kang, Dong-Won Kum, Jae-Seung Bae, You-Ze Cho, and Anh-Ngoc Le. Mobility aware hybrid routing protocol for mobile ad hoc network. In *Information Networking (ICOIN), 2012 International Conference on*, pages 410 –414, feb. 2012. doi: 10.1109/ICOIN.2012.6164411. 26

[Kasera and Ramanathan, 1997] Kamal K. Kasera and Ram Ramanathan. A location management protocol for hierarchically organized multihop mobile wireless networks. In *Universal Personal Communications Record, 1997. Conference Record., 1997 IEEE 6th International Conference on*, pages 158 –162 vol.1, 12-16 1997. doi: 10.1109/ICUPC.1997.625502. 29

[Kim and Lee, 2011] Dong-Bum Kim and Seung-Ik Lee. A new hybrid routing algorithm: Ghr (group hierarchical routing). In *ICT Convergence (ICTC), 2011 International Conference on*, pages 573 –577, sept. 2011. doi: 10.1109/ICTC.2011.6082666. 36

[Kim and Yoo, 2007] Hun Kim and Myungsik Yoo. A scalable ad hoc routing protocol based on logical topology for ubiquitous community network. In *Advanced Communication Technology, The 9th International Conference on*, volume 2, pages 1306 –1310, 12-14 2007. doi: 10.1109/ICACT.2007.358597. 114

[Kleinrock and Stevens, 1971] Leonard Kleinrock and K. Stevens. Fisheye: A lenslike computer display transformation. , 1971. 18

[Kumar et al., 2012] S. Kumar, Nadeem Javaid, Z. Yousuf, H. Kumar, Zahoor Ali Khan, and Ayesha Bibi. Dsdv, dymo, olsr: Link duration and path stability. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, pages 1862 –1866, june 2012. doi: 10.1109/TrustCom.2012.133. 23

[Kushki et al., 2007] Azadeh Kushki, Konstantinos N. Plataniotis, and Anastasios N. Venetsanopoulos. Kernel-Based positioning in wireless local area networks. *Mobile Computing, IEEE Transactions on*, 6(6):689–705, 2007. ISSN 1536-1233. doi: 10.1109/TMC.2007.1017. URL `10.1109/TMC.2007.1017`. 63, 64

[Lee and Gerla, 2000] Sung-Ju Lee and Mario Gerla. Aodv-br: backup routing in ad hoc networks. In *Wireless Communications and Networking Conference, 2000. WCNC. 2000 IEEE*, volume 3, pages 1311–1316, 2000. 21

[Leong et al., 2006] Ben Leong, Barbara Liskov, and Robert Morris. Geographic routing without planarization. In *Proceedings of the 3rd conference on Networked Systems Design & Implementation - Volume 3*, NSDI'06, pages 25–25, Berkeley, CA, USA, 2006. USENIX Association. 42

[Lim and Datta, 2012] Kwan Hui Lim and A. Datta. An in-depth analysis of the effects of imep on tora protocol. In *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pages 3051 –3056, april 2012. doi: 10.1109/WCNC.2012.6214328. 22

[Lindgren et al., 2012] Anders Lindgren, Avri Doria, Elwyn Davies, and Samo Grasic. Probabilistic routing protocol for intermittently connected networks (prophet). Experimental RFC IRTF, Internet Engineering Task Force, May 2012. URL `http://tools.ietf.org/html/rfc6693.txt`. 4

[Lu et al., 2011] Jia Lu, Bin Zhang, Gang Han, JunHui Wang, and WenHua Dou. A new improvement on dsdv. In *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*, pages 1 –4, sept. 2011. doi: 10.1109/wicom.2011.6040278. 15

[MANETWG, 2012] MANETWG. Ietf mobile ad-hoc networks working group, 2012. URL `http://www.ietf.org/html.charters/manet-charter.html`. 21

[Manoj et al., 2001] Balakrishnan S. Manoj, R. Ananthapadmanabha, and C. Siva Ram Murthy. Link life based routing protocol for ad hoc wireless networks. In *Computer Communications and Networks, 2001. Proceedings. Tenth International Conference on*, pages 573 –576, 2001. doi: 10.1109/ICCCN.2001.956324. 61, 62

[Marron and Härdle, 1986] James Stephen Marron and Wolfgang Härdle. Random approximations to some measures of accuracy in nonparametric curve estimation. *Journal of Multivariate Analysis*, 20(1):91 – 113, 1986. ISSN 0047-259X. doi: DOI:10.1016/0047-259X(86)90021-7. 66

[OPNET Technologies, 2012] OPNET Technologies. Opnet modeler wireless suite simulator. Technical report, 2012. URL `http://www.opnet.com/`. 67, 81

[WiMAX Forum, 2011] WiMAX Forum. Ieee standard for local and metropolitan area networks part 16: Air interface for broadband wireless access systems amendment 3: Advanced air interface. *IEEE Std 802.16m-2011(Amendment to IEEE Std 802.16-2009)*, pages 1 –1112, 5 2011. doi: 10.1109/IEEESTD.2011.5765736. 1

[Mei et al., 2011] Allessandro Mei, Giacomo Morabito, Paolo Santi, and Julinda Stefa. Social-aware stateless forwarding in pocket switched networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 251 –255, april 2011. doi: 10.1109/INFCOM.2011.5935076. 4

[Murthy and Garcia-Luna-Aceves, 1996] Shree Murthy and Jose J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mob. Netw. Appl.*, 1(2):183–197, October 1996. ISSN 1383-469X. doi: 10.1007/BF01193336. 26

[Myers and Richardson, 1999] William Myers and Richardson. Comparison of propagation models. Technical Report 0, IEEE P802.16 Broadband Wireless Access Working Group, August 1999. URL `http://ieee802.org/16/tg2_orig/contrib/80216cc-99_13.pdf`. 82

[Navidi and Camp, 2004] William Navidi and Tracy Camp. Stationary distributions for the random waypoint mobility model. *IEEE Trans. Mob. Comput.*, 3(1):99–108, 2004. URL `http://dblp.uni-trier.de/db/journals/tmc/tmc3.html`. 113

[Nikaein et al., 2000] Navid Nikaein, Houda Labiod, and Christian Bonnet. Ddr-distributed dynamic routing algorithm for mobile ad hoc networks. In *Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC. 2000 First Annual Workshop on*, pages 19 –27, 2000. doi: 10.1109/MOBHOC.2000.869209. 35, 37

[Nikaein et al., 2001] Navid Nikaein, Christian Bonnet, and Neda Nikaein. Harp - hybrid ad-hoc routing protocol. In *Proceedings of IST: International Symposium on Telecommunications*, September 2001. 35

[Nordemann and Tonjes, 2012] Frank Nordemann and Ralf Tonjes. Transparent and autonomous store-carry-forward communication in delay tolerant networks (dtns). In *Computing, Networking and Communications (ICNC), 2012 International Conference on*, pages 761 –765, 30 2012-feb. 2 2012. doi: 10.1109/ICCNC.2012.6167525. 6, 13, 49, 87, 88

[Oda et al., 2007] Ryotaro Oda, Tomoyuki Ohta, and Yoshiaki Kakuda. An efficient on-demand hierarchical routing protocol based on autonomous clustering for mobile ad hoc networks. In *Autonomous Decentralized Systems, 2007. ISADS '07. Eighth International Symposium on*, pages 180 –187, march 2007. doi: 10.1109/ISADS.2007.16. 32

[Ohta et al., 2004] Tomoyuki Ohta, Munehiko Fujimoto, Ryotaro Oda, and Yoshiaki Kakuda. A class of hierarchical routing protocols based on autonomous clustering for large mobile ad hoc networks. In *Proc. IEICE Trans. Commun*, volume E8 7-B, pages 2500–2510, April 2004. 32

[Ortiz, 2009] Sixto Ortiz. IEEE 802.11n: The road ahead. *IEEE Computer*, 42(7): 13–15, 2009. ISSN 0018-9162. doi: 10.1109/MC.2009.224. 67, 81, 113

[Paillassa et al., 2011] Beatrice Paillassa, Cholatip Yawut, and Riadh Dhaou. Network awareness and dynamic routing: The ad hoc network case. *Computer Networks*, 55(9):2315 – 2328, 2011. ISSN 1389-1286. doi: 10.1016/j.comnet.2011.03.010. 13

[Palma and Curado, 2009] David Palma and Marilia Curado. Inside-out olsr scalability analysis. In *Ad-Hoc, Mobile and Wireless Networks*, volume 5793 of *Lecture Notes in Computer Science*, pages 354–359. Springer Berlin / Heidelberg, 2009. 85

[Pang and Qin, 2006] Kong Ling Pang and Yang Qin. The comparison study of flat routing and hierarchical routing in ad hoc wireless networks. In *Networks, 2006. ICON '06. 14th IEEE International Conference on*, volume 1, pages 1 –6, sept. 2006. doi: 10.1109/ICON.2006.302584. 54

