
THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

# TOWARDS TAILORED AND ADAPTIVE WIRELESS MULTI-HOP ROUTING PROTOCOLS

Saaidal Razalli Bin Azzuhri
B.Eng (Telecommunications), M.Sc (Information Technology)

*A thesis submitted for the degree of Doctor of Philosophy at*
*The University of Queensland in 2013*
School of Information Technology & Electrical Engineering (ITEE)

**Abstract**

Wireless Mesh Networks (WMNs) and wireless multi-hop networks in general have seen a tremendous development over the past couple of decades. Their independence from a wired backbone network, which allows relatively rapid and low cost deployment, combined with their self-configuring and self-healing capabilities and flexibility, make them suitable for deployment in a wide range of situations. These application scenarios include communications for rural communities, agriculture, natural disaster recovery, automatic (electrical) meter reading etc.

Routing protocols are a critical component for wireless multi-hop networks, and determine to a large extent the network performance. The problem is that current protocols are largely *one-size-fits-all*, and have a fixed set of protocol mechanisms and protocol parameters. This makes it impossible for these protocols to perform equally well in all the possible deployment scenarios, under very different network characteristics, such as network size, network topology, level of node mobility and traffic patterns.

To address this shortcoming, this thesis explores how wireless multi-hop routing protocols can be adapted and tailored towards their specific deployment scenario. Towards this goal, the thesis explores the impact of choosing specific protocol mechanisms and protocol parameters on the performance of wireless multi-hop networks, under different network scenarios. The thesis further presents a new hybrid protocol, which combines end-to-end routing of traditional wireless mobile ad-hoc and networks, with the store-carry-forward routing paradigm of Delay Tolerant Networks (DTNs). An extensive evaluation over a wide range topologies, from highly connected to highly disconnected, shows that this protocol can significantly improve the performance in most cases.

**Declaration by author**

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

I have clearly stated the contribution of others to my thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of my research higher degree candidature and does not include a substantial part of work that has been submitted to qualify for the award of any other degree or diploma in any university or other tertiary institution. I have clearly stated which parts of my thesis, if any, have been submitted to qualify for another award.

I acknowledge that an electronic copy of my thesis must be lodged with the University Library and, subject to the General Award Rules of The University of Queensland, immediately made available for research and study in accordance with the *Copyright Act 1968*.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate I have obtained copyright permission from the copyright holder to reproduce material in this thesis.

**Publications during candidature**

Saaidal R. Azzuhri, Marius Portmann, Wee Lum Tan. *Adaptive Wireless Mesh Networks Routing Protocols*. (2010) 7th International Conference on Ubiquitous Intelligence & Computing and 7th International Conference on Automatic & Trusted Computing (UIC/ATC), 142-147.

Saaidal R. Azzuhri, Marius Portmann, Wee Lum Tan. *Evaluation of Parameterised Route Repair in AODV* (2010) 4th International Conference on Signal Processing and Communication Systems (ICSPCS), 1-7.

Saaidal R. Azzuhri, Marius Portmann, Wee Lum Tan. *Evaluating the Performance Impact of Protocol Parameters on Ad-Hoc Network Routing Protocols*. (2012) 2012 Australasian Telecommunication Networks and Application Conference (ATNAC), 1-6.

Ranjana Pathak, Peizhao Hu, Jadwiga Indulska, Marius Portmann, Saaidal R. Azzuhri. *A Performance Study of Hybrid Protocols for Opportunistic Communications*. (2013) 9th IEEE International Workshop on Performance and Management of Wireless and Mobile Networks (P2MNET), 2013.

Submitted:

Saaidal R. Azzuhri, Peizhao Hu, Jadwiga Indulska, Marius Portmann, Ranjana Pathak. *OLSR-Opportunistic: Towards a Better Approach of Hybrid Protocols in Multihop Wireless Networks.* (2013) submitted to Malaysian Journal of Computer Science (MJCS)

**Publications included in this thesis**

Saaidal R. Azzuhri, Marius Portmann, Wee Lum Tan. *Adaptive Wireless Mesh Networks Routing Protocols*. (2010) 7th International Conference on Ubiquitous Intelligence & Computing and 7th International Conference on Automatic & Trusted Computing (UIC/ATC), 142-147 - Incorporated as Chapter 3.

| Contributor | Statement of contribution |
|---|---|
| Author Saaidal R. Azzuhri (Candidate) | Designed experiments (90%)<br>Conducted experiments (100%)<br>Wrote the paper (80%) |

| | |
|---|---|
| Author Marius Portmann | Designed experiments (10%)<br>Wrote and edited paper (10%) |
| Author Wee Lum Tan | Wrote and edited paper (10%) |

Saaidal R. Azzuhri, Marius Portmann, Wee Lum Tan. *Evaluation of Parameterised Route Repair in AODV* (2010) 4th International Conference on Signal Processing and Communication Systems (ICSPCS), 1-7 - Incorporated as Chapter 5.

| Contributor | Statement of contribution |
|---|---|
| Author Saaidal R. Azzuhri (Candidate) | Designed experiments (60%)<br>Conducted experiments (100%)<br>Wrote the paper (70%) |
| Author Marius Portmann | Designed experiments (20%)<br>Wrote and edited paper (20%) |
| Author Wee Lum Tan | Designed experiments (20%)<br>Wrote and edited paper (10%) |

Saaidal R. Azzuhri, Marius Portmann, Wee Lum Tan. *Evaluating the Performance Impact of Protocol Parameters on Ad-Hoc Network Routing Protocols*. (2012) 2012 Australasian Telecommunication Networks and Application Conference (ATNAC), 1-6 - Incorporated as Chapter 5.

| Contributor | Statement of contribution |
|---|---|
| Author Saaidal R. Azzuhri (Candidate) | Designed experiments (60%)<br>Wrote the paper (70%) |
| Author Marius Portmann | Designed experiments (20%)<br>Wrote and edited paper (20%) |
| Author Wee Lum Tan | Designed experiments (20%)<br>Wrote and edited paper (10%) |

Ranjana Pathak, Peizhao Hu, Jadwiga Indulska, Marius Portmann, Saaidal R. Azzuhri. *A Performance Study of Hybrid Protocols for Opportunistic Communications*. (2013) 9th IEEE International Workshop on Performance and Management of Wireless and Mobile Networks (P2MNET), 2013 - Incorporated as Chapter 6.

| Contributor | Statement of contribution |
|---|---|
| Author Ranjana Pathak | Designed experiments (60%)<br>Wrote the paper (50%) |
| Author Peizhao Hu | Designed experiments (30%)<br>Wrote and edited paper (30%) |
| Author Jadwiga Indulska | Wrote and edited paper (10%) |
| Author Marius Portmann | Wrote and edited paper (10%) |
| Author Saaidal R. Azzuhri (Candidate) | Designed experiments (10%)<br>Provided OLSR-OPP protocol (100%) |

Saaidal R. Azzuhri, Peizhao Hu, Jadwiga Indulska, Marius Portmann, Ranjana Pathak. *OLSR-Opportunistic: Towards a Better Approach of Hybrid Protocols in Multihop Wireless Networks.* (2013) submitted to Malaysian Journal of Computer Science (MJCS) – Incorporated as Chapter 6

| Contributor | Statement of contribution |
|---|---|
| Author Saaidal R. Azzuhri (Candidate) | Designed experiments (80%)<br>Wrote the paper (70%) |
| Author Peizhao Hu | Designed experiments (10%)<br>Wrote and edited paper (10%) |
| Author Jadwiga Indulska | Wrote and edited paper (10%) |
| Author Marius Portmann | Wrote and edited paper (10%) |
| Author Ranjana Pathak | Designed experiments (10%) |

**Contribution by others to the thesis**

Assistant of statistical analysis, interpretation of results and protocol development was provided by Associate Prof Dr. Marius Portmann and NICTA's Network Group (Dr. Peizhao Hu, Dr. Wee Lum Tan and Ms. Ranjana Pathak)

**Statements of parts of the thesis submitted to qualify for the award of another degree**

None

**Acknowledgements**

This study was undertaken whilst in receipt of a Malaysian Public Service Department Scholarship Award, and I gratefully acknowledge the facilities provided by the School of Information Technology & Electrical Engineering, at The University of Queensland. In addition, my research was also funded by NICTA, Australia's Information and Communications Technology (ICT) Research Centre of Excellence.

I would like to expressly acknowledge a number of people who were instrumental in this work and the preparation of this thesis. Firstly, I would like to graciously thank my supervisor Associate Professor Marius Portmann, for inspiring me and for offering me such an exciting research project. Secondly, I would like to thank all the members of the NICTA Queensland Networking Research Group, who made the hard times more tolerable by offering friendship and support through various means. In particular, I would like to acknowledge Dr. Wee Lum Tan and Dr Peizhao Hu, for helping me in my research and extensively guiding me throughout my research. Special thanks also go to my colleagues Mr. Wei Yin and Miss Ranjana Pathak for helping me with programming aspects of my research. I would also like to thank my associate supervisor Professor Jadwiga Indulska for offering me very sound advice on a range of aspects related to my thesis.

On a more personal side, I would like to thank a list of incredibly special people who helped me survive my PhD without completely losing my mind. My amazing mother Wan Halimah who provided continual moral support and always prayed for my health and well being. My brilliant fiancé Sarina Jafrahim that always knew how to cheer me up during my PhD study.

And finally, my appreciation goes to all my siblings (Ahmad Faizal, Aida Yasmin, Mohamed Reza, Aida Alya and Aida Suraya) for the fantastic support during this tough time.

**Keywords**

wireless multi-hop networks, wireless mesh networks, delay tolerant networks, routing protocols

**Australian and New Zealand Standard Research Classifications (ANZSRC)**

ANZSRC code: 080502, Mobile Technologies, 50%
ANZSRC code: 080503, Networking and Communications, 40%
ANZSRC code: 080504, Ubiquitous Computing, 10%

**Field of Research (FoR) Classification**

FoR code: 1005 , Communications Technologies, 100%

## Table of Contents

## List of Figures

## List of Tables

**List of Abbreviations**

| Abb | Description |
|---|---|
| ACK | Acknowledgement |
| ADV | Adaptive Distance Vector |
| AHI | AODV Hello Interval |
| AHR | Adaptive Hello Rate |
| AODV | Ad-Hoc On Demand Distance Vector |
| AOLSR | Adaptive OLSR |
| ARM-DSDV | Adapting to Route Demand and Mobility - DSDV |
| ART | Active Route Timeout |
| BATMAN | Better Approach To Mobile Ad-Hoc Network |
| BBR | Bubble Rap Routing |
| CAR | Context Aware Routing |
| CDF | Cumulative Distribution Function |
| CML | Chameleon |
| CPU | Central Processing Unit |
| DOA | DSR Over AODV |
| DSDV | Destination Sequenced Distance Vector |
| DSR | Dynamic Source Routing |
| DT-DYMO | Delay Tolerant DYMO |
| DTN | Delay Tolerant Networks |
| DYMO | Dynamic MANET On Demand |
| FHI | Fast Hello Interval |
| GAB | Global Association Based |
| HD | High Dynamic |
| HI | Hello Interval |
| HWMP | Hybrid Wireless Mesh Protocol |
| HYMAD | Hybrid MANET-DTN |
| IEEE | Institute of Electrical and Electronics Engineers |
| IETF | Internet Engineering Task Force |
| IFQ | Interface Queue |
| LAB | Link Association Based |
| LABA | Local Association Based Advertisement |

| | |
|---|---|
| LA-OLSR | Link Aware Optimized Link State Routing |
| LD | Low Dynamic |
| MAC | Media Access Control |
| MANET | Mobile Ad-Hoc Networks |
| MASP | Mobility Adaptive Self Parameterization |
| MPR | Multi Point Relay |
| NECTAR | Neighbourhood Contact History |
| NOMAD | Native OLSR For Mobile Ad-Hoc and Disrupted |
| ns-2 | Network Simulator 2 |
| OHI | OLSR Hello Interval |
| OLSR | Optimized Link State Routing |
| OLS-OPP | OLSR Opportunistic |
| PAODV | Prediction AODV |
| PD | Partitioning Degree |
| PDR | Packet Delivery Ratio |
| PROPHET | Probabilistic Routing Using History of Encounters and Transitivity |
| RAPID | Resource Allocation Protocol for Intentional DTN |
| RERR | Route Error |
| RFC | Request For Comments |
| RM-AODV | Radio Aware Metric – Ad-Hoc On Demand Distance Vector |
| RA-OLSR | Radio Aware – Optimized Link State Routing |
| RREP | Route Reply |
| RREQ | Route Request |
| RTT | Round Trip Time |
| RWP | Random Waypoint |
| SAW | Spray and Wait |
| SCF | Store Carry Forward |
| SF-BATMAN | Store Forward BATMAN |
| SHARP | Sharp Hybrid Adaptive Routing |
| TC | Topology Control |
| TCI | Topology Control Interval |
| Thr | Threshold |
| TLF | Time To Link Failure |
| TLW | Time Without Link Changes |

| | |
|---|---|
| TTL | Time To Live |
| TWC | Time Without Link Changes |
| Wifi | Wireless Fidelity |
| WLAN | Wireless Local Area Network |
| WMN | Wireless Mesh Networks |
| WPR | Way Point Routing |

# Chapter 1 INTRODUCTION AND MOTIVATION

## 1.1 Overview

Wireless multi-hop networks, also referred to as wireless ad-hoc or mesh networks have seen a tremendous development over the past couple of decades. (Unless specifically mentioned otherwise, the generic term Wireless Mesh Networks (WMNs) is used in this thesis for any wireless multi-hop networks where end-to-end routes are required at the time of packet forwarding, as opposed to wireless multi-hop networks using the store-carry-forward approach, such as Delay Tolerant Networks (DTNs) discussed further below.)

Wireless Mesh Networks independence from a wired backbone network, which allows relatively rapid and low cost deployment, combined with their self-configuring and self-healing capabilities and flexibility, make them suitable for deployment in a wide range of situations. These application scenarios include communications for rural communities [16], [17],[18], agriculture [19], natural disaster recovery [20], military deployments [21], automatic (electrical) meter reading [22], railway networks [15] and mining [23], [24]. The wide range of deployment scenarios makes it challenging to design network protocols that perform equally well in all cases.

Another type of wireless multi-hop network is Delay Tolerant Networks (DTNs), or opportunistic networks. In contrast to traditional Wireless Mesh Networks, where an end-to-end path between source and destination nodes is established and available while the communication session is ongoing, in DTNs an end-to-end path typically does not exist at a single point in time, due to sparseness of the network and corresponding low node density. In DTNs, packets are forwarded using the store-carry-forward paradigm, exploiting the mobility of nodes and packet exchanges during opportunistic encounters between pairs of nodes WMNs [67].

An essential component of any wireless multi-hop network is the routing protocol, which determines the path of packets taken from source to destination nodes. The quality of the end to end path established by the routing protocols is a critical factor that determines the overall performance of the network.

There has been extensive research carried out to develop efficient and reliable routing protocols for Wireless Mesh Networks. While this work has been initially focussed on Mobile Ad-hoc Networks (MANETs), where networks consist mostly of mobile client devices, more recently, this has been extended to more generic Wireless Mesh Networks, including networks with dedicated infrastructure nodes and no or very limited node mobility. Several wireless routing protocols have been developed to provide communication in a wireless ad-hoc or mesh environments. Key examples of such protocols include AODV [2], OLSR [3], DSDV [27], DYMO [28], DSR [29], and TORA [30]. There has also been a large number of works that have evaluated and compared the performance of such routing protocols, such as [35], [36] or [37].

Similarly, there has been extensive research into routing protocols for Delay Tolerant Networks. Epidemic [79], Spray and Wait [80] and PROPHET [83] are key examples of such DTN routing protocols. The research into routing protocols for DTNs has mostly been separate from the research into routing protocols for more traditional wireless multi-hop networks such as Wireless Mesh Networks. In contrast to this traditional approach, one of the key contributions of this thesis is the exploration of a protocol that can perform well in both WMN and DTN environments and adaptively chooses the most suitable routing paradigm.