[Panisson et al., 2011] André Panisson, Alain Barrat, Ciro Cattuto, Wouter Van den Broeck, Giancarlo Ruffo, and Rossano Schifanella. On the dynamics of human proximity for data diffusion in ad-hoc networks. *Ad Hoc Networks*, (0):–, 2011. ISSN 1570-8705. doi: 10.1016/j.adhoc.2011.06.003. 52

[Park and Corson, 2001] Vincent Park and Scott Corson. Temporally-ordered routing algorithm (tora) version 1 functional specification. *Internet Engineering Task Force, MANET Working Group - draft-ietf-manet-tora-spec-04*, July 2001. 22

[Parvathi, 2012] P. Parvathi. Comparative analysis of cbrp, aodv, dsdv routing protocols in mobile ad-hoc networks. In *Computing, Communication and Applications (ICCCA), 2012 International Conference on*, pages 1 –4, feb. 2012. doi: 10.1109/ICCCA.2012.6179145. 36

[Patel and Srivastava, 2010] Brijesh Patel and Sanjay Srivastava. Performance analysis of zone routing protocols in mobile ad hoc networks. In *Communications (NCC), 2010 National Conference on*, pages 1 –5, jan. 2010. doi: 10.1109/NCC.2010.5430164. 25

[Pei et al., 2000] Guangyu Pei, Mario Gerla, and Tsu-Wei Chen. Fisheye state routing: a routing scheme for ad hoc wireless networks. In *Communications, 2000. ICC 2000. 2000 IEEE International Conference on*, volume 1, pages 70 –74 vol.1, 2000. doi: 10.1109/ICC.2000.853066. 18

[Perkins and Bhagwat, 1994] Charles Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, pages 234–244, 1994. 15

[Perkins et al., 2003a] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir R. Das. Ad hoc on-demand distance vector (aodv) routing. RFC Experimental 3561, Internet Engineering Task Force, July 2003a. URL `http://www.ietf.org/rfc/rfc3561.txt`. 62

[Perkins et al., 2003b] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir R. Das. Ad hoc on-demand distance vector (aodv) routing. *Internet Engineering Task Force, Request For Comments 3561*, July 2003b. URL `http://www.ietf.org/rfc/rfc3561.txt`. 20

[Pham et al., 2005] Dung Pham, Y. Ahmet Sekercioglu, and Gregory K. Egan. Performance of ieee 802.11b wireless links: An experimental study. In *TENCON 2005 2005 IEEE Region 10*, pages 1 –6, 2005. doi: 10.1109/TENCON.2005.300898. 68

[Piran et al., 2011] Mohammad Jalil Piran, Garimella Rama Murthy, G. Praveen Babu, and Ehsan Ahvar. Total gps-free localization protocol for vehicular ad hoc and sensor networks (vasnet). In *Computational Intelligence, Modelling and Simulation (CIMSiM), 2011 Third International Conference on*, pages 388 –393, sept. 2011. doi: 10.1109/CIMSim.2011.77. 3, 5

[Prasad et al., 2009] Pratap S. Prasad, Prathima Agrawal, and Krishna M. Sivalingam. Effects of mobility in hierarchical mobile ad hoc networks. In *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*, pages 1 –5, jan. 2009. doi: 10.1109/CCNC.2009.4784895. 3

[Preveze and Safak, 2010] Barbaros Preveze and Aysel Safak. Comparative analysis of novel long life routing methods in mobile networks. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2010 IEEE 21st International Symposium on*, pages 1596 –1601, sept. 2010. doi: 10.1109/PIMRC.2010.5671945. 23

[Qin and Zimmermann, 2010] Min Qin and R. Zimmermann. An adaptive strategy for mobile ad hoc media streaming. *Multimedia, IEEE Transactions on*, 12(4):317 –329, 2010. ISSN 1520-9210. doi: 10.1109/TMM.2010.2046275. 113

[Qin et al., 2005] Min Qin, Roger Zimmermann, and Leslie S. Liu. Supporting multimedia streaming between mobile peers with link availability prediction. In *Proceedings of the 13th annual ACM international conference on Multimedia*, MULTIMEDIA '05, pages 956–965, New York, NY, USA, 2005. ACM. ISBN 1-59593-044-2. doi: http://doi.acm.org/10.1145/1101149.1101350. 60

[R Development Core Team, 2010] R Development Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2010. URL `http://www.R-project.org/`. ISBN 3-900051-07-0. 67

[Rajkumar et al., 2012] G. Rajkumar, R. Kasiram, and D. Parthiban. Optimizing throughput with reduction in power consumption and performance comparison of dsr and aodv routing protocols. In *Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on*, pages 943 –947, march 2012. doi: 10.1109/ICCEET.2012.6203855. 21

[Ravilla et al., 2011] D. Ravilla, V. Sumalatha, and P.C.S. Reddy. Performance comparisons of zrp and izrp routing protocols for ad hoc wireless networks. In *Energy, Automation, and Signal (ICEAS), 2011 International Conference on*, pages 1 –8, dec. 2011. doi: 10.1109/ICEAS.2011.6147133. 25

[Ros and Ruiz, 2007] Francisco J. Ros and Pedro M. Ruiz. Cluster-based olsr extensions to reduce control overhead in mobile ad hoc networks. In *Proceedings of the 2007 international conference on Wireless communications and mobile computing*, IWCMC '07, pages 202–207, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-695-0. doi: http://doi.acm.org/10.1145/1280940.1280985. 30

[Saeed et al., 2008] N.H. Saeed, Maysam F. Abbod, and Hamed S. Al-Raweshidy. Modeling MANET utilizing artificial intelligent. In *Computer Modeling* *and Simulation, 2008. EMS '08. Second UKSIM European Symposium on*, pages 117–122, 2008. doi: 10.1109/EMS.2008.8. 62

[Sagar et al., 2012] S. Sagar, N. Javaid, Z.A. Khan, J. Saqib, A. Bibi, and S.H. Bouk. Analysis and modeling experiment performance parameters of routing protocols in manets and vanets. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, pages 1867 –1871, june 2012. doi: 10.1109/TrustCom.2012.89. 82

[Samar et al., 2004] Prince Samar, Marc R. Pearlman, and Zygmunt J. Haas. Independent zone routing: an adaptive hybrid routing framework for ad hoc wireless networks. *IEEE/ACM Trans. Netw.*, 12(4):595–608, August 2004. ISSN 1063-6692. doi: 10.1109/TNET.2004.833153. 25

[Santo and Fortunato, 2010] Santo and Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010. ISSN 0370-1573. doi: 10.1016/j.physrep.2009.11.002. 4

[Sehgal and Gondal, 2007] Shoaib Sehgal and Iqbal Gondal. Mobile multimedia communications: Concepts, applications, and challenges, November 2007. 29

[Shakeri and Hosseini, 2010] Majid Shakeri and Ebrahim Hosseini. Simulation and evaluation of routing protocols in wireless mobile ad hoc network. In *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*, pages 603 –607, 2010. doi: 10.1109/WCINS.2010.5545215. 76, 113

[Shen et al., 2012] Zukang Shen, Aris Papasakellariou, Juan Montojo, Dirk Gerstenberger, and Fangli Xu. Overview of 3gpp lte-advanced carrier aggregation for 4g wireless communications. *Communications Magazine, IEEE*, 50(2):122 –130, february 2012. ISSN 0163-6804. doi: 10.1109/MCOM.2012.6146491. 1

[Shu and Li, 2007] Xiao Shu and Xining Li. Link failure rate and speed of nodes in wireless network. In *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, pages 1441–1444, 2007. doi: 10.1109/WICOM.2007.365. 61, 62

[Siekkinen et al., 2012] Matti Siekkinen, Jukka K. Nurminen, and Antti Yla-Jaaski. Creating green incentives and mechanisms through packet-level energy accounting. In *Wireless Communications and Networking Conference Workshops (WCNCW), 2012 IEEE*, pages 12 –17, april 2012. doi: 10.1109/WCNCW.2012.6215475. 122

[Sivakumar and Chelliah, 2012] Nithya Rekha Sivakumar and Chandrasekar Chelliah. Simulation and evaluation of the performance on probabilistic broadcasting in fsr (fisheye state routing) routing protocol based on random mobility model in manet. In *Computational Intelligence, Communication Systems and Networks (CICSyN), 2012 Fourth International Conference on*, pages 371 –376, july 2012. doi: 10.1109/CICSyN.2012.74. 19

[Song and Lutfiyya, 2009] Wang-Cheol Song and Hanan Lutfiyya. Delivery-guaranteed geocast in manets by using zhls. In *Computer Sciences and Convergence Information Technology, 2009. ICCIT '09. Fourth International Conference on*, pages 86 –90, nov. 2009. doi: 10.1109/ICCIT.2009.194. 35

[Spaho et al., 2012] Evjola Spaho, Makoto Ikeda, Leonard Barolli, Fatos Xhafa, A. Biberaj, and J. Iwashige. Performance comparison of dsdv and dymo protocols for vehicular ad hoc networks. In *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*, pages 629 –634, march 2012. doi: 10.1109/AINA.2012.56. 15

[Suresh and Duraiswamy, 2010] A. Suresh and K. Duraiswamy. Scalable instant way point routing protocol for manet. In *Computing Communication and Networking Technologies (ICCCNT), 2010 International Conference on*, pages 1 –8, july 2010. doi: 10.1109/ICCCNT.2010.5591658. 30

[Tnnesen et al., 2012] A. Tnnesen, T. Lopatic, H. Gredler, B. Petrovitsch, A. Kaplan, and S.-O. Tcke. Olsrd - an ad-hoc wireless mesh routing daemon, July 2012. URL `http://www.olsr.org`. 70, 84

[Toh, 1996] Chai-Keong Toh. A novel distributed routing protocol to support ad-hoc mobile computing. In *Computers and Communications, 1996., Conference Proceedings of the 1996 IEEE Fifteenth Annual International*

Phoenix Conference on, pages 480–486, 1996. doi: 10.1109/PCCC.1996.493675. 23

[Trindade and Vazao, 2011a] Joao Trindade and Teresa Vazao. Hran: Heat routing protocol for ad-hoc networks. In *Ad Hoc Networking Workshop (Med-Hoc-Net), 2011 The 10th IFIP Annual Mediterranean*, pages 131 –134, june 2011a. doi: 10.1109/Med-Hoc-Net.2011.5970479. 26

[Trindade and Vazao, 2011b] Joao Trindade and Teresa Vazao. Hran: a scalable routing protocol for multihop wireless networks using bloom filters. In *Proceedings of the 9th IFIP TC 6 international conference on Wired/wireless internet communications*, WWIC'11, pages 434–445, Berlin, Heidelberg, 2011b. Springer-Verlag. ISBN 978-3-642-21559-9. 26, 89

[Wand and Jones, 1994] Matt P. Wand and M. Chris Jones. *Kernel Smoothing.* Chapman and Hall/CRC, 1 edition, December 1994. ISBN 0412552701. 64, 65

[Wang et al., 2010] Haiquan Wang, Le Mi, Chunhe Xia, Liangshuang Lv, and Meng Chen. Tldv: Tree-like locator distance vector routing protocol for manets. In *Educational and Information Technology (ICEIT), 2010 International Conference on*, volume 2, pages V2–291 –V2–296, sept. 2010. doi: 10.1109/ICEIT.2010.5607575. 31

[Wang and Olariu, 2004] Lan Wang and Stephan Olariu. A two-zone hybrid routing protocol for mobile ad hoc networks. *IEEE Trans. Parallel Distrib. Syst.*, 15(12):1105–1116, December 2004. ISSN 1045-9219. doi: 10.1109/TPDS.2004.73. 25

[Wu et al., 2009] Xianren Wu, Hamid Sadjadpour, and Jose Garcia luna aceves. Link dynamics in MANETS restricted node mobility: modeling and applications. *Wireless Communications, IEEE Transactions on*, 8(9):4508–4517, 2009. ISSN 1536-1276. doi: 10.1109/TWC.2009.070309. 61, 62

[Xing et al., 2009] Bo Xing, Karim Seada, and Nalini Venkatasubramanian. An experimental study on wi-fi ad-hoc mode for mobile device-to-device video delivery. In *INFOCOM Workshops 2009, IEEE*, pages 1 –6, April 2009. doi: 10.1109/INFCOMW.2009.5072111. 81

[Yang and Tseng, 2007] Chun-Chuan Yang and Li-Pin Tseng. Fisheye zone routing protocol: A multi-level zone routing protocol for mobile ad hoc networks. *Comput. Commun.*, 30(2):261–268, January 2007. ISSN 0140-3664. doi: 10.1016/j.comcom.2006.08.014. 26

[Yang et al., 2007] Hyo-Sik Yang, Luke Ritchie, Andréa W. Richa, and Martin Reisslein. Manet routing with provably low complexity through constant density clustering and route request broadcast. *Wirel. Pers. Commun.*, 43(2):605–621, October 2007. ISSN 0929-6212. doi: 10.1007/s11277-007-9252-9. 85

[Yelemou et al., 2012] Tiguiane Yelemou, Philippe Meseure, and Anne-Marie Poussard. Improving zrp performance by taking into account quality of links. In *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pages 2956 –2960, april 2012. doi: 10.1109/WCNC.2012.6214310. 25

[Yu et al., 2007] Ming Yu, Aniket Malvankar, Wei Su, and Simon Y. Foo. A link availability-based QoS-aware routing protocol for mobile ad hoc sensor networks. *Computer Communications*, 30(18):3823–3831, December 2007. ISSN 0140-3664. doi: 10.1016/j.comcom.2007.09.009. 60, 62

[Zhang and Chong, 2009] Mingyang Zhang and Peter H. J. Chong. Performance comparison of flat and cluster-based hierarchical ad hoc routing with entity and group mobility. In *WCNC'09: Proceedings of the 2009 IEEE conference on Wireless Communications & Networking Conference*, pages 2450–2455, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-2947-9. 37

[Zhang et al., 2010] Xin Ming Zhang, Feng Fu Zou, En Bo Wang, and Dan Keun Sung. Exploring the dynamic nature of mobile nodes for predicting route lifetime in mobile ad hoc networks. *Vehicular Technology, IEEE Transactions on*, 59(3):1567 –1572, 2010. ISSN 0018-9545. doi: 10.1109/TVT.2009.2038708. 61, 62

[Zhou et al., 2009] Biao Zhou, Zhen Cao, and Mario Gerla. Cluster-based inter-domain routing (cidr) protocol for manets. In *Wireless On-Demand Network Systems and Services, 2009. WONS 2009. Sixth International Con-*

ference on, pages 19 –26, feb. 2009. doi: 10.1109/WONS.2009.4801843. 43

[Zhou and Ying, 2010] Shan Zhou and Lei Ying. On delay constrained multicast capacity of large-scale mobile ad-hoc networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1 –5, 2010. doi: 10.1109/INFCOM.2010.5462257. 6, 113

[Zhu et al., 2010] Guohun Zhu, Xi Jiang, Chun Wu, and Zhiqiang He. A cluster head selection algorithms in wireless network based on maximal weighted independent set. In *Ubiquitous Information Technologies and Applications (CUTE), 2010 Proceedings of the 5th International Conference on*, pages 1 –6, 2010. doi: 10.1109/ICUT.2010.5678667. 36

Appendix A

Deferred Routing Specification

Having presented the Deferred Routing scheme and its mechanisms in Chapter 3, further details are required in order to better understand how routing information is handled and how data packets are correctly forwarded throughout the network. For this purpose, several algorithms in pseudocode are presented in this Chapter, in conjunction with a description of their main objective. Due to usage of virtual clusters by Deferred Routing (DefeR), which aggregate different contexts into new ones, the terms cluster and context will be used interchangeably. The provided algorithms are suited to be used with any link-state routing protocol, which will be held responsible for intra-cluster routing.

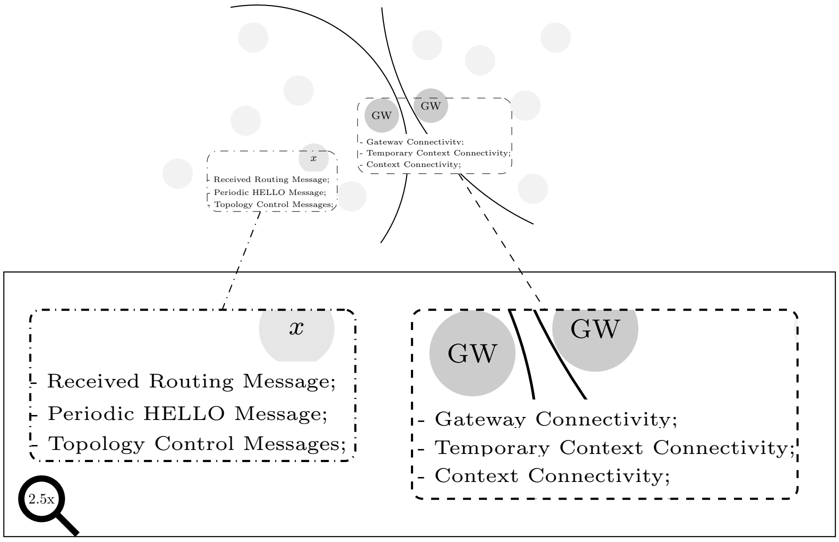


Figure A.1: DefeR Procedures in the Network

A.1 Required Algorithms

In DefeR the most important procedures handle the routing messages exchanged between Gateway (GW) nodes, which create new context connectivity information, and between normal nodes which propagate it, as shown in figure A.1. However, in all these procedures, the contextualization of the received Cluster Identifiers (CIDs) is of extreme importance. In each received routing message a CID needs to be processed. Nonetheless, despite the value of this CID, it must be computed according to each node's own perspective. The procedure taken with each received CID is detailed in Algorithm 6, which returns the correct view of the received CID, regarding the node where it was received.