## 1.2 Research Challenges and Ideas

Most research into routing protocols for wireless multi-hop networks so far, including both WMNs and DTNs, has focussed on developing protocols with a fixed, given routing behaviour, and a fixed set of critical protocol parameters. The problem is that possible deployment scenarios of wireless multi-hop networks can vary significantly in regards to a wide range of parameters and characteristics, such as network size, network topology and node density, mobility pattern, traffic pattern etc. It is clear that there can be no single routing protocol that can perform optimally in all these, potentially very different network scenarios. Current wireless multi-hop routing protocols are largely *one-size-fits-all*, and are unable to adapt their operation to specific network and deployment scenarios. The goal of this thesis is to explore the potential of protocol adaptation, where the wireless multi-hop routing protocols can be adapted to different network scenarios and tailored in terms of their behaviour.

An example of such protocol adaptation is by tuning of critical routing protocol parameters, in particular timing parameters such as HELLO intervals, Active Route Timeout parameters etc. For example, the AODV routing protocol has more than 20 protocol parameters that are set at a fixed value, according to the RFC [2], and can potentially be used for adaptation.

There has been very limited research into the potential of wireless multi-hop routing protocol adaptation, and significant research challenges remain to be addressed, such as which aspects of the protocol or which protocol parameters offer the greatest potential for performance improvement via adaptation and tailoring to different network situations. The aim of this thesis is to take a step towards addressing these challenges.

To this end, this thesis explores how the choice of critical routing protocol parameters affects the network performance for different network scenarios. This evaluation and corresponding results are presented in Chapter 5, where the performance of protocols such as AODV, DYMO (AODVv2) and OLSR with a combination of different protocol parameter settings under a range of different network scenarios are explored via simulation experiments.

One of the findings of Chapter 5 is that the node density, and hence network connectivity, has a critical impact on network performance for traditional ad-hoc and WMN routing protocols such as OLSR, AODV and others. Once the node density goes below a certain threshold, and the network becomes increasingly sparse, end-to-end routes such as established by protocols such as AODV and OLSR become increasingly fragile. Consequently, the network performance decreases dramatically. A highly sparse network topology and general lack of end-to-end routes are characteristics of Delay Tolerant Networks. The thesis explores in Chapter 6 how the performance of WMN routing protocol can be enhanced in such scenarios, by incorporating the DTN concept of store-carry-forward into a WMN routing protocol. This is explored using the OLSR protocol as a basis.

This is in contrast to the traditional research into wireless routing protocols, which has a binary view of wireless multi-hop networks, i.e. either as WMNs (end-to-end routes are expected to be available all the time) or DTNs or opportunistic networks (end-to-end routes are not expected to be available). This traditional approach ignores the significant ‘grey-zone’ between those two boundary cases, which can occur in practical situations. The goal of this

thesis was to develop a simple and practical protocol which can perform well across a wide range of degree of network connectivity. The protocol developed and evaluated in this thesis is based on OLSR, and is called OLSR-OPP, for OLSR with Opportunistic Network extensions. The performance of OLSR-OPP is evaluated across a wide range of levels of network connectivity (or node densities), and shown to significantly improve performance over the original OLSR protocol, as well as Spray-and-Wait, a well-known DTN routing protocol.

## 1.3 Summary of Research Contributions

The key research contributions of this thesis are summarised below:

- Experimental evaluation of impact of WMN routing protocol parameters and mechanisms on network performance, under a range of network scenarios
  - Evaluation of a parameterised route repair mechanism in AODV
  - Investigation of performance of key WMN protocols (AODV, DYMO, OLSR, and HWMP) for a range of mobility scenarios, and investigation of reasons for packet loss
  - Investigation of the impact of HELLO interval parameter on network performance in AODV, under a range of mobility scenarios
  - Investigation of the performance of the OLSR protocol in a range of node density scenarios

- Development of new hybrid routing protocol (OLSR-OPP) that integrates routing mechanisms of both WMN and DTN (or opportunistic) routing protocols
  - Prototype implementation of OLSR-OPP in ns-2 network simulator
  - Systematic performance evaluation of OLSR-OPP in networking scenarios ranging from highly dense and connected to highly sparse and disconnected, considering a wide range of intermediate levels
  - Comparison of OLSR-OPP to OLSR and Spray-and-Wait protocol

# Chapter 2 BACKGROUND - WIRELESS MESH NETWORKS AND ROUTING PROTOCOLS

## 2.1 Overview

Wireless Mesh Networks (WMN) presents an emerging communications technology with a great potential for a wide range of applications, where traditional wired or wireless networks cannot be deployed or proof to be inefficient or uneconomical. Example application scenarios are emergency response, disaster recovery, mining, rural communications, etc. Wireless Mesh Networks are essentially a type of wireless multi-hop networks, and are therefore related to other wireless multi-hop networks, such as Wireless Sensor Networks and Mobile Ad-hoc Networks (MANET). According to [1], MANETs can be considered as the simplest variant of WMNs. In other publications, WMNs are considered to be a special type of MANET [26]. In order to address the ambiguity of the term "Wireless Mesh Network", we will provide a definition and identify the most relevant features, to be used in this thesis.

For the context of this thesis, we define Wireless Mesh Networks broadly as wireless multi-hop networks where an end-to-end path is required at the time of communication. This is in contrast to Delay Tolerant Networks (DTNs), where generally no end-to-end routing path exists at any single point in time. WMN scan consist of a combination of infrastructure based mesh routers and mobile mesh clients. Mesh routers are generally infrastructure devices whose main role is to route and forward packets. Mesh clients are mobile end user devices, such as smart phones, which might or might not take part in the routing, as discussed in the following.

Even though WMNs can support a range of wireless communication technologies, they are most often implemented with IEEE 802.11 (WiFi) radios due to their low cost and high availability. In the following, we list a set of key features and characteristics that are typical of Wireless Mesh Networks.

*i.* *Wireless multi-hop*: The most obvious property is the wireless multi-hop nature of WMNs, where packets are sent from source to destination via multiple hops of wireless transmissions.

*ii.* *Mesh Topology, Redundancy*: As the name indicates, WMN have a mesh topology, which means that there is typically a high level of redundancy in the network topology. If a link fails, a new route can typically be established via an alternative path. It is the role and challenge of a WMN routing protocol is to use this redundancy efficiently to implement a self-healing capabilities and provide improved reliability and robustness of the network.

*iii.* *Dynamic Network Topology*: A WMN can consist of a combination of both static and mobile nodes with varying degrees of mobility. Even in a completely static network with no mobility, the network topology of a WMN can be highly dynamic, due to the variability of wireless links. Effects such as interference, fading etc. can result in links being disabled and enabled, and thereby resulting in topology changes. Therefore, routing protocols for WMN need to be able to cope with highly dynamic topologies, and need to be able to find paths between nodes in a constantly changing environment.

The above mentioned characteristics are also shared by MANETs. In our broad definition of WMNs, MANETs are a subset of WMNs as further discussed in the following section. In addition to the above MANET features, WMNs can have the following features and characteristics:

i. Infrastructure Component: Unlike MANET where the network is made up entirely of end-user devices, WMNs can have an infrastructure component, i.e. the mesh routers, forming the backbone infrastructure. In contrast to client devices (mesh clients), they can be equipped with multiple radios, and are generally less resource constrained.

ii. Configuration Flexibility: In contrast to MANET, where all nodes need to provide routing and forwarding functionality, WMN also allow normal IEEE 802.11 clients to connect to the network, without providing this functionality.

As a consequence of these characteristics, WMNs have a set of features which make them attractive for a wide range of application scenarios. These key features are:

i. *Rapid Deployment Capability*. The most time consuming aspect in the deployment of traditional wireless networks is the deployment of the wired backbone infrastructure. By replacing this wired infrastructure with a wireless multi-hop network, the deployment time can be greatly reduced. This feature is especially important for applications scenarios such as emergency response, disaster recovery and counter terrorism.

ii. *Low Cost*. Often, the most expensive aspect of deploying a traditional wireless network is the deployment of the backbone wiring. By making the backbone wireless, the deployment of WMNs can be made a lot less expensive. Another reason why WMNs are generally considered a cost effective solution, is their use of widely available and cheap IEEE 802.11 (WiFi) hardware.

*iii.* *Robustness.* WMN have a fully distributed architecture with no single point of failure, with a network topology that has a significant level or redundancy. WMN routing protocols can use this to implement self-healing capabilities which allows the network to recover in case of link failure. WMNs therefore can achieve a high level of robustness.

## 2.2 WMN Classification

This section presents a classification of WMNs and describes the different types of network setups. In the following, we differentiate between three basic types of network configurations, as also suggested in [1]:

i) *Infrastructure mesh networks* consist of dedicated devices of the network infrastructure, i.e. mesh routers, which provide the wireless backbone infrastructure. Client devices i.e. mesh clients, do not take part in the routing. Instead, they connect to the access points in the mesh network by traditional wireless access technologies.

ii) *Client mesh* networks consist exclusively of mesh clients such as laptops or smartphones. The client devices participate in the mesh routing and packet forwarding, and therefore provide the network services, without any dedicated infrastructure nodes (routers). MANETs fall into this category of WMNs.

iii) *Hybrid mesh networks* consist of both infrastructure devices (mesh routers) and client devices (mesh clients), and both types of nodes contribute to routing and forwarding of packets.

Figure 2.1 depicts a hierarchical and layered network architecture that integrates various configurations of WMN, and shows the most general type of WMN, i.e. a hybrid WMN.. On the top level of Figure 2.1, there is the backbone mesh gateways connected to the Internet via wired links, indicated by solid lines. The gateways provide wireless Internet access (dashed lines) to the second level entities, the so-called *mesh routers*. These wireless routers form the core of the network and provide its backbone. On the lowest level, there are the mobile user devices, i.e. the *mesh clients*. In this example, the mesh clients form part of the network infrastructure by providing packet routing and forwarding services. For example, a mesh client can forward packets of another mesh client who is out of transmission range of a mesh router.

The architecture outlined above needs further discussion. First of all, the mesh gateways are specific mesh routers that have a wired, high-speed connection to the Internet. These wired connections are considered not to be part of the WMN. Thus, the WMN itself is fully wireless. The mesh routers and gateways are typically installed at certain fixed positions. They establish a long term infrastructure. However, the network can easily be extended by adding new routers and gateways, since the backbone links are wireless.

Mesh routers and mesh gateways together establish a wireless multi-hop network that serves as a backbone. Traffic that cannot be delivered directly to the destination node by mesh clients is routed hop-by-hop through the wireless backbone. Furthermore, a WMN routes traffic from a mesh client to a mesh gateway that can forward it to the Internet and vice versa. This way of communication is very different to conventional wireless Local Area Networks

(LANs), which provide only gateway or bridge functionality and where wireless Access Points requires a wired backbone infrastructure to deliver data packets.

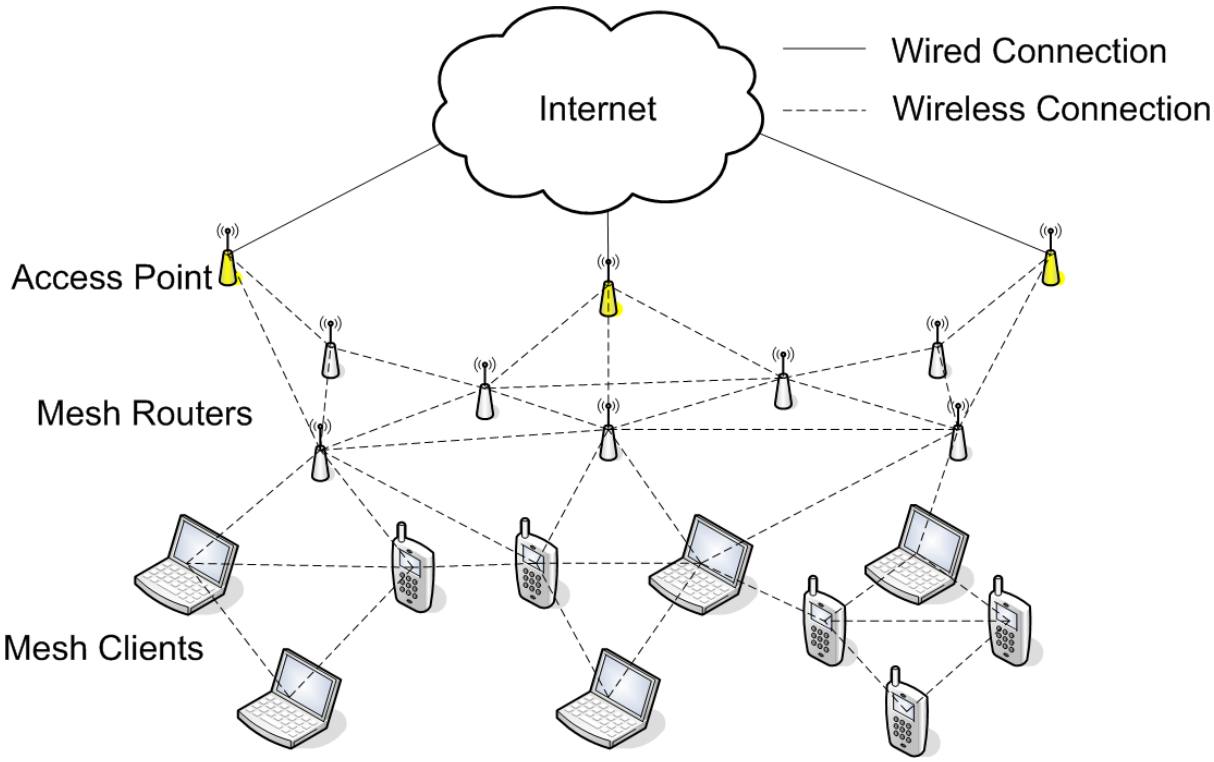



Figure 2.1: Wireless Mesh Networks

In contrast, in a WMN mesh routers have multi-hop routing capabilities and data packets are forwarded along multiple wireless hops to their final destination. As mentioned, routing and forwarding of packets can be provided by both mesh routers as well as mesh clients. Even though mesh routers are typically static, they can be mobile, and mesh clients are quite often mobile. As a consequence, the network topology can be very dynamic.

It is the responsibility of the routing protocol to establish paths between source and destination nodes in the network, which is the focus of this research. Routing in WMN has a set of specific challenges due to the specific characteristics and features of these types of networks. The following section provides and overview of WMN routing, and discusses a number of key WMN routing protocols.

## 2.3 Routing in Wireless Mesh Networks

Routing in general can be referred to as the process of finding the end-to-end path between a source node and a destination node. This has to be done reliably, fast, and with minimal overhead. In general, one goal of routing is to choose a suitably 'efficient' path where efficiency can be measured in terms of throughput, delay, overhead and other metrics. In the context of WMN, we can differentiate between three basic types of routing protocols: proactive, reactive and hybrid. The following discussions will provide more details for all these three types of routing techniques.

### 2.3.1 Proactive Routing

In proactive routing protocols, nodes constantly exchange routing and link information and update their routing tables accordingly in order to be ready when data has to be sent. This type of protocol is used in traditional wired networks e.g. the Internet. The two main approaches of protocols for dissemination of link and routing information, and the computation of routes are of the type "Link State" and "Distance Vector".

In Link State routing each node regularly broadcasts information about its local links (and link costs) to all nodes in the network. Therefore, all nodes have a complete view of the network topology, and can compute the shortest path to any destination in the network. With Distance Vector routing, nodes exchange their complete routing table with their immediate neighbours only. Due to this limited exchange, Distance Vector protocols more slowly adapt to topology changes than Link State protocols. In both Distance Vector and Link State protocols, nodes calculate the shortest path to destination nodes (according to some cost metric, e.g. number of hops), based on the information received.

Proactive protocols attempt to maintain up-to-date state information for all nodes in the network and have been proven to work well for wired networks, but it is recognised that they scale poorly in highly dynamic WMN, e.g. due to node mobility. However, this can be addressed by limiting the scope and frequency of dissemination of such routing information, thus resulting in a more robust and scalable proactive routing protocols such as DSDV [27] or OLSR [3].

### 2.3.2 Reactive Routing

In contrast to the proactive approach, reactive routing gathers the routing information *on-demand,* when it is required. A source node will ask its neighbours for a route when it has data to send, via a Route Request message. If the neighbours do not have any known route, they broadcast the request, and so on. Once the final destination has been reached by these broadcasts, an answer is built and forwarded back to the source. This source can then transmit the data on the newly discovered route. Each device used for forwarding the routing packets has learned the route at the same time.The reactive method works well in a wireless environment in presence of mobile nodes and a continuously changing topology.