Algorithm 6 View Determination Procedure

```
procedure DETERMINE_VIEW($CID_{foreign}$)

    $level_{own} \leftarrow$ GET_LEVEL($own\_context_{id}$)
    $level_{foreign} \leftarrow$ GET_LEVEL($CID_{foreign}$)
    if $level_{foreign} > level_{own}$ then //Needs to be Raised
        $CID_{foreign} \leftarrow$ JOIN_VIEW($CID_{foreign}, level_{foreign} - level_{own}$)
    else if $level_{foreign} < level_{own}$ then
        $own\_context_{id} \leftarrow$ JOIN_VIEW($own\_context_{id}, level_{own} - level_{foreign}$)
    end if

    if $own\_context_{id}$ mod $2 = 0$ then //To check if the CIDs are "brothers"
        $even \leftarrow -1$
    else
        $even \leftarrow 1$
    end if
    while $own\_context_{id} + even \neq CID_{foreign}$ and $own\_context_{id} \neq CID_{foreign}$ do
        //Perform a join until both CIDs are at the same level
        $CID_{foreign} \leftarrow$ JOIN_VIEW($CID_{foreign}, 1$)
        $own\_context_{id} \leftarrow$ JOIN_VIEW($CID_{foreign}, 1$)
        if $own\_context_{id}$ mod $2 = 0$ then
            $even \leftarrow -1$
        else
            $even \leftarrow 1$
        end if

    end while
    return $own\_context_{id}$

end procedure

procedure JOIN_VIEW($CID, n_{level}$)
    $CID_{new} \leftarrow \lceil [CID - (2^{n_{level}+1} - 2)]/2^{n_{level}} \rceil$
    return $CID_{new}$

end procedure

procedure GET_LEVEL($CID$)
    $Level \leftarrow \lfloor log_2(CID + 1) \rfloor$
    return $Level$
end procedure

```

An important part of the view determination of each CID is the assignment of CIDs in the first place. The binary hierarchy is used not only for performance purposes, but also to accommodate the creation and deletion of clusters. Whenever a new cluster is added, or when an existing one grows enough and divides itself, two new CIDs are created. The existing Cluster Identifier of the dividing cluster, or of the cluster to where the new one is attached, is kept unchanged becoming a virtual cluster. The remaining two CIDs are calculated using the old CID: $New\_CID_{left} = Old\_CID \times 2 + 1; New\_CID_{right} = Old\_CID \times 2 + 2$. This simple but efficient numbering of clusters allows the determination of the hierarchical level of each CID directly and it also allows to determine which clusters are contained within each virtual cluster without difficulty, as detailed in Algorithm 7.

Algorithm 7 Cluster Containment Procedure

```
procedure CONTAINS_CONTEXT($container, contained$)
    if $contained < container$ then
        return $FALSE$
    end if
    $container_{level} \leftarrow$ GET_LEVEL($container$)
    $contained_{level} \leftarrow$ GET_LEVEL($contained$)
    $gap \leftarrow contained_{level} - container_{level}$
    $contained \leftarrow$ JOIN_VIEW($contained, gap$)
    if $contained = container$ then
        return $TRUE$
    else
        return $FALSE$
    end if
end procedure
```

Having defined how cluster information must be handled, the remaining processes still need to gather and share this information so that it can be used in the routing process. Being a purely proactive protocol, the DefeR scheme relies on periodic messages such as *HELLO* messages to propagate routing information. These messages have their typical role of link sensing, such as in Optimized Link State Routing (OLSR), but in addition to this they also include information about adjacent clusters and the existing GWs capable of reaching them. The number of adjacent clusters is limited to $\lfloor log_2(CID + 1) \rfloor$ due to the used aggregation scheme, thus having little overhead on these messages.

Another task to be carried by these messages is the dissemination of the mapping Identifier (ID)-CID of each node in the network. Typically the used ID will be the Internet Protocol (IP) address of the nodes, but any other identification can be used. It is important to highlight that this duty could be assigned to the clustering protocol as it already manages this information, however, in order to be completely independent from any clustering algorithm, the DefeR protocol will include this information. As a scalability measure, each mapping will only be included in these messages, for a limited amount of time when for instance, a change occurs due to mobility. Algorithm 8 details how the necessary information is processed and included in *HELLO* messages.

Algorithm 8 Periodic *HELLO* messages

```
procedure Send_HELLO_Message
    ...
    //Share IP Mappings
    $message.ip\_mappings_{shared} \leftarrow$ List_Create()
    for each $mapping_{entry}$ in $temp\_ip\_context\_mapping_{table}$ do
        $new_{tuple} \leftarrow$ Pmo_Alloc($Ip\_Context\_Mapping_{pmh}$)
        $new_{tuple}.src_{addr} \leftarrow mapping_{entry}.ip_{addr}$
        $new_{tuple}.context_{id} \leftarrow mapping_{entry}.context_{id}$
        List_Insert($message.ip\_mappings_{shared}, new_{tuple}, TAIL$)
    end for
    //Share Context Connectivity
    $message.context_{connectivity} \leftarrow$ List_Create()
    for each $context_{entry}$ in $context\_connectivity_{table}$ do
        $conn_{tuple} \leftarrow$ Pmo_Alloc($Connectivity\_Tuple_{pmh}$)
        $conn_{tuple}.gateways \leftarrow$ List_Create()
        for each $gw_{entry}$ in $context_{entry}.gw_{table}$ do
            if $gw_{entry}.exp_{time} < 0$ then
                break
            end if
            if $gw_{entry}.seq_{num} = -1$ then
                $gw_{entry}.seq_{num} = message.seq_{num}$
            end if
            $new_{tuple} \leftarrow$ Pmo_Alloc($Context\_Connectivity\_Tuple_{pmh}$)
            $new_{tuple}.hop_{count} \leftarrow gw_{entry}.hop_{count}$
            $new_{tuple}.gw_{addr} \leftarrow gw_{entry}.gateway$
            $new_{tuple}.age \leftarrow gw_{entry}.metric$
            $new_{tuple}.originator \leftarrow gw_{entry}.originator$
            $new_{tuple}.seq_{num} \leftarrow gw_{entry}.seq_{num}$
            $new_{tuple}.exp_{time} \leftarrow gw_{entry}.exp_{time} - current_{time}$
            List_Insert($conn_{tuple}.gateways, new_{tuple}, TAIL$)
        end for
        List_Insert($message.context_{connectivity}, conn_{tuple}, TAIL$)
    end for
    ...
end procedure
```

In proactive routing protocols such as OLSR, *HELLO* messages are simply exchanged between neighbour nodes and only Topology Control (*TC*) messages are forwarded through the network, using Multipoint Relay (MPR) nodes, propagating network topology information. In the DefeR scheme the same approach in intra-cluster routing is used aiming at reducing communication overhead. Additionally, only context connectivity information is added to these messages (Algorithm 9), including the ID-CID mapping information in *HELLO* messages only.

As a result of nodes being separated into different Routing Contexts, each message must be processed according to its own Context. Therefore, a routing message is only

Algorithm 9 Topology Control Messages

```
procedure SEND_TC_MESSAGE
    ...
    //Share Context Connectivity
    $message.context_{connectivity} \leftarrow$ LIST_CREATE()
    for each $context_{entry}$ in $context\_connectivity_{table}$ do
        $conn_{tuple} \leftarrow$ PMO_ALLOC($Connectivity\_Tuple_{pmh}$)
        $conn_{tuple}.gateways \leftarrow$ LIST_CREATE()
        for each $gw_{entry}$ in $context_{entry}.gw_{table}$ do
            if $gw_{entry}.exp_{time} < 0$ then
                continue
            end if
            if $gw_{entry}.seq_{num} = -1$ then
                $gw_{entry}.seq_{num} = message.seq_{num}$
            end if
            $new_{tuple} \leftarrow$ PMO_ALLOC($Context\_Connectivity\_Tuple_{pmh}$)
            $new_{tuple}.hop_{count} \leftarrow gw_{entry}.hop_{count}$
            $new_{tuple}.gw_{addr} \leftarrow gw_{entry}.gateway$
            $new_{tuple}.age \leftarrow gw_{entry}.metric$
            $new_{tuple}.originator \leftarrow gw_{entry}.originator$
            $new_{tuple}.seq_{num} \leftarrow gw_{entry}.seq_{num}$
            $new_{tuple}.exp_{time} \leftarrow gw_{entry}.exp_{time} - current_{time}$
            LIST_INSERT($conn_{tuple}.gateways, new_{tuple}, TAIL$)
        end for
        LIST_INSERT($message.context_{connectivity}, conn_{tuple}, TAIL$)
    end for
    ...
end procedure
```

considered for intra-cluster routing if it belongs to the same Context of the node that received it. In order to implement this behaviour, the DefeR procedures for routing messages processing were modified to firstly, obtain all the known ID Mappings and secondly, the existing Context Information.

The context connectivity information is created by border nodes which, upon receiving foreign routing messages (from a different cluster), extract the previously mentioned content and discard the packet, so that intra-cluster routing procedures may be unaware of such message. From the obtained content, the gathered ID-CID tuples are modified in order to reflect the node's own network perspective and finally, the entire GW information is absorbed by the new GW node for future connections with different clusters.

Whenever routing messages are received within the same context, nodes retrieve and process the complete DefeR routing information, letting the packet then be processed by the used link-state protocol, which uses the remaining information for maintaining the routing tables within the context, as specified in Algorithm 10.

Having obtained the necessary knowledge about the existing contexts and how they can be reached through the available GWs, the remaining process required is to properly route data packets. Whenever the destination node is within the same context as

Algorithm 10 Processing Received Routing Messages

```
procedure Process_Routing_Message($message$) //The same code for $HELLO$ and TC messages
    ...
    $src_{addr} \leftarrow message.src_{ip}$
    $context_{id} \leftarrow message.context_{id}$
    Ip_Context_Mapping_Create($ip\_context\_mapping_{table}, src_{addr}, context_{id}$, true)

    for each $mapping_{entry}$ in $message.ip\_mappings_{shared}$ do
        //If this information comes from within the same context, it will be always processed
        //otherwise only if it does not contain an aggregated view of this context
        if $context_{id} \neq own\_context_{id}$ and
                Contains_Context($mapping_{entry}.context_{id}, own\_context_{id}$) then
            continue
        end if

        Ip_Context_Mapping_Create($ip\_context\_mapping_{table}, mapping_{entry}.ip_{addr}$,
                                                  $mapping_{entry}.context_{id}$, false)
    end for

    if $context_{id} \neq own\_context_{id}$ then //This Node is a Gateway
        Gw_Connectivity_Create($gw\_connectivity_{table}, src_{addr}, context_{id}, 1$,
                                                  $own_{addr}, -1, GW\_TIMEC$)
        for each $context_{entry}$ in $message.context_{connectivity}$ do
            //Ignore information regarding the same context, as well as
            //information about the received context's message
            if Determine_View($context_{id}$) = Determine_View($context_{entry}.context_{id}$)
                        and Contains_Context($context_{entry}.context_{id}, own\_context_{id}$) then
                continue
            end if
            for each $gw_{entry}$ in $context_{entry}.gateways$ do
                Gw_Connectivity_Create($gw\_connectivity_{table}$,
                    $src_{addr}, context_{entry}.context_{id}, gw_{entry}.hop_{count} + 1$,
                        $gw_{entry}.originator, gw_{entry}.seq_{num}, gw_{entry}.exp_{time}$)
            end for
        end for

        //Since this message comes from a different context, it must not be processed
        //by the Link-state protocol which only handles Intra-Context Messages
        return //Consequently, end the procedure

    else//Message received from the same context
        for each $context_{entry}$ in $message.context_{connectivity}$ do
            //Ignore information regarding the own context
            //(should only occur with out-of-date moving nodes)
            if Contains_Context($context_{entry}.context_{id}, own\_context_{id}$) then
                continue
            end if
            for each $gw_{entry}$ in $context_{entry}.gateways$ do
                //A node does not need information about its own connectivity
                if $gw_{entry}.gw_{addr} = own\_addr$ then
                    continue
                end if
                Context_Connectivity_Create($context\_connectivity_{table}$,
                    $gw_{entry}.gw_{addr}, context_{entry}.context_{id}, gw_{entry}.hop_{count}$,
                        $gw_{entry}.originator, gw_{entry}.seq_{num}, gw_{entry}.metric, gw_{entry}.exp_{time}$)
            end for
        end for
    end if
    ...
end procedure

```

the source node, the used link-state routing protocol should be able to immediately retrieve an appropriate route for incoming packets. However, if source and destination are in different contexts, all the gathered Context information will be used by the Inter-Cluster procedures, responsible for ensuring that the packets are able to reach their destination. Since the routing paradigm is different from typical Link-State or Distance-Vector protocols, an end-to-end path is not established and the packets are rather forwarded to the most suitable GW capable of reaching the desired context with the smallest number of cluster-hops. The necessary steps to accomplish this task are detailed in Algorithms 11 and 12.

Algorithm 11 Inter-Cluster Routing

```
procedure PACKET_ARRIVAL_HANDLE($Packet_{Ptr}, resend$)
    ...
    Get_Packet_Information

    //Look up the destination's Context ID
    if CONTEXT_ID_LOOKUP($Packet_{ptr}.dest_{addr}$)= $-1$ then //Context not found!
        return
    else if ($next_{hop} \leftarrow$ NEXT_HOP_FINDER($Packet_{ptr}.dest_{addr}$)) = $-1$ then
        //Next hop not found!
        if not resend then
            ADD_PACKET_RESEND($Packet_{Ptr}$)
        end if
        return
    end if

    SEND_PACKET($Packet_{ptr}, next_{hop}$)
end procedure
```

The previously presented procedures depict the overall mechanisms of the DefeR protocol so that routing messages are sent and received appropriately, and are then used for the routing of incoming data packets. Despite this, in order to handle and process all the gathered information, other extremely important procedures are required.

The details of mapping a node to its Context are crucial for the forwarding of a packet to its destination. Even though this mapping could be done with any unique node identifier, the IP address of each node is used in conjunction with its own Context, so that this DefeR specification is compatible with most of the existing routing protocols.

As previously seen, IP-CID cluster information is propagated in each *HELLO* message. However, since nodes are not likely to constantly change Context, this information is only propagated for a certain amount of time and only when a change is detected, for instance when a node changes its context. This propagation contains only the registered changes stored in a temporary table, not the entire mapping set, and it is only

Algorithm 12 Inter-Cluster Routing (continued)

```
procedure NEXT_HOP_FINDER($dest_{addr}$)

    //Get the destination's Context ID
    if ($context_{id} \leftarrow$CONTEXT_ID_LOOKUP($dest_{addr}$)) $= -1$ then
        //Context not found!
        return
    else if $context_{id} = own_{context}$ then
        //Intra-Cluster Routing failure detected!
        return $-1$
    end if
    //Find the most suitable next hop
    if $gw_{entry} \leftarrow$HASH_TABLE_GET($gw_{table}, context_{id}$) then
        //This node is a possible Gateway
        //Get the next hop information from an ordered list
        //(the first occurrence will be the best entry)
        $next\_hop\_tuple \leftarrow$ LIST_ACCESS($gw_{entry}.gateway\_table, 0$)
        $number_{hops} \leftarrow next\_hop\_tuple.hop\_count$
        $next_{hop} \leftarrow next\_hop\_tuple.gateway$
    end if
    if $number_{hops} \neq 1$ then
        //This node is not a GW, or it may not be the best within its context
        //Check if any other node has better connectivity
        if $gw_{entry} \leftarrow$HASH_TABLE_GET($connectivity_{table}, context_{id}$) then
            //Get the next hop information from an ordered list
            //(the first occurrence will be the best entry since it is ordered)
            $next\_hop\_tuple \leftarrow$ LIST_ACCESS($gw_{entry}.gateway\_table, 0$)
            if $next\_hop\_tuple \rightarrow hop\_count < number_{hops}$ then
                $number_{hops} \leftarrow next\_hop\_tuple.hop\_count$
                $next_{hop} \leftarrow next\_hop\_tuple.gateway$
            end if
        end if
    end if

    return $next_{hop}$
end procedure
procedure CONTEXT_ID_LOOKUP($dest_{addr}$)
    if $mapping_{entry} \leftarrow$HASH_TABLE_GET($ip\_context\_mapping_{table}, dest_{addr}$) then
        return $mapping_{entry}.context_{id}$
    else
        return $-1$
    end if
end procedure
procedure ADD_PACKET_RESEND($Packet_{Ptr}$)
    //Save the packet for later resend
    $packet_{entry} \leftarrow$ PMO_ALLOC($Held\_Packets_{pmh}$)
    $packet_{entry}.exp_{time} \leftarrow current\_time + MAX\_TIMEC$
    LIST_INSERT($pending\_packets_{table}, packet_{entry}, TAIL$)
    INTERRUPT_SCHEDULE_CALL($current_{time} + RETRANSMIT\_TIMEC$,
                                              $packet_{entry}, Packet\_Send\_Retry\_Handle$)
end procedure
procedure PACKET_SEND_RETRY_HANDLE($packet_{ptr}$)
    if PACKET_ARRIVAL_HANDLE($packet_{ptr}, TRUE$) $= -1$ and $packet_{ptr}.exp_{time} > current_{time}$ then
        //Route still not available, resend later
        INTERRUPT_SCHEDULE_CALL($2 \times current_{time} + MAX\_TIMEC - packet_{ptr}.exp_{time}$
                              $+RETRANSMIT\_TIMEC, packet_{ptr}, Packet\_Send\_Retry\_Handle$)
    else
        //Route found or entry expired
        LIST_REMOVE($pending\_packets_{table}, packet_{ptr}$)
    end if
end procedure
```

local, since Context aggregation will restrain the number of Contexts affected by a node changing Context. This behaviour is presented in Algorithms 13 and 14.