The availability of bandwidth in IEEE 802.11 networks is scarce, so the on-demand methods can help conserve it by limiting the amount of routing overhead. The main disadvantage is the increased initial delay, in case when a route does not exist and needs to be discovered before a packet can be sent. Reactive methods are widely accepted for WMNs, and therefore many routing protocols follow this approach. Key examples are DSR [29] and AODV [2], which will be discussed in more detail later in this chapter.

### 2.3.3 Hybrid Routing

Hybrid protocols aim to combine the advantages of reactive and proactive protocols. The goal is to minimise the delay of reactive protocols as well as the routing overhead of proactive protocols. Routes are initially established proactively and the protocol then serves the demand from additionally activated nodes through reactive flooding. The choice for one method or the other requires predetermination for typical cases. The Zone Routing Protocol (ZRP) [62] is such a hybrid reactive/proactive routing protocol. Each mobile node proactively maintains routes within a local region (referred to as the routing zone). Mobile nodes residing outside the zone can be reached with reactive routing.

Another example of a hybrid protocol is the aptly named Hybrid Wireless Mesh Protocol (HWMP) [64]. The protocol proactively maintains routes from nodes to the root or ‘portal’ node, and establishes routes between peer nodes reactively using the AODV protocol. HWMP is discussed in more details in the following section.

## 2.4 WMN Routing Protocols

There are more than one hundred existing routing protocols for wireless multi-hop networks, mostly developed in the context of Mobile Ad-hoc Networks (MANET), Wireless Sensor Networks and Wireless Mesh Networks. While it is impossible to discuss all these protocols in detail, this section gives an overview of relevant examples.

When embarking on the development of a routing protocol, there is considerable scope for making design choices. Routing protocols can operate in many ways because there are various methods used for paths metrics and computation, distribution of routing information, various data structures for storing such information and several strategies for node coordination. The following discussion will illustrate the range of WMN routing protocols based on key examples.

### 2.4.1 Destination-Sequenced Distance Vector Routing (DSDV)

DSDV [27] was one of the first proactive routing protocols available for Wireless Ad-hoc networks. It has not been standardised by any standards authority, but is still a relevant protocol and often used as a reference. DSDV is based on the Bellman-Ford algorithm. With DSDV, each routing table will contain all available destinations, with the associated next hop, the associated metric (numbers of hops), and a sequence number originated by the destination node. Tables are updated in the topology via exchange between neighbouring nodes. Each node will broadcast to its neighbours entries in its table. This exchange of entries can be made by dumping the whole routing table, or by performing an incremental update, i.e. via exchanging just recently updated routes. Nodes which receive this data can then update their tables if they received a better route, or a more recent one. Updates are performed on a regular basis, and are instantly sent in the event of a detected topology change.

If there are frequent topology changes, full table exchanges are more efficient, whereas in a more stable topology, incremental updates will cause less overhead. The route selection is performed based on the metric and sequence number criteria. The sequence number provides a freshness indicator for the routing information, maintained by the destination node. It allows choosing fresher routes over stale ones.

As with every proactive routing protocol, DSDV reduces the latency by continually maintaining a route to all destinations at all times. However, DSDV has a few limitations, mainly in the route table update process. One of the major problems is that data is exchanged only between neighbours, and a topology change can take a significant amount of time to propagate, resulting in limited convergence. This problem and limitation is more significant for highly dynamic networks.

### 2.4.2 Optimized Link State Routing (OLSR)

OLSR is another proactive protocol, originated at INRIA (Institut National de Recherche en Informatique et Automatique), France. It has been proposed for standardisation to the Internet Engineering Task Force (IETF) with the RFC 3626 [3] document in October 2003. As the name implies, OLSR is a link state protocol, where nodes broadcast local link information in the entire network. In OLSR, shortest routes are computed based on Dijkstra's algorithm. OLSR is the most widely used proactive routing protocol for WMNs. It addresses the high overhead problem common to proactive link state routing protocols with the introduction of multipoint relays (MPR).

Multipoint relays reduce the overhead of broadcasting link state messages in the network by adding a layer of hierarchy. Multipoint relays aggregate link state updates on behalf of other nodes, and distributes them in the network. OLSR defines two types of messages. It uses "HELLO" messages in order to inform its immediate neighbours about its current links states. These "HELLO" messages contain a timeout, a hold time, and information about link status. In contrast to DSDV, it is not the entire routing table that is exchanged. OLSR will use this to maintain its link state information. "HELLO" packets are broadcast on a regular basis.

OLSR also uses "TOPOLOGY CONTROL" (TC) messages. This type of message is event triggered. Each node which detects a change in its direct neighbourhood will send a TC message containing its network address and a list of its MPRs. This packet is used to inform other nodes of topology changes. This will start a new route calculation process.Only Multi Point Relay (MPR) nodes send TC packets to their selector nodes. TC messages contain a list of one-hop neighbours which have selected this node as their MPR. TC messages are used for routing table calculation and maintaining the network topology. For node to be selected as a

MPR is highly dependent on its "WILLINGNESS" value. The "WILLINGNESS" parameter in OLSR is defined as how willing or able a node is to forward traffic. The parameter is specified as one of eight levels (0-7), with a default value of 3.A "WILLINGNESS" value of 0 means a node will never be selected as a MPR, while value of 7 means it is always ready to be selected as MPR.

OLSR increases the network performance compared to DSDV, due to the multipoint relay mechanism. This mechanism reduces the amount of data exchanged by avoiding duplicate data transmissions. MPRs also propagate changes more quickly in the network, thereby reducing the route fluctuation impact in a mobile environment. Compared to DSDV, OLSR converges more quickly to changed topologies and uses less control traffic. However, on large topologies, OLSR is still vulnerable to quick network changes, and incurs a relatively large overhead.

OLSRv2 [59] is currently being developed within the IETF. It maintains most of the key mechanisms of OLSR such as the MPR selection mechanism and link update dissemination. OLSRv2 provides increased flexibility and a more modular design.

### 2.4.3 Ad-Hoc on Demand Distance Vector (AODV)

AODV was defined as an IETF standard in July 2003 (RFC 3561) [2], as an improved version of DSDV. AODV is a reactive protocol and establishes routes on-demand. The AODV protocol is inspired from the Bellman-Ford algorithm like DSDV. The principal change is that AODV discovers routes on-demand, in contrast to the proactive discovery in DSDV. A node is silent while it does not have any data to send. Then, if the upper layers are requesting a route for a packet, a "ROUTE REQUEST" packet will be broadcast to the immediate neighbourhood of the node. If a neighbour has a route corresponding to the request, a "ROUTE REPLY" message will be returned. This message is like a "use me" answer. Otherwise, each neighbour will forward the "ROUTE REQUEST" to their neighbours via broadcast communication, and increment the hop value in the packet data. They also use information in the "ROUTE REQUEST" message for building a reverse route entry to the originator of the message. This process continues until the destination node has been found, or alternatively a node with a route to the destination has been found.

A route that has been created during the AODV route discovery process will be kept active in the routing table for limited period of time. The parameter that decides how long a node should keep a route in the routing table after the last successful transmission of data packets is called "ACTIVE_ROUTE_TIMEOUT" or ART. When a route is not used for a period of ART seconds, the route will be marked as invalid in the routing table.

Another important aspect of the AODV protocol is the route maintenance. When a link to neighbour is no longer available and it was used on a route to a destination node, this route is not valid anymore. AODV uses "HELLO" packets on a regular basis to check if neighbours are still alive and if the corresponding links are still active. If there is no response to the "HELLO" packet sent to a node, then the originator deletes all associated routes in its routing table. Links can be detected as broken, either with a lack of received "HELLO" messages or alternatively via link layer feedback. Link layer feedback uses information wireless network interface about the success or failure of unicast transmission attempts, indicated via the receipt or lack of a layer 2 acknowledgement. Link layer feedback is an optional feature and not always implemented.

Depending on the location of the link break, i.e. its distance from the source node relative to the distance to the destination node, AODV will either attempt to repair the route locally (local repair), by issuing a "ROUTE REQUEST" message at the node where the link break was detected. Alternatively, this node can send a "ROUTE ERROR" message upstream to the source node, which can then initiate a new route discovery process from scratch. As will be discussed later in this thesis, the choice of which route repair mechanism to apply can have a significant impact on the network performance.

### 2.4.4 Dynamic MANET on Demand (DYMO)

DYMO [28] or AODVv2 as it has been referred to more recently, is a new reactive (on demand) routing protocol, which is currently developed in the context of the IETF's MANET working group. DYMO is work in progress, and the discussion here is based on the information from [28].

DYMO builds upon experience with previous approaches to reactive routing, especially with the routing protocol AODV. It aims at a somewhat simpler design, helping to

reduce the system requirements of participating nodes, and simplifying the protocol implementation. DYMO retains proven mechanisms of previously explored routing protocols like the use of sequence numbers to enforce loop freedom. At the same time, DYMO provides enhanced features, such as covering possible MANET–Internet gateway scenarios and optional implementation of a feature called path accumulation, which is essentially based on the source route accumulation feature available in DSR protocol.

Besides route information about a requested target, a node will also receive information about all intermediate nodes of a newly discovered path. This is major difference between DYMO and AODV, the latter of which only generates routing table entries for the destination node and the next hop, while DYMO stores routes for each intermediate hop. To efficiently deal with highly dynamic scenarios, links on known routes may be actively monitored, e.g. by using the MANET Neighbourhood Discovery Protocol [59] or by examining feedback obtained from the data link layer. Detected link failures are made known to affected nodes by sending a route error message (RERR) to all nodes in range, informing them of all routes that have become unavailable. Should this RERR in turn invalidate any routes known to these nodes, they will again inform all their neighbours by multicasting a RERR containing the routes concerned, thus effectively flooding information about a link breakage through the MANET. DYMO also does not implement a local link repair mechanism, which is in contrast to AODV.

### 2.4.5 Dynamic Source Routing (DSR)

As a reactive protocol, DSR [29] has some similarity with AODV. The key difference to AODV is that DSR uses source routing, which means that each time a data packet is sent, it contains the list of nodes via which it will be forwarded. In other terms, each packet contains the route it will use. This mechanism allows nodes on the route to cache new routes, and also, allows the originator to specify the route it wants, depending on criteria such as load balancing and QoS. This mechanism also avoids routing loops.

If a node has to send a packet to another one, and it has no route for that, it initiates a route discovery process. This process is very similar to the AODV protocol as a route request is broadcast to the initiator's neighbourhood until the destination node is found or a node with a route to the destination node is found. Thus, the difference is that every node used for

broadcasting this route request packet deduces the route to the originator, and keeps it in cache. Also, there can be many route replies for a single request. Another difference with AODV is in the route maintenance process. DSR does not use broadcasts such as AODV's "HELLO" packets. Instead, it uses layer two built-in acknowledgments. If DSR detect a route break in its routing table, it will use RERR messages to notify its neighbours.

### 2.4.6 Hybrid Wireless Mesh Protocol (HWMP)

HWMP is the default routing protocol for WLAN mesh networking, as defined in the upcoming IEEE 802.11s wireless mesh standard [64]. 802.11s is an extension of the IEEE 802.11 MAC standard and defines an architecture and protocol to support broadcast, multicast and unicast communication over self-configuring wireless multi-hop networks, using "radio-aware" routing metrics.

HWMP is the default routing protocol of IEEE 802.11s, and is required to be supported by any standards compliant implementation. The hybrid nature of HWMP consists of a combination of proactive and reactive routing. The proactive part is used to maintain routes of all nodes to special node called 'mesh portal', which typically provides gateway connectivity to an external network. This is achieved via the portal node periodically broadcasting announcement messages, which sets up a tree topology with the portal at the root.

The reactive part of HWMP is used to find optimal routes between peer nodes in the mesh network. It is largely based on AODV, as described earlier. It uses the distance vector routing and AODV's well-known on-demand route discovery process with route request and route reply messages. Destination sequence numbers are used to recognize stale routing information. However, there are some significant differences in the details. In contrast to most MANET protocols, which are implemented at layer 3 of the protocol stack, HWMP implements its routing functionality at layer 2, and consequently uses MAC addresses instead of IP addresses. Furthermore, HWMP can make use of more sophisticated routing metrics than hop-count, i.e. "radio-aware" metrics. A new path metric field is included in the RREQ/RREP messages that contain the cumulative value of the link metrics of the path so far. The default routing metric of HWMP is the airtime metric.

### 2.4.7 Routing Aware - Optimized Link State Routing (RA-OLSR)

The RA-OLSR [64] protocol is an optional, proactive routing protocol of the emerging IEEE 802.11s standard. It follows closely the specification of the OLSR protocol as described earlier in this chapter. Similar to HWMP, it uses MAC addresses and can work with arbitrary routing metrics such as the airtime metric. Furthermore, it defines a mechanism for the distribution of addresses of non-mesh WLAN clients in the RA-OLSR mesh.

The link state is the value of the link metric and is used in the shortest path computation. Therefore, a link metric field is associated to each reported neighbour in OLSR HELLO messages and TC messages. The value of the link metric is also stored in the corresponding information repositories, the link set and the topology set. The link metric is also used in the heuristic for the selection of the multipoint relays.

Each mesh access point maintains a local association base (LAB) that contains all legacy IEEE 802.11 stations associated with this mesh AP. It broadcasts local association base advertisement (LABA) messages periodically, in order to distribute the association information in the mesh network. The information received from LABA messages is stored in the global association base (GAB) in each node. The information of both LAB and GAB is used in the construction of the routing table and provides routes to legacy stations associated with mesh access points. To save bandwidth, it is possible to advertise only the checksum of the blocks of the LAB. If there is a mismatch between a received checksum and the checksum in the GAB, the node requests an update of the corresponding block of the LAB of the originating node.

## 2.5 *Effects of Mobility and Other Key Factors on WMN Routing Protocol Performance*

Mobility is an important factor in determining the overall performance of wireless mesh network routing protocols. Performance comparisons are typically being made between proactive and reactive routing protocols in the context of different degrees of network mobility. As discussed before, a key difference between proactive and reactive protocols is the approach in which route maintenance is handled. Proactive protocols, as the name

suggests, will actively maintain routes even when no data transmissions are taking place. In contrast, reactive protocols only maintain a route while it used for active data transmission across the network and the route will time-out when transmission stops. This section provides a brief discussion of the respective impact of mobility and other key factors on proactive and reactive wireless mesh routing protocols.

Simulation results in [111] show that reactive protocols (AODV and DYMO) generally outperform proactive protocol (OLSR) in terms of Packet Delivery Ratio (PDR) in scenarios with high mobility rates. In these simulations, a small packet size of 64 bytes was used, with a sending rate of 4 packets per second. The simulations were done using a Random Waypoint (RWP) Mobility Model [58]. Due to their reactive nature, reactive protocols are generally better able to adapt to rapidly changing network environments. In particular, they are able to detect link breaks more quickly, and are able to update routing table accordingly faster.

The problem with proactive protocols, and OLSR in particular, is that they struggle to converge in networks with highly dynamic topologies, caused by node mobility. In addition to the higher signalling overhead of proactive protocols, packet loss is to a large extent due to the slow detection of topology changes, and consequently stale routing table entries, resulting in packets being forwarded over broken links. This is reflected in the results in [111], and is consistent with results in [36], [115], even though in [36] a different proactive routing protocol was used (DSDV).

In terms of routing overhead, proactive protocol such as DSDV and OLSR have a constant overhead, regardless of the level of mobility and traffic load [36], [116]. They require periodic broadcasts of control messages to maintain all available routes. In contrast, in reactive protocols such as AODV and DSR, the routing overhead depends on the level of node mobility and the number of active traffic flows. Frequent link breaks will cause an increased number of routing control packets sent across the network to enable route repair. As for traffic load, more active traffic flows and traffic sources also result in a higher control packet overhead, due to the larger number of routes that need to be maintained.

In terms of end to end delay, proactive protocols generally produce lower delay than reactive protocols, as shown in [36]**Error! Reference source not found.**. Proactive protocols such as OLSR and DSDV continuously maintain routes between all node pairs in a network. In the case when a new route is required, no extra route discovery delay is incurred, which is in contrast to reactive protocols. The well-known cost and drawback of this is the increased signalling overhead in terms of network bandwidth. In situations of high traffic load, this additional signalling overhead of proactive protocols can cause increased congestion in the network, resulting in a greater number of packets being lost, and hence in a lower Packet Delivery Ratio (PDR).