Algorithm 13 IP-Context Mapping Procedure

```
procedure Ip_Context_Mapping_Create($mapping_{table}, ip_{addr}, context_{id}, neighbour$)
    $new_{view}$ ←Determine_View($context_{id}$)
    if $mapping_{entry}$ ←Hash_Table_Get($mapping_{table}, ip_{addr}$) then
        //Same mapping, nothing to be done
        if $new_{view} = mapping_{entry}.context_{id}$ then
            return
        end if
        if $ip_{addr} = own\_ip_{addr}$ and $own\_context_{id} = context_{id}$
                                    and $context_{id} \neq mapping_{entry}.context_{id}$ then
            //Current node changed context
            for each $mapping_{entry}$ in $mapping_{table}$ do
                //Update each entry to the new context
                $mapping_{entry}.context_{id}$ ←Determine_View($mapping_{entry}.context_{id}$)
            end for
        end if
        if Contains_Context($context_{entry}.context_{id}, own\_context_{id}$)
                                          and $context_{id} = own\_context_{id}$ then
            return $mapping_{entry}$
        else
            $mapping_{entry}.context_{id} \leftarrow new_{view}$
        end if
        //New updated view received!
        if $context_{id} = own\_context_{id}$ and $neighbour$ then
            for each $temp\_mapping_{entry}$ in $temp\_ip\_context\_mapping_{table}$ do
                //Check if this entry is already in the temporary table
                if $temp\_mapping_{entry}$ ←Hash_Table_Get($mapping_{table}, ip_{addr}$) then
                    //Update temporary Context ID
                    $temp\_mapping_{entry}.context_{id} \leftarrow mapping_{entry}.context_{id}$
                    Interrupt_Schedule_Call($current_{time} + IP\_TIMEC, ip_{addr},$
                                          $Temp\_Ip\_Mapping\_Expiry\_Handle$)
                    continue
                end if
                $temp\_mapping_{entry}$ ← Pmo_Alloc($Ip\_Context\_Mapping_{pmh}$)
                $temp\_mapping_{entry}.ip_{addr} \leftarrow mapping\_entry.ip_{addr}$
                $temp\_mapping_{entry}.context_{id} \leftarrow mapping\_entry.context_{id}$
                Interrupt_Schedule_Call($current_{time} + IP\_TIMEC,$
                        $temp\_mapping_{entry}.ip_{addr}, Temp\_Ip\_Mapping\_Expiry\_Handle$)
                Hash_Table_Insert($temp\_ip\_context\_mapping_{table},$
                        $temp\_mapping_{entry}.ip_{addr}, temp\_mapping_{entry}$)
            end for
        else
            //Simple Update
            if not($temp\_mapping_{entry}$ ← Hash_Table_Get($temp\_ip\_context\_mapping_{table}, ip_{addr}$)) then
                $temp\_mapping_{entry}$ ← Pmo_Alloc($Ip\_Context\_Mapping_{pmh}$)
                $temp\_mapping_{entry}.ip_{addr} \leftarrow ip_{addr}$
                Hash_Table_Insert($temp\_ip\_context\_mapping_{table}, ip_{addr}, temp\_mapping_{entry}$)
            end if
            $temp\_mapping_{entry}.context_{id} \leftarrow new_{view}$
            Interrupt_Schedule_Call($current_{time} + IP\_TIMEC,$
                        $ip_{addr}, Temp\_Ip\_Mapping\_Expiry\_Handle$)
        end if
    else
        ... (continues)
    end if
end procedure
```

Context Connectivity information is shared within each cluster and overheard by nodes in different clusters. This information is created by gateway nodes for the clusters they are directly connected with, and for clusters that can be reached through other clusters. Since each GW performs the same way, it is important to check the validity of the information. In order to avoid out-of-order messages, a sequence number and the identifier of the originator GW of such information is used to check if it has already been received, being discarded if so. However, in certain occasions, despite having already been received, information consistent with the existing one may have a more accurate expiry time, due to delays, and should be taken into account as explained in Algorithm 15. These problems do not occur when context connectivity information is obtained from within the same context.

Algorithm 14 IP-Context Mapping Procedure (continued)

```
procedure IP_CONTEXT_MAPPING_CREATE($mapping_{table}, ip_{addr}, context_{id}, neighbour$)
    if $mapping_{entry} \leftarrow$ HASH_TABLE_GET($mapping_{table}, ip_{addr}$) then
        . . . (previous code)
    else
        . . . (continuation)
        //There is no previous knowledge about this IP
        //Allocate memory first and add the new entry
        $temp\_mapping_{entry} \leftarrow$ PMO_ALLOC($Ip\_Context\_Mapping_{pmh}$)
        $temp\_mapping_{entry}.context_{id} \leftarrow new_{view}$
        $temp\_mapping_{entry}.ip_{addr} \leftarrow ip_{addr}$
        HASH_TABLE_INSERT($temp\_ip\_context\_mapping_{table}, ip_{addr}, temp\_mapping_{entry}$)
        INTERRUPT_SCHEDULE_CALL($current_{time} + IP\_TIMEC, ip_{addr},$
                                        $Temp\_Ip\_Mapping\_Expiry\_Handle$)

        //Do the same to the actual mapping table
        $mapping_{entry} \leftarrow$ PMO_ALLOC($Ip\_Context\_Mapping_{pmh}$)
        $mapping_{entry}.context_{id} \leftarrow new_{view}$
        $mapping_{entry}.ip_{addr} \leftarrow ip_{addr}$
        HASH_TABLE_INSERT($mapping_{table}, ip_{addr}, mapping_{entry}$)

        if $neighbour$ then
            //A new node has been found, existing mapping information must be shared
            for each $mapping_{entry}$ in $mapping_{table}$ do
                if ($temp\_mapping_{entry} \leftarrow$ HASH_TABLE_GET($temp\_ip\_context\_mapping_{table}, ip_{addr}$)) then
                    //Reset the existing timer
                    INTERRUPT_SCHEDULE_CALL($current_{time} + IP\_TIMEC, ip_{addr},$
                                                    $Temp\_Ip\_Mapping\_Expiry\_Handle$)
                    continue
                else//Create a new Temp Entry
                    $temp\_mapping_{entry} \leftarrow$ PMO_ALLOC($Ip\_Context\_Mapping_{pmh}$)
                    $temp\_mapping_{entry}.context_{id} \leftarrow mapping_{entry}.context_{id}$
                    $temp\_mapping_{entry}.ip_{addr} \leftarrow mapping_{entry}.ip_{addr}$
                    HASH_TABLE_INSERT($temp\_ip\_context\_mapping_{table}, ip_{addr}, temp\_mapping_{entry}$)
                    INTERRUPT_SCHEDULE_CALL($current_{time} + IP\_TIMEC, ip_{addr},$
                                                    $Temp\_Ip\_Mapping\_Expiry\_Handle$)
                end if
            end for
        end if
    end if
end procedure
```