# Chapter 3 BACKGROUND - DELAY TOLERANT NETWORKS AND ROUTING PROTOCOLS

## 3.1 Overview

Delay Tolerant Networks (DTNs) represent another type of wireless multi-hop networks, with a number of key differences to Wireless Mesh Networks. DTNs, as the name suggest, are designed to survive periods of network and connectivity disruptions. DTNs make no assumption of end-to-end connectivity at a single point in time between source and destination nodes, in contrast to WMNs [67]. Instead of relying on end-to-end routes, DTNs rely on the concept of store-carry-forward, where data is exchanged opportunistically during temporary encounters of mobile nodes.

Other key characteristic of DTNs are they may have low asymmetric bandwidth, low transmission range, widely scattered nodes [68], [69] and potentially also significant power constraints [70]. Researchers have proposed a number of DTN applications. For example, ZebraNet [65] has been proposed to monitor the long term behaviour of wild animals (such as Zebras as the name suggested) that are sparsely distributed over a large geographical area. Another example of DTN application is communication among the villages of Saami reindeer's herders living in remote areas in northern Sweden [66].

This chapter aims to give an overview of the basic characteristics and key aspects of DTNs, with a particular focus on DTN routing protocols.

## 3.2 DTN Key Characteristics

We consider a DTN example from [71], as illustrated in Figure 3.1. The example shows vehicles such as buses, a number of cars and trucks, all equipped with radio transceivers, which allow them to communicate with each other, if within range. There are also wireless access points which have access to Internet.

Communication links are established opportunistically and intermittently, when two nodes come into transmission range of each other. For extended periods of time, individual nodes or vehicles might have no connectivity at all. Due to this, vehicles might have to store data and carry it with them for some distance, until they encounter another vehicle or maybe an access points to forward the data to.

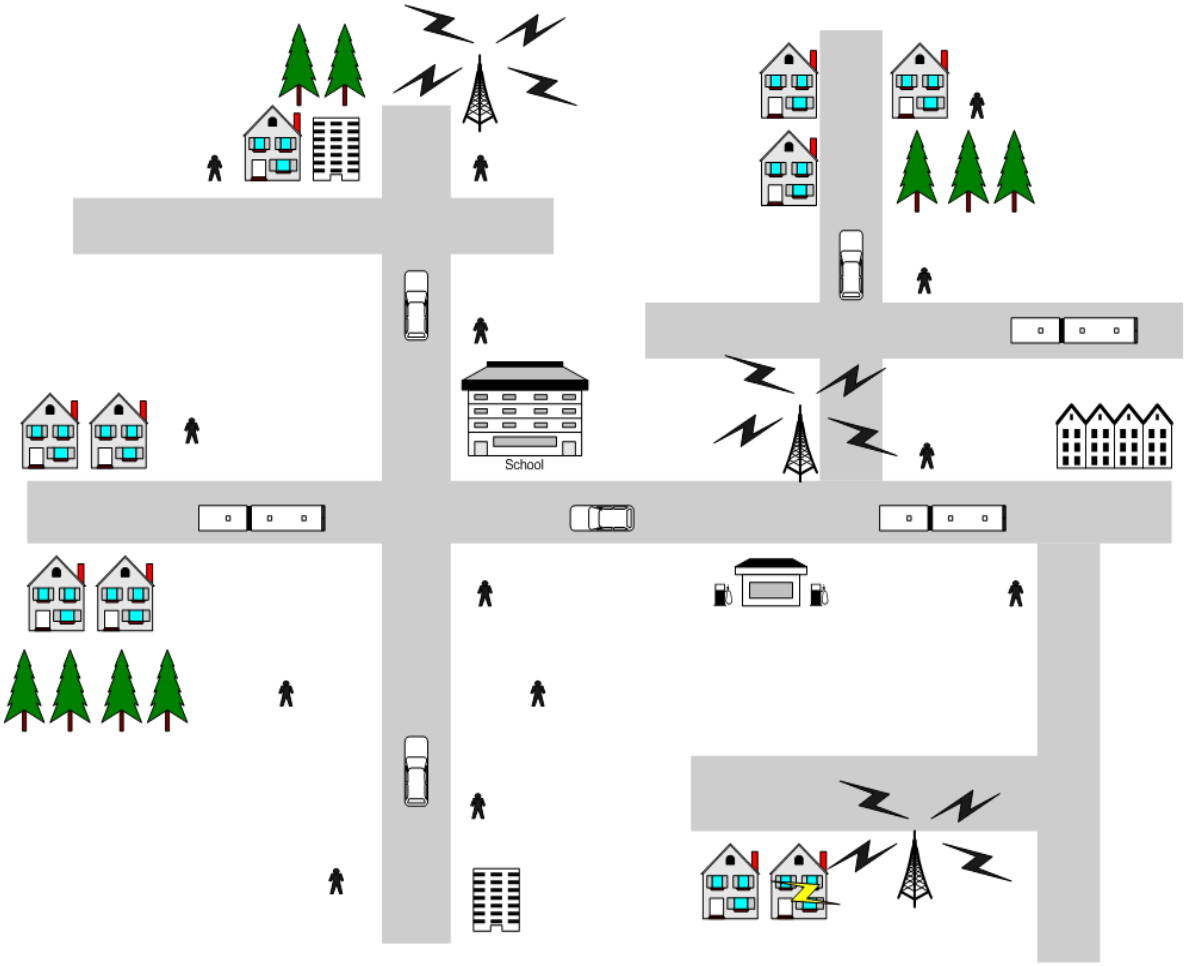



Figure 3.1: Example of Vehicle and People based DTN

DTNs can also be formed by a collection of smart phones and other mobile devices equipped with radio transceivers. The availability of a range of sensors on typical smartphones enables participatory sensing applications, measuring traffic conditions, air pollution, etc. Using free direct peer-to-peer communication between devices using WiFi or Bluetooth, an opportunistic network or a DTN can be established and used for the dissemination of the sensed information. But smartphones have some limitations in this

context, such as limited battery life. User mobility can be highly unpredictable, resulting in unpredictable encounters and network connectivity. This has the consequence of potentially long delays of connectivity interruptions. Under these challenging conditions, it is difficult to ensure that data is delivered efficiently and reliably to the intended destination. DTN routing protocols need to be able to cope with these challenges. The following section gives a brief summary of DTN characteristics. Following that, we will give an overview of key DTN routing protocols.

## 3.3 DTN Key Characteristics

Delay Tolerant Networks (DTNs) are different from classical MANETs or WMNs in the way in which packets are forwarded and delivered to their destination. In WMNs and MANETs, the routing protocol tries to establish an end-to-end route between source and destination nodes which needs to be maintained while the communication session lasts. In contrast, DTNs do not assume that such an end-to-end route exists at any given point in time. The reason for a lack of end-to-end routes is typically the increased sparseness of the network. In DTNs data packets, or "bundles", are delivered in a store-carry-forward approach, using pair wise communication between nodes during opportunistic and typically intermittent encounters. In the following, we list a set of key features and characteristics that are typical of Delay Tolerant Networks, which is very important when designing routing protocols.

*Dynamic/Unpredictable Topologies*:

Similar to MANETs, in DTNs the location of nodes and hence the network topology can vary from time to time because of node mobility, and nodes can have different mobility patterns. For example in Figure 3.1, we show a network between vehicles and groups of people. Other examples include monitoring wild animals' habitat movement [72] and communication on trains [73]. Because of the dynamic nature of these networks, it is difficult to predict the network topology.

*High variation in connection duration*:

In DTNs, when nodes encounter each other, and since the duration of these encounters are unpredictable and can be short, it is important for routing protocols to decide whether to forward data packets or not, and which packets to forward, in order to

maximize the probability of successful delivery to the destination node. This decision can depend on a number of factors, such as buffer capacity, encounter history, etc. For example, in ZebraNet [65] this decision is crucial to maximize the delivery probability as nodes may encounter each other during very limited time periods only.

*Lack of topology and path information*:

As DTNs typically do not have complete routing information due to a lack of available topology information. Because of this, it is impossible to calculate the best route globally, such as is the case in link state routing protocols. DTNs rely on local information and metrics, obtained from pair wise node encounters, in order to decide which packets to forward and to which nodes.

*Resources limitation:*

Routing protocols also have to take into consideration the typically limited resources of DTN nodes such as buffer capacity, CPU, memory and also power (battery lifetime). For example, smart phones typically have very limited battery power and buffer capacity, so it is very important to carefully manage their energy, and consequently their communication pattern. In water pollution or wildlife habitat movement monitoring, nodes can be deployed for months or years before the data is being collected and batteries are replaced. In addition, DTN routing protocols can distribute or leverage resources or such as data or power across multiple nodes. For example, a node may want to forward all a fraction of stored data to other nodes, in order to free up buffer memory.

### 3.3.1 DTN Bundle Layer

The main feature in DTN routing protocols is to implement the *store-carry-forward* mechanism, which means that nodes will store the data locally, and when they next encounter other nodes, they will forward it based on the forwarding rules of the routing protocol. These stored data blocks are called ‘Bundles’, and detailed “bundle” protocol details are described in an IETF RFC “Bundle Protocol Specification” [74]. Most of the DTN protocol designs are based on this approach.

As per [74], the underlying of Bundle layer is shown in Figure 3.2 and Figure 3.3 [75]. In these figures, the implementation of store and forward message switching in DTNs is implemented by overlaying a new protocol layer called the "Convergence Layer" on top of heterogeneous region specific lower layers [76]. From Figure 3.2 we can see that the bundle layer ties together the 'region specific' lower layers.

This enables applications to communicate across the multiple regions. Regions can be loosely defined in this context of areas of the network with a common set of protocols and network characteristics. Bundles are also called "messages' as in message switching. The bundle layer stores the messages (or bundles) and forwards them (or possibly fragments of bundles) when they encounter other nodes. Bundles can be arbitrarily long and can be an aggregate of lower layer network protocol packets. Bundles can be broken into fragments during transmission. Applications sitting on top of the bundle layer need to use the specific primitives DTN communication provided by the layer.

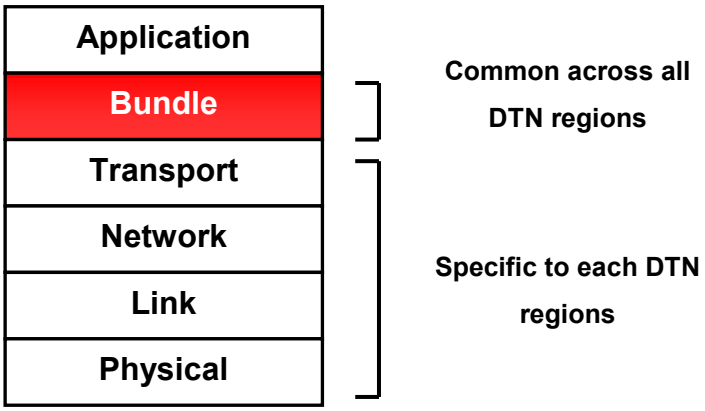



Figure 3.2: DTN Network Stack with Convergence Layer [75]

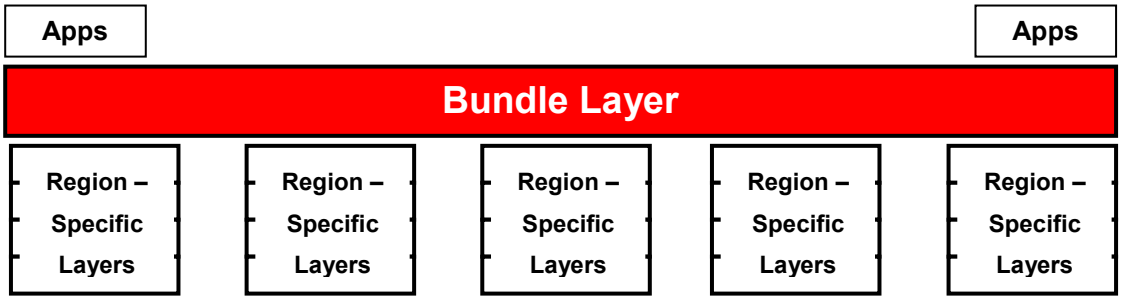



Figure 3.3: Bundle Protocol Layer [75]

### 3.3.2 DTN Node Roles

This section presents a simple classification that considers the functional differences of Delay Tolerant Network nodes. The main idea here is to differentiate between DTN nodes and to bring into focus the aspect of DTN service provisioning and management. In DTNs, a node is an entity with a "Bundle Layer". A node can be a host, router or gateway, as described below, or a combination [75].

*Host:*

The function of a host is to send and received bundles, but it does not forward them. A host can be a source or destination in a DTN bundle transfer.

*Router:*

The role of a router in a DTN is to forward bundles within a single DTN region. A DTN router may optionally also become a host, depending on the protocol's management mechanism.

*Gateway:*

A DTN Gateway is similar to a DTN router, with the additional capability to forward bundles between two or more DTN regions. Like a router, a gateway may also optionally become a host in a DTN.

## 3.4 Routing in Delay Tolerant Networks

As mentioned before, routing in Delay Tolerant Networks is fundamentally different from other multi-hop wireless networks such as MANETs and WMNs. In DTNs, the network is characterized by intermittent connectivity, long delays and lack of an instantaneous end-to end path. In such scenarios, traditional MANET/WMN routing protocols such as AODV or OLSR are not practical and perform poorly. Since they fail to establish end-to-end routes, they will eventually, after a relatively short period of time, just start to drop packets.

This is the fundamental difference between WMNs and DTNs. Since DTN routing protocols implement the store-carry-forward approach, where data is stored in the nodes for extended periods of time, in the hope that, due to node mobility, either the destination node is directly encountered at some time, or the date can be forwarded to anther DTN router with hopefully a good probability of encountering the destination node.

DTN routing protocols are designed to be as efficient as possible in cases of highly sparse networks and intermittent connectivity. As implied by the name, they must be able to tolerate long delays and cannot make any connectivity assumptions. All these things combined make the design of efficient DTN routing protocol a challenging task.

In the store-carry-forward (SCF) routing approach in DTNs [77], nodes will have to buffer the message until they get the opportunity to forward them during any contacts or encounters with other nodes. The real challenge is to decide when and how to forward each bundle or message, and to which node to forward to when the opportunity arises. Making good forwarding decision will increase the probability of message delivery, and will reduce the required time, while bad forwarding decision can result in failure to deliver the bundles.

DTN routing protocols also must take into account the limited resources of nodes, such as the buffering capacity of bundle messages, as well as the high unpredictability of DTNs. DTN protocols therefore need to integrate an efficient buffer management message distribution mechanism. Good routing protocols will wisely distribute their buffer resources across multiple nodes; e.g. they move some of the stored bundle message to other nodes in order to free up memory, while maintaining the number of message copies in the network.

Generally, there are two basic types of DTN routing protocols as described in [78], i.e. *replication based* and *knowledge based* protocols. The following discussions will provide a brief overview of these two types of DTN protocols.

### 3.4.1 Replication Based Protocols (Flooding)

As the name suggests, replication based protocols work by making several replicas of the original message. Each node will maintain a number of copies of each message and will retransmit them when the opportunity arises during encounters with other nodes. The protocol will flood the network with the bundle messages and has a very high delivery probability of successful message delivery towards the intended destination.. These types of protocols do not require global or local knowledge about the network.

However, this approach has one main drawback, it is a relatively resources hungry scheme, mainly due to the high level of redundancy and message duplication. This can result in network congestion and can severely degrade overall network performance. It is a challenging task to limit this network congestion, while maintaining a high probability of message delivery. Reliability and resource consumption can typically be traded off by adjusting the level of message replication, e.g. the number of message copies that are forwarded in the network.

### 3.4.2 Knowledge (Forwarding) Based Protocols

Knowledge or Forwarding based approaches generally use fewer network resources than the replication based protocols. They require some form of network topology information and global/local knowledge of the network to achieve a more targeted message exchange, as opposed to the more random approach of flooding based protocols.

Using global and local network information, knowledge based protocols can find the best (according to some metric) path towards the intended destination. Therefore, knowledge based protocols can be much more efficient in terms of resource usage, and can be more scalable. However, one drawback of knowledge based protocols is that they generally do not achieve the high delivery probability in a lot of DTN environments that replication based protocols can [76].

## 3.5 DTN Routing Protocols

There are a large number of existing routing protocols that are designed for Delay Tolerant Networks. There is significant research underway to further develop them for better resources efficiency and performance. This section will provide a brief overview of some of the key DTN routing protocols.

### 3.5.1 Direct Contact

Direct Contact is one of the simplest techniques in DTN routing. It is the degenerate case of a flooding based approach, where the source node simply stores the bundle message and waits until it comes into contact with the destination node, and then 'directly' delivers the data via a one-hop transmission. However, since no knowledge of network information is needed for any data transmission, this approach is generally considered as flooding based approach. It is a simple protocol and consumes minimal resources.

However, this scheme only works if source and destination nodes come into direct contact. Obviously, this might take a very long time, or might never happen at all. As a result, the protocol has very limited performance in terms message delivery delay and message delivery rate in most DTN environments.

### 3.5.2 Epidemic Routing

Epidemic routing [79] is a native flooding based protocol in nature. Nodes will continuously replicate and transmit a message across the network whenever nodes encounter other nodes that do not already possess a copy of the message. In epidemic routing all nodes can be carriers of a message. The basic protocol assumes that each node has unlimited storage space and bandwidth and is able to store all the messages transmitted during an encounter with other nodes. Nodes only receive a copy of a message if they do not already have a copy.

Epidemic routing maintains a "summary vector" which is a list of message in the database. The summary vectors are exchanged between nodes during encounters, and will determine which messages are not redundant and are candidates for transmission. Eventually, all nodes will receive the message, provided there are a sufficient number of message exchanges and an adequate level and type of mobility. Epidemic routing is robust to node or

network failure since messages keep on propagating throughout the network, providing a high degree of redundancy. Messages can be distributed quickly in connected portions of the network. In other words, epidemic routing will simply replicate messages to all encountered nodes, provided they have never seen the messages before. Epidemic routing makes no assumptions about network topology, mobility patterns, or encounter probabilities.

Generally, epidemic routing can achieve a very high message delivery ratio. However, epidemic routing is a relatively resource intensive protocol. It requires large amounts of buffer space, bandwidth and power. Messages will continue to be propagated and stored in the nodes' buffers, even after they have been delivered to the destination, resulting in high level of resource consumption.

### 3.5.3 Spray and Wait

Spray and Wait [80] is one of the most widely used and cited DTN routing protocols. The protocol aims to lower the overhead by only distributing a limited number of message copies. Spray and Wait routing process consist of two different phases, as the name suggests, a 'Spray' phase and a 'Wait' phase.

When a node wants to send a message, a parameter 'k' is attached to that message, indicating the maximum number of message copies allowed in the network. Nodes deliver or "spray" a message copy to k different other nodes, or "relays". After this is completed, a node enters the wait phase, where it simply waits until it encounters the destination node for that message, and can successfully deliver it.

Compared to epidemic routing, the spray and wait protocol limits the overhead by limiting the level of redundancy of message propagation in the network. The level of message distribution and redundancy can be controlled via the parameter k, by trading off routing overhead versus performance, i.e. delivery probability.

One drawback of the Spray and Wait protocol is that relay nodes can only deliver a message in the wait phase when they directly encounter the destination node, and hence it relies on a high degree of node mobility.

### 3.5.4 Spray and Focus

Spray and Focus [81] is a modified version of Spray and Wait. The main limitation of the Spray and Wait protocol is that it only allows a maximum of two hops to deliver a bundle message.

Similar to Spray and Wait, Spray and Focus has two phases, the 'Spray' phase and the 'Focus' phase. The 'Spray' phase is similar as in Spray and Wait routing. In the Focus phase, rather than only delivering messages to destination nodes directly via direct one-hop transmission, relay nodes can also forward a single message copy to another relay node, according to a specific utility based criterion [81], which is based on a set of timers which record the time since the two nodes last encountered each other.

Spray and Focus was designed to increase the delivery probability of the Spray and Wait routing scheme, while minimally increasing the overhead.

### 3.5.5 MaxProp Routing

MaxProp Routing [82] is a flooding based protocol designed for vehicle based DTNs. In MaxProp, in the event of a carrier node encountering another node, all messages not yet sent to that contact node will be replicated and transferred. The key contribution of MaxProp is in determining which messages are to be transmitted first, and which one are to be dropped. For this, the protocol maintains an ordered queue, with the order based on the estimated probability of a future path to the given destination.

Each node maintains a vector with these path likelihoods, and the corresponding values are updated based on node encounters and successful path establishments. MaxProp uses Djikstra's algorithm to calculate the entire path from node to node using the path likelihoods.

MaxProp also uses acknowledgements for every successfully delivered message. These acknowledgements will help flushing out any redundant message from the network, so buffer space in DTN nodes can be freed and more optimally used. With these approaches, MaxProp can significantly increase the message delivery rate and at the same time reduce the network latency and overhead.

### 3.5.6 Probabilistic Routing Protocol Using History of Encounters and Transitivity (PROPHET)

PROPHET routing [83] is a probability based routing protocol that uses past node encounters to determine the likelihood of a node to meet destination nodes in the future. In PROPHET, during node encounters, a carrier of a message will evaluate the probability of the encountered nodes (potential carriers) to directly meet the final destination of the message. If the probability of a potential carrier is higher than a certain threshold (set by the protocol) or higher than the probability of the current carrier itself, the message will be delivered to that new carrier. The current node that transferred the message does not need to delete the message after sending it, provided there it still has enough buffer space available. The reason being is that the forwarding node may still encounter a better node in the future, i.e. one with a higher delivery probability, or even the destination node itself. This will improve the delivery ratio of the message.

Routing in PROPHET was designed for better resources utilization compared to Epidemic routing, which is quite a resource hungry protocol. The PROPHET protocol is specified in an IETF draft [84] and is maintained by the Delay Tolerant Networking Research Group (DTNRG) [85]. It has been evaluated in real world situations such as *Sami Network Connectivity (SNC)* [86], which is the network of the Saami nomadic tribe, which is the rural population in the remote Swedish Lapland and currently being monitored and developed by EU research grouped called Networking for Communications challenged Communities (N4C) [87].

### 3.5.7 Resource Allocation Protocol for Intentional DTN (RAPID)

RAPID (Resource Allocation Protocol for Intentional DTN routing) is a DTN routing protocol proposed in [88]. It is a replication based flooding protocol aiming at improving performance of DTNs based by optimising a particular performance metric such as worst-case delivery delay or the packet delivery ratio within a given deadline.

RAPID considers DTN routing as a resource allocation problem, based on a utility function, which determines how packets are replicated and forwarded in the networks.

The protocol considers network resources such as buffer storage and bandwidth before transmitting any bundle message. This is a crucial and critical decision, especially if the network is very resource limited.

### 3.5.8 Bubble Rap Routing (BBR)

Bubble Rap Routing (BRR) [91] is a social-based forwarding protocol for DTNs, particularly aimed at 'Pocket Switched Networks' (PSNs), consisting of mobile devices carried by people, such as smart phones, and making use of human mobility. BRR tries to exploit the nature of human mobility to improve the network performance. To achieve this, is introduces to social metrics which form the basis for making forwarding decisions.

The protocol has two basic ideas and intuitions. First, people have varying roles and popularities in society, and these can be assumed to be true also in a 'Pocket Switched Networks'. Secondly, people form communities in their social lives, and this also expected to be observed in the network. These social concepts are reflected in the BBR protocol via two social metrics, Community and Centrality, which form the basis for forwarding decisions. In BRR, messages 'bubble' up and down the 'social hierarchy', based on the community and centrality metrics, which are calculated based on local and global information.

The implementation of Bubble Rap routing is based from the study of real human movement traces. The authors of [91] have shown that based on real human mobility traces, BRR routing has better forwarding efficiency than PROPHET routing.

Each node in BRR belongs to at least one community, even a single node can be considered as a community. One drawback of BRR is that it is not easily applicable in scenarios of fast node mobility. For example, in vehicular networks, community is not a clearly definable metric and may not be suitable for more random networks.

## *3.6 Summary*

DTN (or opportunistic networks) provide a special set of characteristics and challenges for designing routing protocols, compared to other wireless multi-hop networks such as WMNs and MANETs. The key difference is that the network is assumed to be

sparser, with highly intermittent connectivity and generally a lack of end-to-end routes available at a single point in time. Consequently, routing and forwarding needs to occur in a very different fashion, using a store-carry-forward approach, where message deliver relies largely on node mobility, resulting and large end-to-end delivery delays.

This chapter has given a brief overview over a few of the key approaches and protocols for DTN routing. All protocols follow the basic store-carry-forward approach, but differ in the details how messages are forwarded, and what information is considered to make these forwarding and routing decisions. What is common to all of these protocols is that they are specifically tailored to the DTN scenarios, and would perform poorly in a WMN or MANET scenario, where end-to-end routes are available.

The overarching goal of this thesis is to explore wireless multi-hop protocols that can operate across a very wide range of network and deployment scenarios, and can adapt its operation based on the specific scenario, and can operate accordingly. Towards this aim, Chapter 6 will explore a new protocol that can combine the features of basic DTN routing with traditional WMN routing, and provide efficient routing across a wide range of network connectivity scenarios, ranging from completely connected WMNs, to highly disconnected DTNs at the other end of the spectrum.

The following chapter provides an overview of key related works in regards to adaptive wireless multi-hop routing protocols, including work on integration of DTN and WMN routing towards the end of the chapter.

# Chapter 4 ROUTING PROTOCOL ADAPTATION - LITERATURE REVIEW

## 4.1 Overview

Currently, no single routing protocol can provide optimal performance in the wide range of often unpredictable and dynamic deployment scenarios of wireless multi-hop networks. The overall aim of this thesis is to explore the potential for protocol tailoring and adaptation to such different environments. This chapter provides an overview of relevant research in this area. The first part of this chapter considers WMN protocols that aim to adapt critical protocol parameters to different network scenarios.

Section 4.2 gives an overview of WMN routing protocol parameters and their impact on the network performance, and their potential for adaptation. Section 4.3 provides a more detailed overview over of specific adaptive WMN routing protocols where adaptation is done via protocol parameter tuning.

The second part of the chapter looks at protocols that adapt protocol operation or strategies to different scenarios. Section 4.4 discusses WMN protocols which use different basic routing strategies (e.g. proactive vs. reactive) in different parts of the network, or for different network scenarios. Finally, Section 4.5 provides a survey of proposals for hybrid WMN/DTN protocols, which can operate in both WMN and DTN environments.

## 4.2 WMN Protocol Parameters

An optimal choice of routing protocol parameters can have a significant impact on the overall network performance. Some protocol parameters are common across a number of WMN protocols. An example is the HELLO interval parameter, defines the frequency in which HELLO packets are exchange between neighbouring nodes for establish connectivity and potentially link quality between nodes, as discussed in Chapter 2. The HELLO interval parameter is used in the AODV, DYMO and OLSR protocols. Some protocol parameters are specific to particular protocols, such as the WILLINGNESS parameter in OLSR.

In this section, we will give a brief overview of a number of studies conducted to investigate the protocol parameters impact on the network performance.

There have been a several studies on the impact of the OLSR HELLO Interval on network performance. Simulations done in [7], [8] and [10] suggest that there exists a trade off between throughput, overhead and power consumption when the HELLO interval parameter is 'tuned' accordingly.

Using a low HELLO interval value (i.e. a high HELLO message frequency) might improve the protocol reactivity to link failures (quick detection and fast re-route) and increase throughput, but it will have a negative impact on overhead and power consumption [8]. Therefore, a suggested adaptive mechanism is to auto-configure the HELLO interval parameter based on the link failure frequency (or node speed) in order to improve network performance. It has been found that the HELLO interval has no obvious relationship with network density [9].

Similarly for the case of AODV, from the experiments conducted in [6], the authors have shown that the HELLO interval parameter in AODV has a high impact on power consumption, overhead and network throughput. Experiments in [6] are also suggesting that an auto-configuration mechanism which tunes this parameter based on how dynamic the network is, i.e. according to link failure frequency, has the potential to increase network performance.

Maintaining an active route is one of the important features of a routing protocol. In contrast to proactive protocols, reactive WMN routing protocols such as AODV only maintain a routing table entry while the route is actively being used to forward packets between source and destination nodes. In AODV, the Active Route Timeout (ART) parameter determines after how many seconds since the last successful packet transmission, a routing table entry should be maintained [2]. If the active route timeout expires, i.e. if a route is not used to forward any packet for ART seconds, the route will be considered invalid. The purpose of this is to remove stale routes. From simulations done in [4] and [5], the ART parameter has a high impact on key network performance parameters such as the Packet

Delivery Ratio (PDR), which is the fraction of successfully delivered data packets to the total number of packets sent.

When a link breakage caused by a high levels of node mobility become more frequent, a greater ART value will degrade the network performance, since nodes keep sending packets via broken and stale routes for an extended period of time. For more stable or static networks, a greater value of ART can reduce the routing overhead, since it reduces the frequency of frequent new route discoveries.

In OLSR, the parameter called Topology Control (TC) messages are disseminated in the network to update nodes regarding any topology changes, and to allow calculation of fresh routes. The interval in which these messages are sent is the TC interval. A simulation based study has been done that shows that the TC interval has a larger impact on routing overhead [10] than the HELLO interval in OLSR, but a smaller impact on route setup time [9] than the HELLO interval parameter, and almost negligible effect on throughput [10]. These results indicate that there is a potential for parameter adaptation.

There are also non-timing based parameters that control certain aspects of WMN routing protocols. For example, in OLSR, the *WILLINGNESS* parameter is defined as the readiness and ability of a node to forward traffic. It is specified in the range of integers from 0 to 7, with 3 as the default value. A node with WILLINGNESS value of 0 will never be selected as a forwarder and a value of 7 means that it will always ready to be selected as a packet forwarder. The WILLINGNESS parameter is also important for the selection of MPR nodes. A node selected as an MPR will use a lot of resources in particular power.

The WILLINGNESS parameter can be changed adaptively. Small test-bed experiments in [11] show that the WILLINGNESS value can be changed based on the level of battery lifetime of the mobile devices, to maximise the overall lifetime of the network. Additionally, one of the key roles and challenges of WMN routing protocols is to deal with link failures and to repair routes in this situation.

AODV has two basic route repair approaches to deal with link failures. Routes can either be repaired by re-establishing a new route from scratch starting from the source node

(*Source Repair*), or they can be locally repaired by the node that detects the link break along the end-to-end path (*Local Repair*). *Local Repair* can decrease the cost and time of a route repair, and increase overall network performance.

However, in some cases, it can also result in increased path length. It is clear that an unsuccessful *Local Repair* attempt results in additional network overhead and increased time required for the route re-establishment. Depending on the situation, *Local Repair* can result in a significant increase or decrease in network performance, compared to *Source Repair*. AODV can be configured to employ different route repair mechanisms, and there have been some prior works that have evaluate the different approaches.

The default behaviour in AODV, as discussed in Chapter 2, is to use Source Repair if the link break happened close to the source, and to use Local Repair if the link break happened closer to the destination node. Simulations done by Pereira et al. [50] showed that setting up AODV to always do Local Repair results in better PDR performance than when using Source Repair in low traffic volume scenarios. However, Source Repair outperformed Local Repair and the default AODV behaviour for networks with relatively high traffic load.

Authors in [51], [52], [53] and [54] have also evaluated the different options of local repair and the results have generally shown that for low traffic load, Local Repair is better than Source Repair, but for higher load scenarios, doing Local Repair can result in better performance.

## 4.3 Parameter Adaptive WMN Routing Protocols

The behaviour and parameters of traditional WMN routing protocols is typically set statically at compile time and is not tailored to any particular deployment scenario. However, there have been a few proposals for protocols that allow protocol parameters to be adapted to the various network environments, such as [12], [13], [39], [40], [41], [43]. . This section will give an overview of the key works in this area. Most of these parameter adaptive WMN protocols are extensions of traditional WMN routing protocols.

### 4.3.1 Adaptive AODV

Adaptive AODV [39] uses the level of mobility in the network to adapt the HELLO message frequency. The idea is that in a more dynamic network topology, more frequent HELLO messages will allow to more quickly react to those changes. Node mobility is determined by periodically checking the routing table, summing up the new and lost neighbours since the last check. The node mobility parameter $N_m$ is defined as follows:

$$N_m = New_x + Left \qquad (1)$$

where $N_m$ = *node mobility*

$New_x$ = *number of new neighbours in last measurement interval*

*Left* =*number of neighbours lost during last measurement interval*

This mobility metric will be used to decide the value of the *HELLO_INTERVAL (HI)* AODV protocol parameter. Three discrete states of mobility are defined: *low*, *normal* and *high*. If the value $N_m$ reaches the threshold value of 5, the HELLO_INTERVAL parameter is set 0.75 seconds. If $N_m$ goes below 1, the HELLO_INTERVAL parameter will be set to 1.25 seconds. Otherwise, the default HELLO_INTERVAL value of standard AODV of 1 second will be used. The simulation results in [39] show some improvements in both PDR and packet latency when compared to native AODV.