Algorithm 15 Creation of Context Connectivity

```
procedure Context_Connectivity_Create($connectivity_{table}, ip_{addr}, context_{id}, hop_{count},$
                                      $originator, seq_{num}, metric, exp_{time}$)
    $own_{view} \leftarrow$Determine_View($context_{id}$)
    if $connectivity_{entry} \leftarrow$Hash_Table_Get($connectivity_{table}, own_{view}$) then
        for each $gw_{tuple}$ in $connectivity_{entry}.gateway_{table}$ do
            if $gw_{tuple}.gateway = ip_{addr}$ then
                if $seq_{num} > 0$ and $gw_{tuple}.originator = originator$ then
                    //Discard out of order messages
                    if $gw_{tuple}.seq_{num} > seq_{num}$ or
                            ($gw_{tuple}.seq_{num} = seq_{num}$ and $gw_{tuple}.hop_{count} < hop_{count}$) or
                                ($gw_{tuple}.seq_{num} = seq_{num}$ and $gw_{tuple}.hop_{count} = hop_{count}$
                                    and $gw_{tuple}.exp_{time} \leq exp_time$) then
                        return
                    end if
                end if
                $gw_{tuple}.exp_{time} \leftarrow exp_{time}$
                $gw_{tuple}.hop_{count} \leftarrow hop_{count}$
                $gw_{tuple}.seq_{num} \leftarrow seq_{num}$
                $gw_{tuple}.originator \leftarrow originator$
                $gw_{tuple}.metric \leftarrow metric$
                Interrupt_Schedule_Call($exp_{time}, gw_{tuple},$
                                        $Context\_Connectivity\_Expiry\_Handle$)
                $ordered_{position} \leftarrow$ Connectivity_InsertSort($ip_{addr}, hop_{count}, metric$)
                //Re-insert this entry in an ordered position
                List_Insert($connectivity_{entry}.gateway_{table}, gw_{tuple}, ordered_{position}$)
                return
            end if
        end for
        //There is still no entry for this GW
        $gw_{tuple} \leftarrow$ Pmo_Alloc($Gw\_Connectivity\_Tuple_{pmh}$)
        $gw_{tuple}.gateway \leftarrow ip_{addr}$
        $gw_{tuple}.hop_{count} \leftarrow hop_{count}$
        $gw_{tuple}.seq_{num} \leftarrow seq_{num}$
        $gw_{tuple}.originator \leftarrow originator$
        $gw_{tuple}.metric \leftarrow metric$
        $gw_{tuple}.exp_{time} \leftarrow exp_{time}$
        Interrupt_Schedule_Call($exp_{time}, gw_{tuple},$
                                $Context\_Connectivity\_Expiry\_Handle$)
        $ordered_{position} \leftarrow$ Connectivity_InsertSort($ip_{addr}, hop_{count}, metric$)
        List_Insert($connectivity_{entry}.gateway_{table}, gw_{tuple}, ordered_{position}$)
    else
        //There is still no entry for this Context
        $connectivity_{entry} \leftarrow$ Pmo_Alloc($Context\_Connectivity\_Tuple_{pmh}$)
        $connectivity_{entry}.gateway_{table} \leftarrow$ List_Create()
        $gw_{tuple} \leftarrow$ Pmo_Alloc($Gw\_Connectivity\_Tuple_{pmh}$)
        $gw_{tuple}.gateway \leftarrow ip_{addr}$
        $gw_{tuple}.hop_{count} \leftarrow hop_{count}$
        $gw_{tuple}.seq_{num} \leftarrow seq_{num}$
        $gw_{tuple}.originator \leftarrow originator$
        $gw_{tuple}.metric \leftarrow metric$
        $gw_{tuple}.exp_{time} \leftarrow exp_{time}$
        Interrupt_Schedule_Call($exp_{time}, gw_{tuple},$
                                $Context\_Connectivity\_Expiry\_Handle$)
        List_Insert($connectivity_{entry}.gateway_{table}, gw_{tuple}, TAIL$)
        Hash_Table_Insert($connectivity_{table}, own_{view}, connectivity_{entry}$)
    end if
end procedure
```

A node becomes a Gw when it receives routing messages from different Contexts, announcing it to all other nodes in its own Context, by creating new Context Connectivity information, as previously seen. New gateways must also create their own information database of foreign gateways to which they can connect. This information is named Gateway Connectivity and its kept for each gateway, giving origin to the Context Connectivity information, as seen in Algorithms 16, 17 and 18.

Algorithm 16 Creation of Gateway Connectivity

```
procedure GW_CONNECTIVITY_CREATE($connectivity_{table}, ip_{addr}, context_{id},$
        $hop_{count}, originator, seq_{num}, exp_{time}$)
    $own_{view} \leftarrow$DETERMINE_VIEW($context_{id}$)
    //Check the Context Connectivity Table to avoid already existing information
    if $connectivity_{entry} \leftarrow$HASH_TABLE_GET($context\_connectivity_{table}, own_{view}$) then
        for each $gw_{tuple}$ in $connectivity_{entry}.gateway_{table}$ do
            if $seq_{num} < 0$ and $number_{hops} = 1$ then //This node is a direct gateway
                break
            end if
            if $gw_{tuple}.originator = originator$ then
                if $gw_{tuple}.seq_{num} \geq seq_{num}$ and $gw_{tuple}.number_{hops} < number_{hops}$ then
                    //This connectivity information was already known
                    return
                end if
            end if
        end for
    end if

    if $connectivity_{entry} \leftarrow$HASH_TABLE_GET($connectivity_{table}, own_{view}$) then
        for each $gw_{tuple}$ in $connectivity_{entry}.gateway_{table}$ do
            if $gw_{tuple}.gateway = ip_{addr}$ then
                if $seq_{num} > 0$ and $gw_{tuple}.originator = originator$ then
                    //Discard out of order messages
                    if $gw_{tuple}.seq_{num} > seq_{n}um$ or
                            ($gw_{tuple}.seq_{num} = seq_{num}$ and $gw_{tuple}.hop_{count} < hop_{count}$) or
                                ($gw_{tuple}.seq_{num} = seq_{num}$ and $gw_{tuple}.hop_{count} = hop_{count}$
                                    and $gw_{tuple}.exp_{time} \leq exp_{t}ime$) then
                        return
                    end if
                end if
                $gw_{tuple}.exp_{time} \leftarrow exp_{time}$
                $gw_{tuple}.hop_{count} \leftarrow hop_{count}$
                $gw_{tuple}.seq_{num} \leftarrow seq_{num}$
                $gw_{tuple}.originator \leftarrow originator$
                INTERRUPT_SCHEDULE_CALL($exp_{time}, gw_{tuple}, Context\_Gw\_Expiry\_Handle$)
                ...
            end if
        end for
        ...
    end if
end procedure
```

When a Gw is created it should share its connectivity with the remaining nodes in its own Context. However, until a routing message containing this information is sent, it must not be considered for routing purposes. For this reason, a new temporary Context Connectivity entry is kept in a separate table, as detailed in Algorithm 19.

Algorithm 17 Creation of Gateway Connectivity (continued)

```
procedure Gw_Connectivity_Create($connectivity_{table}, ip_{addr}, context_{id},$
                                 $hop_{count}, originator, seq_{num}, exp_{time}$)
    if $connectivity_{entry} \leftarrow$Hash_Table_Get($connectivity_{table}, own_{view}$) then
        ...
        for ... do
            if ... then
                if $gw_{tuple}.last_{u}pdate \leq 0$ then
                    $gw_{tuple}.total_{u}pdate \leftarrow 6$
                    $gw_{tuple}.number_{u}pdate \leftarrow 1$
                    $gw_{tuple}.last_{u}pdate \leftarrow current_{time}$
                else if $current_{time} - gw_{tuple}.last_{u}pdate < hello\_interval - 0.5$ then
                    //This can never happen unless the HELLO message
                    //has been received by piggyback (link_chain HELLO messages)
                    $gw_{tuple}.last_{u}pdate \leftarrow current_{time}$
                else
                    $gw_{tuple}.total_{u}pdate \leftarrow gw_{tuple}.total_{u}pdate + current_{time} - gw_{tuple}.last_{u}pdate$
                    $gw_{tuple}.number_{u}pdate \leftarrow gw_{tuple}.number_{u}pdate + 1$
                    $update\_averages \leftarrow gw_{tuple}.total_{u}pdate / gw_{tuple}.number_{u}pdate$
                    $error_{estimated} \leftarrow$ Kernel_Fit($update\_averages$)
                    $error_{current} \leftarrow$ Kernel_Fit($current_{time} - gw_{tuple}.last_{u}pdate$)
                end if
                if $current\_error > estimated\_error$ then
                    $gw_{tuple}.metric \leftarrow NODROPC\_METRIC \times (.6 \times error_{current} + .4 \times error_{estimated})$
                else
                    $gw_{tuple}.metric \leftarrow NODROPC\_METRIC \times (.3 \times error_{current} + .7 \times error_{estimated})$
                end if
                //Update the list and then insert this new entry
                $ordered_{position} \leftarrow$ Gw_InsertSort($gateway_{tuple}.gateway, gateway_{tuple}.hop_{count},$
                                        $gw_{tuple}.metric, connectivity_{entry}.gateway_{table}$)
                //Update info to the best GW (may have changed)
                if ($curr_{position} = 0$ and $ordered_{position} = 0$) then
                    Temp_Context_Connectivity_Create($temp\_context\_connectivity_{table}, own_{addr},$
                        $context_{entry}.context_{id}, gw_{tuple}.hop_{tuple}, gw_{tuple}.originator, gw_{tuple}.seq_{num},$
                            $gw_{tuple}.metric, gw_{tuple}.exp_{time} - GW\_TIMEC + CONNECTIVITY\_TIMEC$)
                end if
                //Re-insert this entry in an ordered position
                List_Insert($connectivity_{entry}.gateway_{table}, gw_{tuple}, ordered_{position}$)
                return
            end if
        end for
        //There is still no entry for this Gw
        $gw_{tuple} \leftarrow$ Pmo_Alloc($Gw\_Connectivity\_Tuple_{pmh}$)
        $gw_{tuple}.gateway \leftarrow ip_{addr}$
        $gw_{tuple}.hop_{count} \leftarrow hop_{count}$
        $gw_{tuple}.seq_{num} \leftarrow seq_{num}$
        $gw_{tuple}.originator \leftarrow originator$
        $gw_{tuple}.metric \leftarrow NODROPC\_METRIC$ //The new Gw starts worst value possible
        $gw_{tuple}.total_{u}pdate \leftarrow 6$
        $gw_{tuple}.number_{u}pdate \leftarrow 1$
        $gw_{tuple}.last_{u}pdate \leftarrow current_{time}$
        $gw_{tuple}.exp_{time} \leftarrow exp_{time}$
        Interrupt_Schedule_Call($exp_{time}, gw_{tuple}, Context\_Gw\_Expiry\_Handle$)
        $ordered_{position} \leftarrow$ Gw_InsertSort($gateway_{tuple}.gateway, gateway_{tuple}.hop_{count},$
                                $gw_{tuple}.metric, connectivity_{entry}.gateway_{table}$)
        List_Insert($connectivity_{entry}.gateway_{table}, gw_{tuple}, ordered_{position}$)
        if $ordered_{position} = 0$ then
            Temp_Context_Connectivity_Create($temp\_context\_connectivity_{table}, own_{addr},$
                $context_{entry}.context_{id}, gw_{tuple}.hop_{tuple}, gw_{tuple}.originator, gw_{tuple}.seq_{num},$
                    $gw_{tuple}.metric, gw_{tuple}.exp_{time} - GW\_TIMEC + CONNECTIVITY\_TIMEC$)
        end if
    else
        //There is still no GW entry for this Context
        ...
    end if
end procedure
```

Algorithm 18 Creation of Gateway Connectivity (continued)

```
procedure GW_CONNECTIVITY_CREATE($connectivity_{table}, ip_{addr}, context_{id}, hop_{count},$
                                  $originator, seq_{num}, exp_{time}$)
    if $connectivity_{entry} \leftarrow$HASH_TABLE_GET($connectivity_{table}, own_{view}$) then
        ...
    else
        //There is still no GW entry for this Context
        $connectivity_{entry} \leftarrow$ PMO_ALLOC($Context\_Connectivity\_Tuple_{pmh}$)
        $connectivity_{entry}.gateway_{table} \leftarrow$ LIST_CREATE()

        $gw_{tuple} \leftarrow$ PMO_ALLOC($Gw\_Connectivity\_Tuple_{pmh}$)
        $gw_{tuple}.gateway \leftarrow ip_{addr}$
        $gw_{tuple}.hop_{count} \leftarrow hop_{count}$
        $gw_{tuple}.seq_{num} \leftarrow seq_{num}$
        $gw_{tuple}.originator \leftarrow originator$
        $gw_{tuple}.metric \leftarrow NODROPC\_METRIC$ //The new Gw starts worst value possible
        $gw_{tuple}.total_{u}pdate \leftarrow 6$
        $gw_{tuple}.number_{u}pdate \leftarrow 1$
        $gw_{tuple}.last_{u}pdate \leftarrow current_{time}$
        $gw_{tuple}.exp_{time} \leftarrow exp_{time}$
        INTERRUPT_SCHEDULE_CALL($exp_{time}, gw_{tuple}, Context\_Gw\_Expiry\_Handle$)
        LIST_INSERT($connectivity_{entry}.gateway_{table}, gw_{tuple}, TAIL$)
        HASH_TABLE_INSERT($connectivity_{table}, own_{view}, connectivity_{entry}$)
        TEMP_CONTEXT_CONNECTIVITY_CREATE($temp\_context\_connectivity_{table}, own_{addr},$
                $context_{entry}.context_{id}, gw_{tuple}.hop_{tuple}, gw_{tuple}.originator, gw_{tuple}.seq_{num},$
                    $gw_{tuple}.metric, gw_{tuple}.exp_{time} - GW\_TIMEC + CONNECTIVITY\_TIMEC$)
    end if
end procedure
```

Whenever this information is propagated, the temporary entry becomes definitive and is used in the inter-cluster routing process.

The received IP Context mapping information is important to determine a packet's destination Context and is kept indeterminately or until it is updated. However, this information is not sent in every *HELLO* message and only when a change occurs (i.e. a node changes Context), will it be included in a temporary table, in which every entry expires after a certain amount of time. Similarly, all the received Context entries are kept only for a certain amount of time, after which they are re-inserted in the end of their list, marked as expired and eventually being removed if no new update arrives. By keeping the last sequence number, this information is not immediately removed, in order to avoid receiving duplicate information. The expiry handle procedures take care of these processes of maintaining or removing the obtained information.

The remaining algorithms that were not detailed are "InsertSort" and "kernel_fit" procedures. The specificities of the "InsertSort" were omitted because they only calculate the position in which a new entry should be positioned according to its metric. This can be achieved in several different ways and different metrics can also be used. Regarding the details of the "kernel_fit" procedure, they are presented in Chapter 4 where a link-quality estimator is defined. This procedure uses the interval time between

Algorithm 19 Creation of Temporary Context Connectivity

```
procedure TEMP_CONTEXT_CONNECTIVITY_CREATE($connectivity_{table}, ip_{addr}, context_{id}, hop_{count},$
                                            $originator, seq_{num}, metric, exp_{time}$)
    $own_{view}$ ←DETERMINE_VIEW($context_{id}$)
    if $connectivity_{entry}$ ←HASH_TABLE_GET($connectivity_{table}, own_{view}$) then
        for each $gw_{tuple}$ in $connectivity_{entry}.gateway_{table}$ do
            if $gw_{tuple}.gateway = ip_{addr}$ then
                if $seq_{num} > 0$ and $gw_{tuple}.originator = originator$ then
                    //Discard out of order messages
                    if $gw_{tuple}.seq_{num} > seq_{num}$ or
                            ($gw_{tuple}.seq_{num} = seq_{num}$ and $gw_{tuple}.hop_{count} < hop_{count}$) or
                                ($gw_{tuple}.seq_{num} = seq_{num}$ and $gw_{tuple}.hop_{count} = hop_{count}$
                                    and $gw_{tuple}.exp_{time} \leq exp_{t}ime$) then
                        return
                    end if
                end if
                $gw_{tuple}.exp_{time} \leftarrow exp_{time}$
                $gw_{tuple}.hop_{count} \leftarrow hop_{count}$
                $gw_{tuple}.seq_{num} \leftarrow seq_{num}$
                $gw_{tuple}.originator \leftarrow originator$
                $gw_{tuple}.metric \leftarrow metric$
                INTERRUPT_SCHEDULE_CALL($exp_{time}, gw_{tuple},$
                                        $Context\_Connectivity\_Expiry\_Handle$)
                //The position is not relevant
                LIST_INSERT($connectivity_{entry}.gateway_{table}, gw_{tuple}, TAIL$)
            end if
        end for
        //There is still no entry for this GW
        $gw_{tuple} \leftarrow$ PMO_ALLOC($Gw\_Connectivity\_Tuple_{pmh}$)
        $gw_{tuple}.gateway \leftarrow ip_{addr}$
        $gw_{tuple}.hop_{count} \leftarrow hop_{count}$
        $gw_{tuple}.seq_{num} \leftarrow seq_{num}$
        $gw_{tuple}.originator \leftarrow originator$
        $gw_{tuple}.metric \leftarrow metric$
        $gw_{tuple}.exp_{time} \leftarrow exp_{time}$
        INTERRUPT_SCHEDULE_CALL($exp_{time}, gw_{tuple},$
                                $Context\_Connectivity\_Expiry\_Handle$)
        LIST_INSERT($connectivity_{entry}.gateway_{table}, gw_{tuple}, TAIL$)
    else
        //There is still no entry for this Context
        $connectivity_{entry} \leftarrow$ PMO_ALLOC($Context\_Connectivity\_Tuple_{pmh}$)
        $connectivity_{entry}.gateway_{table} \leftarrow$ LIST_CREATE()

        $gw_{tuple} \leftarrow$ PMO_ALLOC($Gw\_Connectivity\_Tuple_{pmh}$)
        $gw_{tuple}.gateway \leftarrow ip_{addr}$
        $gw_{tuple}.hop_{count} \leftarrow hop_{count}$
        $gw_{tuple}.seq_{num} \leftarrow seq_{num}$
        $gw_{tuple}.originator \leftarrow originator$
        $gw_{tuple}.metric \leftarrow metric$
        $gw_{tuple}.exp_{time} \leftarrow exp_{time}$
        INTERRUPT_SCHEDULE_CALL($exp_{time}, gw_{tuple},$
                                $Context\_Connectivity\_Expiry\_Handle$)
        LIST_INSERT($connectivity_{entry}.gateway_{table}, gw_{tuple}, TAIL$)
        HASH_TABLE_INSERT($connectivity_{table}, own_{view}, connectivity_{entry}$)
    end if

end procedure
```

HELLO messages to determine the quality of the considered wireless link, according to the estimated percentage of packet loss, returning a value between 1, when too many packets are lost, and 0 for optimal link quality with no losses.

A.2 Summary

A detailed specification of the algorithms required to implement Deferred Routing were presented in this Chapter. These algorithms resulted from the implementation of the protocol in the OPNET simulator and were the basis for the performance evaluation previously provided. In the existing implementation the OLSR protocol was used for intra-cluster routing as the link-state protocol required by DefeR. However, any other proactive routing protocol could be used.