### 4.3.2 ARM-DSDV

ARM-DSDV proposed in [41] aims to adapt the DSDV protocol to networks with varying levels of mobility. Similar to Adaptive AODV, it uses the rate of neighbour changes as the mobility metric. In particular, the number of changes in the 1-hop neighbourhood during the update interval is used. Each node compares its current 1-hop neighbours with the 1-hop neighbours from the last update interval and counts the number of new and lost neighbours. Each mobile node will average the mobility metric of itself and its neighbours over a time interval *TW-SMOOTH* and adjust the routing *UPDATE_PERIOD* parameter in DSDV accordingly. The *UPDATE_PERIOD* parameter determines the frequency of routing updates.

The normalized mobility metric value will be included in the routing-protocol control message. The routing update message contains a sender ID, update period and the sender's mobility metric. Performance comparisons between ARM-DSDV and DSDV presented in [41] have shown that ARM-DSDV achieves improved PDR and lower overhead compared to DSDV.

### 4.3.3 Adaptive OLSR (AOLSR)

Adaptive OLSR (AOLSR) was proposed in [13] with a mechanism that uses the frequency of link breakages to adaptively change the value of the *HELLO_INTERVAL* and *WILLINGNESS* OLSR protocol parameters. The purpose of AOLSR is to sense link changes and adapt the routing behaviour in order to increase the network performance. AOLSR uses the number of link breaks as the mobility metric and applies it to OLSR. Each node checks its link table every second, and compares the number of symmetric neighbours with the ones seen previously. Nodes keep records of link breaks over intervals of three seconds. AOLSR defines 3 states: *Default, Fast-Response* and *Fast-OLSR.* When the number of link breaks reaches an upper threshold, a node will change its *HELLO_INTERVAL* to the *FAST_HELLO_INTERVAL* value of 1 second. It will change back to *Default* if the monitored link breaks are equal or less than a lower threshold *LOWER_LINKBREAKS (*set to 1) for three consecutive measurement intervals.

A node in *Default* mode changes to *Fast-Response* when it receives a *fast hello* message from its neighbour, indicating that at least one of its neighbours is in *Fast-OLSR* mode, but not the node itself. A node will change back to *Default* mode when it no longer has a *Fast-OLSR* neighbour or it will change to *Fast-OLSR* mode the same way as in *Default* mode. The mode changing mechanism of AOLSR is illustrated in Figure 4.1.

AOLSR will not just adapt the *HELLO_INTERVAL* value based on its mobility metric, but it also changes OLSR's WILLINGNESS parameter, which is important for the selection of MPR nodes. Based on the simulation results presented in [13], AOLSR performed better than OLSR in terms of PDR.

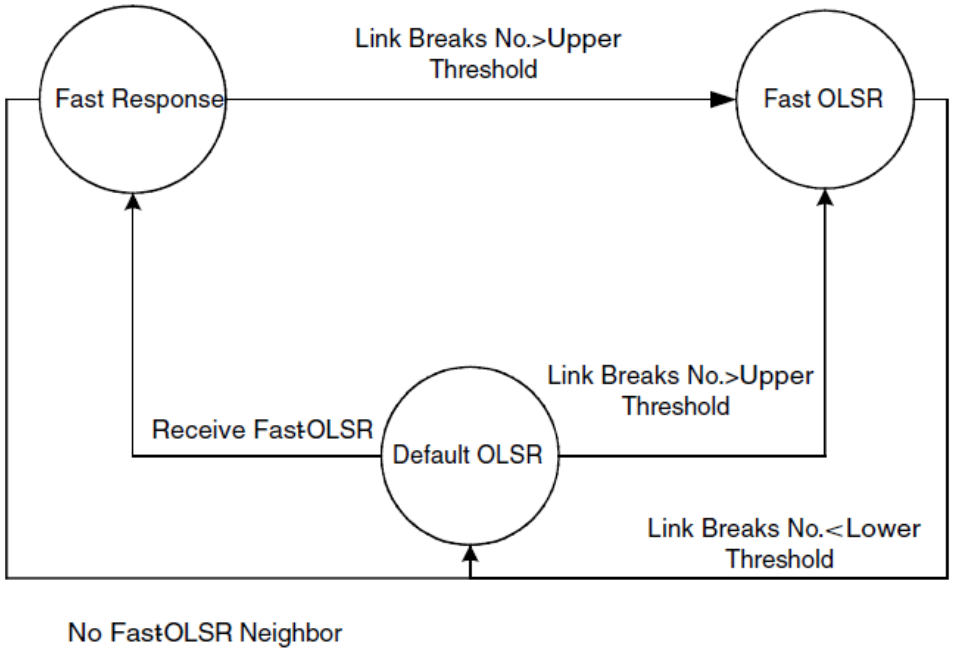



Figure 4.1: Mode switching in AOLSR [13]

### 4.3.4 Mobility Adaptive Self-Parameterization (MASP)

MASP proposed in [43] uses a mobility metric called MANET Relative Velocity Indicator (MARVIN) to measure the relative mobility of nodes in the network. The MARVIN metric is computed based on the number of changes in the 1-hop neighbourhood of a node and a weighted number of neighbours (history of previous number of 1-hop neighbours), where the individual weight of each neighbour is based on the time since the last packet was received from this neighbour. From the calculated value of MARVIN will be mapped onto suitable OLSR routing protocol parameters, i.e. the *HELLO_INTERVAL* and *TC_INTERVAL* parameters.

### 4.3.5 Link Availability Prediction AODV (PAODV)

The authors in [12] have developed a method to dynamically adapt the *Hello Interval (HI)* parameter of the AODV protocol according to the network topology, and named it Link Availability Prediction AODV (PAODV). The concept is illustrated in Figure 4.2, where *R* is the transmission range. The example on the left shows a network with nodes relatively far away from the source node in the centre, indicated in red. Since these nodes are on the edge of the transmission range, they are likely to move out of range, resulting in a change in topology.

For this potentially highly dynamic scenario, it makes sense to send HELLO messages with a high frequency, in order to be able to detect topology changes quickly.

The scenario on the right in Figure 4.2 is more stable, since all nodes are well within the transmission range of the source node, and hence the topology is more stable. In this case, a higher value of the *Hello Interval* parameter can be used. The authors propose to use GPS in order to determine the location of nodes. Based on this information, a prediction of the link lifetime can be estimated according to the method proposed in[25]. The *Hello Interval* value is largely determined by the link with shortest lifetime period and is given as

$$HI\ (i) = min\{T_L(i, j)\} \qquad (2)$$

Where $j$= *size of neighbour set of node i*

where $\boldsymbol{T_L(i, j)}$ is the predicted lifetime of the link between node *i* and its neighbour *j*.

However, this approach has several drawbacks. The requirement of each node to have GPS installed inside will increase the cost and battery consumption of nodes. Furthermore, GPS will perform poorly in indoors areas where the GPS signal is effectively 'shielded', resulting in location errors or lack of location information altogether.

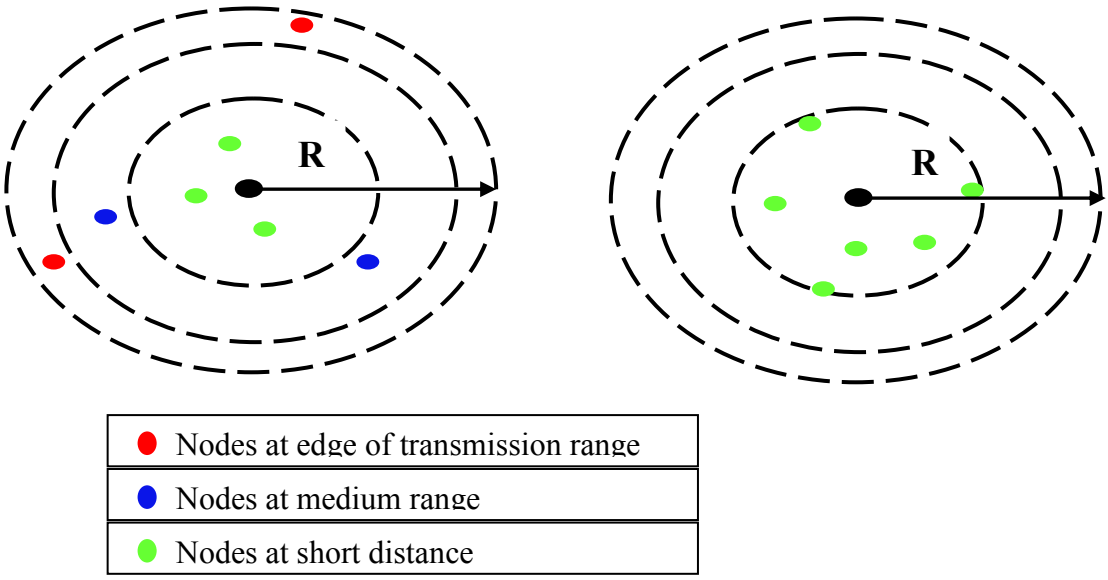



Figure 4.2: MASP Network Topology Scenarios [12]

### 4.3.6 Adaptive Hello Rate (AHR)

Similar to Adaptive AODV, *Adaptive Hello Rate (AHR)* proposed in [40] aims to adapt the *HELLO_INTERVAL (HI)* parameter depending on network conditions. AHR defines two mobility levels: *low* and *high*. This is determined via the Time to Link Failure (TLF) and Time Without Link Changes (TWC) parameters. TLF is defined as the estimated lifetime of a link, and TWC is defined as the time difference between the last link change in the routing table of a node and the current time.

*AHR* by default will be in the *low* dynamic state with the *HELLO_INTERVAL* parameter set to AODV's default value of 1second. If the estimated *TLF* parameter is lower than a defined threshold value called, it will be changed to the *high* dynamic state, with the HELLO_INTERVAL parameter set to 0.2 seconds. The protocol will revert back to the *low* dynamic state if the measured *TWC* parameter becomes greater than a given threshold value. Simulation results in [40] show an improved network bandwidth of 5-10% over standard AODV.

### 4.3.7 Summary of Parameter Adaptive WMN Routing Protocols

Table 4.1 summarizes the key properties of the discussed adaptive routing protocols; in particular the table shows the measured or estimated network condition which forms that basis for protocol adaptation. We see that all protocols use the level of mobility as a measure how dynamic the network topology is, as a basis for parameter adaptation. Most protocols use frequency of 1-hop neighbour changes as an estimator of mobility, but node location and speed is also used.

All protocols, with the exception of ARM, adapt the HELLO_INTERVAL parameter, common to both the AODV and OLSR protocols.

| Adaptive Protocols | Base Protocol | Measured Network parameter(s) | Adapted Protocol Parameter(s) |
|---|---|---|---|
| Adaptive AODV | AODV | Mobility (via frequency of changes in 1-hop neighbours) | HELLO_INTERVAL |
| ARM | DSDV | Mobility (via frequency of changes in 1-hop neighbours) | UPDATE_PERIOD |
| AOLSR | OLSR | Mobility (via frequency of Link Breaks) | HELLO_INTERVAL & WILLINGNESS |
| MASP | OLSR | Mobility (via frequency of changes in 1-hop neighbours) | HELLO_INTERVAL & TC_INTERVAL |
| PAODV | AODV | Mobility (via node speed and node location) | HELLO_INTERVAL |
| AHR | AODV | Mobility (via frequency of link failure and route changes) | HELLO_INTERVAL |

Table 4.1: Summary of Parameter Adaptive WMN Protocols

## 4.4 WMN Routing Strategy Adaptation

The previous section gave an overview of approaches to adapt specific protocol parameters to different network conditions, in particular the level of node mobility. This section looks at a different type of protocol adaptation, i.e. where different routing approaches or strategies are employed, for different regions or types of networks. Most of these protocols can be considered as hybrid routing protocols, since they combine two types of routing approaches, e.g. proactive and reactive [14], [42], [44], [45].

These protocols use different routing strategies in different regions or at different times in the same network. For instance, such strategy adaptive or hybrid protocols may benefit from forming clusters of nodes within the same network and applying different routing schemes for communications within and outside the clusters, or adapt the frequency

and size of the routing updates according to network conditions. In the following, we will discuss key examples of this type of WMN routing protocol research proposals.

### 4.4.1 SHARP

SHARP [14] is a hybrid routing protocol that combines both proactive and reactive routing. It adapts between reactive and proactive routing by dynamically varying the amount of routing information shared proactively. SHARP also considers application requirements when deciding on which routing strategy to use.

SHARP defines two zones, proactive and reactive zones, which are determined by the *zone radius*. Nodes in the proactive zone will use a TORA based proactive routing protocol, while nodes in reactive zone will use the reactive AODV protocol for route discovery. Basically, proactive zones are created automatically around destination nodes and the size of the radius of the zone is determined by the amount of incoming data traffic and the level of mobility in the network. Destination nodes that receive a large number of data packets will be called a 'favourite destination' and will have a large proactive zone radius. Destination nodes with little or no data traffic will have a small or no proactive zone, and consequently rely on pure reactive routing.

As the zone radius increases, the routing overhead will also increase, which is not surprising, since a proactive protocol has more overhead than reactive protocols. In addition to calculating the number of data packets it receives, a destination node also estimates the loss rate in the network from the count of the data packet sent and delay jitter.

These results of routing overhead, delay jitter and loss rate will influence which of the routing strategies supported in SHARP is being used. The three routing strategies in SHARP are: minimal packet Overhead SHARP (SHARP-PO) for power and bandwidth constrained networks, targeted loss rate SHARP (SHARP-LR) for loss sensitive application such as TCP, and targeted delay jitter (SHARP-DJ) for multimedia applications. Simulation results show that SHARP performs well in networks with high mobility when compared to AODV, but has slightly higher overhead in low mobility networks [14].

### 4.4.2 Chameleon (CML)

Chameleon (CML) [44] is a hybrid routing protocol combining the features of AODV and OLSR. It is an adaptive hybrid protocol specifically designed for multimedia communications in emergency response scenarios. CML adapts its routing behaviour according to the size of the network. The hybrid protocol has three modes of operation which are *proactive*, *reactive* and *oscillation* mode. The system will operate in proactive mode (denoted as *p-phase*) if the network size is less than 10 nodes, and will switch to reactive mode (denoted as *r-phase*) for a network size of more than 10 nodes.

When a node detects an increase in the network size beyond 10 nodes, while it is in p-phase, the protocol will not directly go to r-phase, but will go to the oscillation mode first for a fixed period of time. If after this time, the network size is still larger than 10, the protocol will move to r-phase. While in oscillation mode, the protocol still operates as it would in p-phase. The aim of this approach is to avoid oscillations between the r-phase and p-phase of the protocol. Note that, when a change of phase happens, a node will send a special packet called CML Change Phase (CLM-CP) packet to inform other neighbours that a phase change is taking place in the node. Simulations in [49] show that CML has lower jitter than AODV and OLSR, but has a slightly higher packet loss rate than AODV for larger networks.

### 4.4.3 Way Point Routing (WPR)

Way Point Routing (WPR) [45] is a hybrid-type hierarchical routing protocol, which maintains a hierarchy only for active routes. A number of intermediate nodes on a route are selected as *way points* and the route is divided into segments by*way points*. WPR is a combination of the DSR and AODV protocols, called *DSR over AODV* (DOA).

Nodes selected as a waypoint node will use DSR while other nodes will use AODV as their routing protocol (*intra-segment* routing uses AODV, *inter-segment* routing uses DSR). Waypoint nodes divide a route into segments. In that way, a network can improve its scalability. It is also adaptive to the level node mobility (relatively) by monitoring the number of link breaks. If nodes move slowly, indicated via no or a small number of link breaks, the network will have longer segments, in terms of number of hops per segment. Simulation results on PDR and end-to-end delay show that WPR can outperform both AODV and DSR.

### 4.4.4 Adaptive Distance Vector (ADV)

ADV [42] is a combination of the reactive AODV and proactive DSDV protocols. ADV has no explicit route repair mechanism, relying instead on the routing updates to re-establish broken routes. Unlike the periodic updates in the traditional distance vector protocols, ADV routing updates are triggered adaptively in response to network load and topology changes. Routing overhead is reduced by varying the size (full or partial updates) and frequency of routing updates in response to traffic and node mobility.

The mobility of the network, as seen by a node, is determined by the number of neighbour changes observed by the node in its 1-hop neighbourhood in a period of a fixed number of full updates. The number of nodes going out of the 1-hop range can be determined by the number of broken links, whereas those coming into range can be determined when an update is received from a neighbour whose metric is more than 1 (hop distance). If the number of neighbour changes exceeds a pre-set number, the node categorizes the network as HIGH_SPEED and as a LOW_SPEED network otherwise.

Some performance comparisons with DSDV, AODV and DSR have been done using the ns-2 simulator. It has been shown that ADV performs better than these 3 protocols in terms of overhead and latency. In terms of the PDR performance; ADV has an equivalent performance to AODV, and outperforms DSR and DSDV. The improvement of ADV is significant and more noteworthy when the node mobility is high.

### 4.4.5 Summary of Strategy Adaptive WMN Routing Protocols

Table 4.2 summarizes the key properties of each of the strategy adaptive (hybrid) WMN routing protocols discussed in this sub-section. The table lists in the second column the base protocols that form the basis of the adaptive protocol. The third column lists the network parameters that form the basis for making the protocol adaptation decisions, and column four mentions the key adaptation technique employed by the protocol.

| Adaptive Protocol | Base Protocols | Measured Network parameter(s) | Adaptation Technique |
|---|---|---|---|
| SHARP | TORA & AODV | Traffic volume, link quality | Adapting size of proactive vs. reactive zone |
| CML | AODV & OLSR | Network size | Switching between AODV and OLSR mode |
| WPR | DSR & AODV | Mobility (via frequency of link breaks) | Adapt inter and intra segment size |
| ADV | DSDV & AODV | Mobility (via frequency of changes in 1-hop neighbours) | Changing the size and frequency of routing updates |

Table 4.2: Routing Strategy Adaptive (hybrid) WMN Protocols

## 4.5 Hybrid WMN/DTN Routing Protocols

So far we have discussed the adaptive WMN protocols that are used in traditional WMN environments, where (almost) constant end-to-end connectivity between source and destination nodes is assumed. At the other end of the spectrum are DTN routing protocols, which assume a network with very intermittent connectivity. In the real world, there is more likely to be a continuum of connectivity, rather than a 'bimodal' one. One of the contributions of this thesis, discussed in Chapter 6, is to explore a new protocol that can operate efficiently in wireless multi-hop networks with a wide range of connectivity characteristics. This section gives an overview of the (limited) related works in this area; in particular of hybrid routing protocols can operate both in WMN and DTN environments.

### 4.5.1 Context Aware Routing (CAR)

Context Aware Routing or CAR [94] is a combination of a forwarding based DTN and the end-to-end connectivity based DSDV protocols. It was developed for scenarios where future movement or node connectivity is completely unknown and no geographical location information of any host is available. CAR's algorithm is built on the assumption that the only information a node has about its position is logical connectivity. Another assumption of

CAR's protocol is that any node present in the network will cooperate with each other to deliver the message. The choice of the best carrier is based on the evaluation of 'context' information available in the network. This can include a range of parameters, such as node mobility, battery level, node co-location and many more.

The transmission process of bundle messages depends on the destination node, and whether it is present in the same connected part of the network. If a destination node happens to be in the same connected network part of the network as the source node, and end-to-end path is instantaneously available, the message is transmitted using a 'synchronous' protocol to determine this end-to-end route. The end-to-end routing protocol used in CAR is DSDV. If delivery fails in the synchronous mode, the protocol switches to DTN node, and the best carrier is selected within the same network 'region', i.e. the one that is considered as having the highest chance of successful delivery. The message is sent to one or more of these carrier nodes, using the underlying synchronous scheme. Delivery probabilities are synthesised locally from local context information. As mentioned before, context is defined as a set of parameters such as connectivity change rate of a node or energy level that shows the ability of nodes to remain 'alive' to deliver the message.

Since DSDV is the proactive end-to-end protocol used in CAR, every node will periodically send both normal DSDV control message, as well as context information. When a node in range receives such a message, it will update its routing table accordingly.

For DTN routing, CAR only uses a single carrier, rather than a set of carriers. If a node is selected as a carrier, it will receive that message and it will be stored in its local message buffer. CAR requires each node to calculate its delivery probability for each destination node, based on direct observations and encounters, as well as range of indirect context information and attributes. The main task of CAR is to measure, disseminate and combine these attributes. These set of attributes, called 'utilities', is calculated using multi-criteria decision theory. A utility is associated with each context attribute. From that, utilities are then combined using a weighting function.

In [94], authors tested this approach with two sets of attributes: co-location with destination node, and change of degree of connectivity. CAR uses time series analysis using Kalman Filters to predict the network and connectivity conditions.

CAR has several limitations. Firstly, it only uses a single carrier for messages in the store-carry-forward DTN mode, which can result in a relatively low probability of message delivery, since the carrier may never encounter the destination node, or might remove the message if its message buffer is full. Secondly, CAR is based on the DSDV protocol, which has shown to perform poorly in mobile networks, even with the moderate levels of mobility [95], due to its slow response to the link breaks.

### 4.5.2 Hybrid MANET-DTN (HYMAD)

The HYMAD protocol [96] combines both traditional MANET protocols with a DTN approach by dividing nodes into several disjoint groups of fully connected MANETs. These groups will exchange data with each other. In this environment, intra-group communication, i.e. between nodes within the group, will be achieved via MANET routing. In particular, a proactive distance vector routing mechanism, similar to DSDV, is used to achieve a mesh-like connectivity in this mode.

For each group, at least one 'border node' is selected to communicate with other border nodes of different groups. Border nodes use a special flag in control messages to declare their existence and to allow discovery by other border nodes. In HYMAD, communication between groups is done via communication via borders nodes. This is done in DTN mode using Spray-and-Wait DTN routing.

Group re-formation may be needed if disconnections happen between intra-group nodes. In that case, border nodes will be re-elected to reflect the new network conditions. HYMAD introduces the parameter *Dmax*, which is defined as a group diameter, representing the maximum number of hops allowed to reach any node of a group from any other node of the group. When routes are lost due to mobility or other causes, packets will be buffered rather than dropped, and will be handled appropriately according to the Spray and Wait DTN routing mechanism.

Authors show in [96] that HYMAD can outperform native Spray and Wait routing in terms of Delay and Packet Delivery Ratio (PDR) is some, but not all considered mobility pattern scenarios. It is also shown that the performance of HYMAD is strongly dependent on the timer values of the control plane messages. HYMAD is a relatively complex protocol, mainly due to its group formation and border node selection mechanism, which also imposes significant overhead.

### 4.5.3 Integrating DTN and AODV Routing

The authors of [97] have also attempted to integrate DTN and MANET routing, but did not give their protocol a particular name. The decision about DTN versus MANET routing is made at the source node, after an attempt to discover and end-to-end route using standard AODV routing. If an end-to-end route can be established, and it is expected to be stable for the duration of its use, MANET end-to-end routing is used. Otherwise, the source node switches to DTN routing, using either Epidemic or Spray and Wait.

DTN capable nodes are predetermined in the network, and discovered during the AODV route discovery process. During the AODV route discovery process nearby nodes which are capable of DTN routing are discovered.

The limitations of this approach are as follows. First, DTN routing is only available at certain pre-determined nodes, which can limit the delivery ratio of ‘bundle message’ in DTN mode. Secondly, if a source node decides to use an end-to-end route for communication, and this route fails, there is no way to immediately switch to DTN mode and save in-transit packets. Instead, the communication needs to be completely re-established from the source node.

### 4.5.4 Native OLSR for Mobile Ad-Hoc and Disrupted Networks (NOMAD)

NOMAD has been proposed in [98] as a routing protocol for tactical military networks. NOMAD operates as proactive (synchronous) OLSR protocol in the presence of end-to-end paths, but it can transition into an asynchronous DTN mode when required. NOMAD consists of two basic components; a disruption-aware routing protocol (layer 3) based on OLSR, and a caching mechanism for packets or bundle messages, operating at a higher layer.

NOMAD introduces the concept of *real* and *imaginary* routes. Real routes are based on information from normal OLSR Topology Control (TC) messages. Imaginary routes are constructed based on information from a new type of TC messages, which inform nodes about the fact that destination nodes are disrupted, i.e. can no longer be reached via an end-to-end route. When a node looses a link to a one-hop neighbour, nodes on both sides of the disrupted link act as a proxy cache for messages in the store-carry-forward (DTN) mode. These nodes are called *DTN Selectors*. Since the disruption might involve more than a single link break, NOMAD recursively checks the topology dependency to make sure all nodes impacted on by the disruption are handled by the DTN Selector. The role of DTN Selector nodes is similar to MPR nodes in OLSR. They generate TC messages, flagged as imaginary, to signal they are DTN enabled and will forward DTN messages for the disrupted network. Routing and forwarding in DTN mode is based on the *closeness* of nodes, i.e. NOMAD makes the assumption that if a node has been closer to a destination node in terms of its real route, it has a higher probability to deliver the message via the imaginary route in DTN mode. When the end-to-end route is re-established, the new *real* route is propagated via new TC messages.

Simulation based performance evaluations were done in [98] for specific tactical combat scenarios, and the performance of NOMAD protocol was compared to the performance of both OLSR and AODV protocols. NOMAD achieves improved performance over both of these MANET protocols in the considered scenarios.

NOMAD makes a strong assumption that a disconnected node will eventually be connected back to the network. However, this can generally not be assumed. It has relatively basic and DTN routing capabilities, with only a single copy of a message being forwarded. Furthermore, it has been tailored to specific tactical military scenarios, and it is not clear how it performs in more general scenarios, topologies and mobility patterns.

### 4.5.5 Store & Forward BATMAN (SF-BATMAN)

An extension to the BATMAN MANET routing protocol [115] has recently been proposed in [100], to extend the protocols use in DTN environments.

The basic BATMAN protocol establishes routes by each node regularly broadcasting so called Originator Messages (OGMs), which are forwarded in the entire network. Unlike in

traditional link state protocols, the OGM messages do not contain any link information, and nodes do not have a global topology view and therefore do not compute end-to-end paths. When trying to send a packet to a destination node, the sending node will select the neighbour as the next hop, via which it has received the most OGM messages (with the destination node as origin) over a given time interval.

SF-BATMAN makes minimal extensions to the proactive BATMAN protocol to add a basic store-carry-forward capability. If a packet cannot be successfully sent to the next hop neighbour, as per routing table, the protocol stores the packet in a special buffer, instead of dropping it as the basic BATMAN protocol would. SF-BATMAN then regularly iterates over all the packets in this buffer, and tries to send them. Packets are forwarded to nodes in DTN mode based on packet delivery probability, which is based on the time a node had last contact with the destination node.

SF-BATMAN only forwards a single copy of the message and therefore has limited delivery probability in networks with high levels of disruptions. The simulation results shown in [100], show a maximum improvement over native BATMAN of a maximum of around 15% in the considered scenario.

### 4.5.6 Delay Tolerant – Dynamic MANET on Demand Routing (DT-DYMO)

In [101] a hybrid routing scheme composed of the DYMO MANET protocol and the PROPHET DTN routing protocol. When a source node wants to deliver a packet to a destination node, a normal DYMO route discovery mechanism is started. If a route is successfully established, the packet is delivered as per the native DYMO protocol. In case the route discovery fails, a *message carrier* node is selected, to which the packet is then forwarded using end-to-end routing. The selection of the message carrier is based on the likelihood of it having contact with the destination node. This information is gathered during the route discovery process, where nodes which are not the destination node, but are in frequent contact with the destination node, also respond to corresponding route requests. The message carrier then delivers the message via *point-to-point handovers,* to next hops with the highest delivery probability. The calculation of these probabilities is similar to the one used in the PROPHET DTN routing protocol.

The DT-DYMO protocol has a couple of limitations. Firstly, the dissemination of delivery probability information via beacon messages incurs a significant overhead. Secondly, one transmission of a packet has switched to DTN mode, the protocol does not seem to support a switch back to end-to-end mode, and continues delivery of a packet in DTN mode, even if a message carrier with an end-to-end route to the destination node is encountered.

### 4.5.7 Summary of Hybrid WMN/DTN Routing Protocols

Table 4.3 summarizes the key properties of Hybrid WMN-DTN protocol discussed above.

| Protocol | Carrier Node Selection | Key Features |
|---|---|---|
| CAR | Based on delivery probability, calculated from context information such as rate of connectivity change and energy level | Only single carrier node is selected. |
| HYMAD | Each group selects a 'border node' for intra-group communication via DTN routing. | Networks are grouped into several segments, intra-group communication uses DSDV routing, inter-group communication is based on Spray and Wait DTN routing. |
| Integrated DTN-AODV | DTN routers are pre-configured in the network. | Non DTN nodes will use standard AODV routing (not all nodes are DTN capable). |
| NOMAD | DTN selector nodes are selected at position where network segments are disconnected. | DTN selector will send information about DTN routes as extended (imaginary) OLSR TC messages. |
| SF-BATMAN | Based on the delivery probability, calculated from the last time a node had contact with destination node. | Only have single forwarding policy (single copy). |
| DT-DYMO | Based on likelihood of having contact with destination node, established during extended DYMO route discovery process. | Delivery probability calculation based on the approach used in PROPHET DTN routing protocol. |

Table 4.3: Hybrid WMN and DTN Protocol Comparison

# Chapter 5 IMPACT OF ROUTING STRATEGIES AND PARAMETERS ON NETWORK PERFORMANCE

## 5.1 Overview

Wireless multi-hop networks can be deployed in a wide range of scenarios, with widely differing network characteristics, such as network size, topology, node density, mobility and connectivity pattern, traffic pattern, etc. In addition, network characteristics can by dynamic and change significantly over time. There is no single routing protocol that can perform optimally in all these situations. In addition, these networks are often dynamic and their characteristics can evolve significantly over time.

This chapter explores how the choice of protocol mechanisms and protocol parameters impacts on the network performance for a range of wireless multi-hop network scenarios. The overarching aim is to work towards protocols that are tailored and adapted to their respective deployment scenario. In particular, this chapter investigates a parameterised route repair mechanism in AODV, and explores how the choice of the Local Repair threshold parameter, which determines the chosen route repair strategy, impacts on the overall network performance.

Furthermore, explores how the choice of different key routing protocol parameters such as the HELLO interval, as well as other protocol mechanisms, such as link break detection, affect the network performance. This evaluation is done for the following protocols: AODV, DYMO (AODVv2), OLSR and HWMP. Finally, the performance of OLSR is studied for networks with different node densities and levels of connectivity. These evaluations are based on simulation based experiment using the widely used ns-2 [57] discrete event network simulator.

## 5.2 Simulation Environment

As mentioned, the evaluations in this chapter are largely based on quantitative discrete event simulation results. Given the wide range of network scenarios that have been considered

in this (and the following) chapter, in terms of scale, mobility patterns, etc., simulation is a suitable and experimental platform.

It would have simply been impractical and too resource intensive to attempt to use real test-bed experiments for these kinds of evaluations. In addition, simulation provides the required repeatability of experiments, and does not suffer from hard to control environmental effects, such as external interference for example.

This section gives an overview of the simulation environment that has been used. As mentioned, the ns-2 discrete event simulator has been used for all the experiments. In particular, the version that was used is ns-2.34. In addition, to simulate mobile wireless networks, the mobility extensions developed by the CMU Monarch Project at Carnegie Mellon University, known as CMU extension of ns-2 [61], was used.

Ns-2 provides substantial support for a wide range of wireless and WMN protocols, such as AODV, TORA, DSR and AODV. For other protocols such as OLSR and DYMO, plug-ins are available, such as developed by the MANET Simulation and Implementation group at the University of Murcia (MASIMUM) [60].

The ns-2 simulator is written in C++ and a script language called *Object Tool Command Language* (OTcl) is used to define experiments. The outputs of the simulations are recorded in a *trace file* which can be parsed and analysed to extract the relevant performance and other relevant parameters. Users can also visualise their simulations via a program called Network Animator (NAM), which is part of ns-2.

A high level overview of the process of running a simulation in ns-2 using the CMU the mobility extensions is shown in Figure 5.1. Basically, the process involves generating the following input files to ns:

i) A mobility file that describes the movement pattern of the nodes
ii) A communication file that describes the traffic pattern

These files are then used to trigger events in the simulation run. As output, a trace file is generated with detailed information about the simulation events and results. Prior to the simulation, the parameters that are going to be traced during the simulation must be selected. The trace file is then scanned and analysed for the various relevant parameters. In this thesis, Awk and Perl scripts are used for this purpose. Finally, relevant performance parameters can be plotted. In this thesis, both GnuPlot and Microsoft Excel have been used for this.

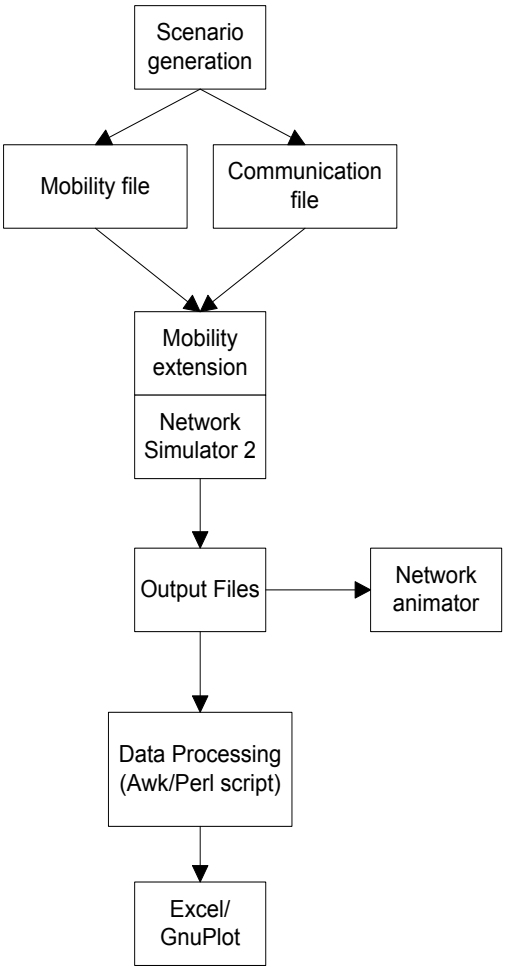



Figure 5.1: ns-2 Simulation Process Overview

Before simulations are being carried out, network parameters to be varied and performance parameters to be measured need to be determined. In our evaluations, two key

network characteristics that are varied are network topology (mobility patterns), and the network load:

**Mobility** – This is obviously one of the most important characteristic of mobile wireless networks. The level of mobility determines the network topology and level of network connectivity. In this thesis, both the ns-2 built-in topology generator, as well as the BonnMotion [104] mobility generator we used.

**Network Load –** Network load in our experiments can be characterized by three key parameters: packet size, number of flows and the packet sending rate. The traffic file is generated using the CMU ns-2 extensions, as previously mentioned.

The key performance metric that is used in this thesis, and in most related studies, is the Packet Delivery Ratio (PDR). The PDR is defined as the number of data packets that are successfully received, divided by the number of total data packets sent. Further metrics such as routing overhead and end-to-end delay are also considered, in particular in the following chapters. For experiments using random mobility models, it is important to run simulations multiple times to achieve some degree of statistical confidence about the results. If not indicated otherwise, these we use 50 simulation runs for these experiments, and the 90% confidence intervals are shown in the relevant graphs.

The following section discusses how this simulation environment has been used to explore different route repair strategies in the AODV routing protocol, and a range of network scenarios.

## 5.3 Parameterised Route Repair in AODV

A critical aspect of any WMN routing protocol is how it deals with route breaks, and how it recovers and repairs the route. This is particularly relevant for more dynamic network topologies. In this section, we investigate route repair strategies in AODV.

More specifically, the performed simulation experiments aim to provide a comparative study between the *Local Repair* and *Source Repair* route repair strategies used in AODV,

under different degrees of mobility and network load.We explore a flexible, parameterised approach for this decision making process. We consider a range of threshold parameter values for different network scenarios, in particular for different levels of network load. Our simulation results clearly show that a decision making process about route repair strategies that is more flexible and adaptive to the level of network load, can lead to a significant performance improvement. We will first start with reviewing the standard route repair approach employed in AODV. Then we discuss our evaluation of a parameterised route repair strategy.

### 5.3.1 AODV Route Repair

There are two basic approaches in which AODV can deal with a route and link break. In the first approach, the node that detects the link break sends a Route Error (RERR) message back to the source, which triggers the source node to initiate a new route discovery process and to establish a new route to the destination node from scratch. We will refer to this approach as *Source Repair*.

Alternatively, the node upstream of the link break can initiate a *Local Repair* mechanism, by locally initiating a route discovery for the destination node via broadcasting a corresponding RREQ message. For the duration of this process, data packets for the destination node should be buffered at the repairing node. If the *Local Repair* attempt is successful, the node initiating the repair will receive a RREP within the fixed amount of time (discovery period), providing a new path to the destination, and communication can resume. The scope of the RREQ messages sent as part of a *Local Repair* is limited via setting their TTL values accordingly. In case the *Local Repair* attempt is not successful, i.e. no valid RREP message is received in response to the RREQ, the repairing node will revert back to the *Source Repair* mechanism, by sending a RERR message back to the source node.

*Local Repair* can decrease the cost and time of a route repair, and increase overall network performance. However, in some cases, it can also result in increased path length. It is clear that an unsuccessful *Local Repair* attempt results in additional network overhead and increased time required for the route re-establishment. Depending on the situation, *Local Repair* can result in a significant increase or decrease in network performance, compared to

*Source Repair*. This provides the motivation for us to investigate how the decision regarding which route repair mechanism to employ can be improved.

AODV supports *Local Repair*. The decision regarding when it is invoked is based on the MAX_REPAIR_TTL parameter. The rule is as follows. *Local Repair* is invoked if the destination node is no farther than MAX_REPAIR_TTL hops away from the place where the link break occurred, otherwise *Source Repair* is chosen [2]. The MAX_REPAIR_TTL parameter is defined as follows:

$$MAX\_REPAIR\_TTL = 0.3 * NET\_DIAMETER \quad (3)$$

The default value for the NET_DIAMETER parameter is 35 [2]. This means that standard AODV chooses to do *Local Repair* if the link breaks happens 10 or fewer hops away from the destination node. As a result, AODV will always choose the *Local Repair* approach, for small to medium size networks, with a path length of no more than 11 hops. Our simulation results show that this is not always optimal.

The Dynamic On demand MANET (DYMO) routing protocol [28] is a more recent proposal, and its core functionality is largely based on AODV. One of the key differences to AODV is that DYMO does not support *Local Repair*. In case of a link break, irrespective of the location of the link break or any other relevant parameters, the route is always re-established from the source node via the *Source Repair* mechanism.

In the following, we will explore a more flexible, parameterised approach to making the decision regarding which route repair mechanism to invoke, with the ultimate goal of increasing the overall network performance.

### 5.3.2 Parameterised Local Repair

As mentioned above, the behaviour of standard AODV in case of a link break, as defined in [2], is to perform *Local Repair* if the destination node is no more than a fixed number of hops from the node that detected the link break, and perform *Source Repair* in all other cases. Rather than having this fixed and absolute threshold as a basis for deciding which route repair strategy to choose, we propose to explore a range of thresholds, expressed in

relative terms to the total path length. We then investigate what the optimal choice is of this threshold for a range of network scenarios.

First, we define the *link break location* parameter $l_{lb}$ as follows:

$$l_{lb} = \frac{hopindexofbrokenlink}{totalnumberof \quad hopsinthepath} \tag{4}$$

The hop index simply counts the number of hops in a path, starting from the source node. From the definition, it follows that $0 < l_{lb} \leq 1$.

```
if l_lb > T_LR
        Attempt Local Repair
        if successful
                resume transmission
        else
                perform Source Repair
else
        perform Source Repair
```

Figure 5.2: Route Repair method selection

We define the *Local Repair Threshold* $T_{LR}$ in terms of the *link break location* parameter, i.e. in terms of how far along the end-to-end path that a link break needs to occur in order to initiate *Local Repair*. For example, a value of $T_{LR} = 0.5$ means that if a link break happens at a link which is more than half way from the source to the destination of a path, *Local Repair* is invoked, otherwise *Source Repair* is used. Figure 5.2 illustrates how this decision is made in general.

This is further illustrated with two examples shown in Figure 5.3. The examples show a 5 hops path between node A (source) and node F (destination). In scenario a) at the top, a link break occurs at the second hop, between nodes B and C. In this case, $l_{lb} = 2/5 = 0.4$. In scenario b), the link break occurs at the 4th hop, between nodes D and E, with $l_{lb} = 4/5 = 0.8$.

In this simulation, we consider 5 different values of the $T_{LR}$ parameter, i.e. $T_{LR}$ = {0, 0.25, 0.5, 0.75, 1}.

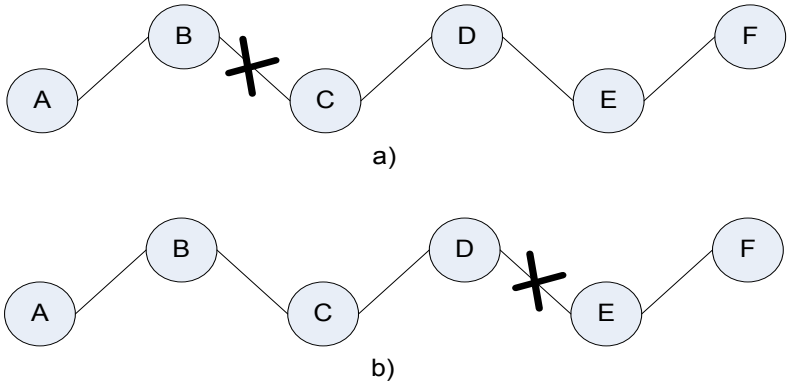



Figure 5.3: Link breaks examples

| $T_{LR}$ | Scenario a) $l_{lb} = 0.4$ | Scenario b) $l_{lb} = 0.8$ |
|---|---|---|
| 0.00 | Local Repair | Local Repair |
| 0.25 | Local Repair | Local Repair |
| 0.50 | Source Repair | Local Repair |
| 0.75 | Source Repair | Local Repair |
| 1.00 | Source Repair | Source Repair |

Table 5.1: Route Repair strategy as a function of $T_{LR}$

Table 5.1 shows the route repair approach chosen according to our proposed method (Figure 4.3), for the two scenarios in Figure 4.4, and for all the five values of $T_{LR}$ considered. For $T_{LR}$ = 0, the protocol will always choose the *Local Repair* option, irrespective of where the link break happened. Similarly, for $T_{LR}$ = 1, the chosen route repair strategy will always be *Source Repair*.

As mentioned above, for small to medium size networks (path length < 11 hops), AODV will always choose *Local Repair*, which corresponds to $T_{LR} = 0$. In contrast, DYMO will always perform *Source Repair*, corresponding to $T_{LR} = 1$.

### 5.3.3 Performance Evaluation

The objective of our experiments is to investigate the performance of route repair strategies in AODV with different values of $T_{LR}$. Using ns-2.33 [57], we simulated 50 mobile nodes moving randomly over a rectangular area of size 1500m x 300m. The mobility model used is the random waypoint model [58], with a pause time of 0 seconds, and a node speed that is uniformly distributed in [*min_speed*, *max_speed*]. In our simulations, we used two sets of values for *min_speed* and *max_speed*, i.e. [5m/s, 15m/s] and [15m/s, 25m/s], resulting in average speeds of 10m/s and 20m/s respectively.

We used constant bit rate (CBR) traffic sources in our simulations. The traffic source and destination nodes are static and are placed at both ends of the simulation area, as illustrated in Figure 5.4. By varying the number of source-destination node pairs, i.e. active data flows, from one to five, and using CBR source rates of 16Kbps and 32Kbps (with a packet size of 512 bytes), we investigate the impact of increasing network traffic load on the performance of route repair strategies in AODV. The *Local Repair Threshold* parameter $T_{LR}$ is varied from 0 to 1, in steps of 0.25. As mentioned earlier, $T_{LR} = 0$ corresponds to the protocol always choosing the *Local Repair* option, while $T_{LR} = 1$ corresponds to *always do Source Repair*. Note that as $T_{LR}$ is increased from 0 to 1.0, the likelihood of the *Local Repair* option being chosen (in the event of a link break) is decreasing.

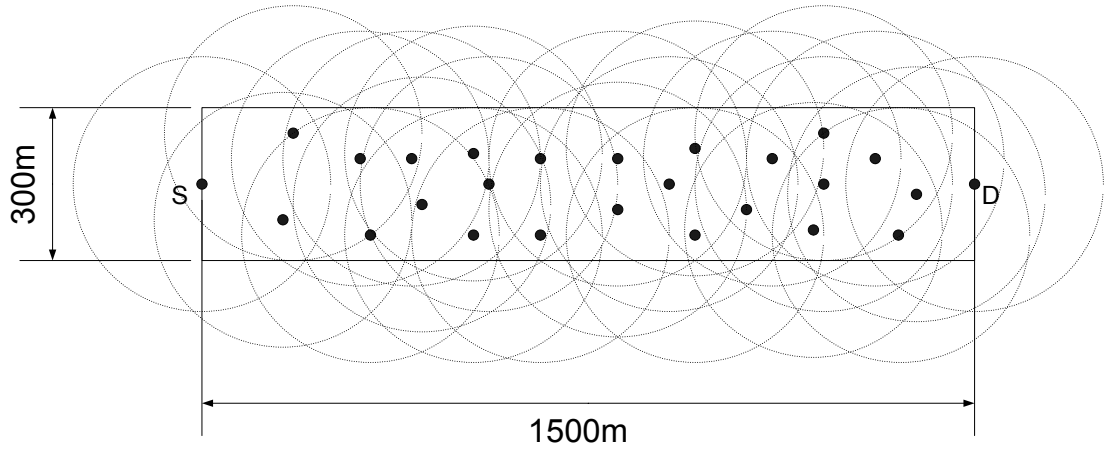



Figure 5.4: Illustration of simulation area (single pair case)

As the performance metric, we use the Packet Delivery Ratio (PDR), i.e. the total number of data packets received at destination node divided by the total number of data packets sent from the source node. For each of the scenarios considered, we performed 30 simulations runs, and we averaged the results. We include the 90% confidence intervals in our results. Table 5.2 shows a summary of the relevant simulation parameters.

| Parameter | Value(s) |
|---|---|
| Local Repair Threshold $T_{LR}$ | 0, 0.25, 0.5, 0.75, 1.0 |
| Number of nodes | 50 |
| Simulation area | 1500m x 300m |
| Mobility model | Random waypoint |
| Node speed (average) | 10m/s and 20m/s |
| Traffic type | Constant bit rate (CBR) |
| Traffic source rate | 16Kbps and 32 Kbps |
| Packet Size | 512 bytes |

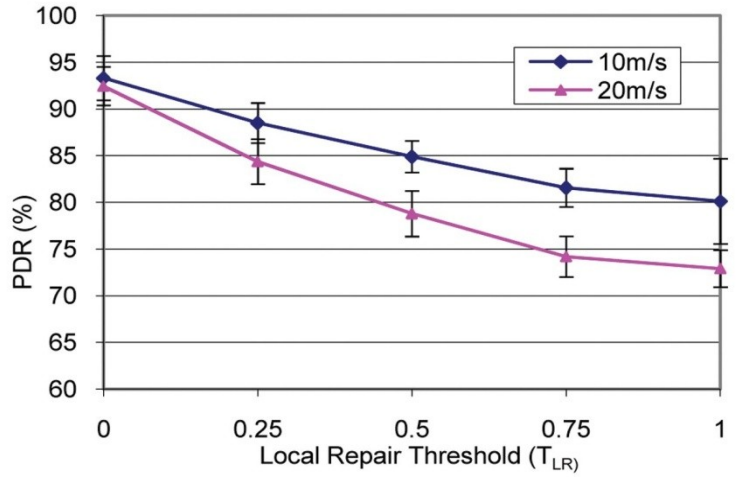
| | |
|---|---|
| Number of flows (src-dst pairs) | 1, 2, 3, 4, 5 |
| Transmission range | 250 meters |
| 802.11 MAC rate | 11Mbps |
| RTS/CTS | Enabled |
| Radio Propagation Model | Two-ray ground |
| Simulation time | 900 seconds |
| Number of simulation runs | 30 |

Table 5.2: Simulation Parameters

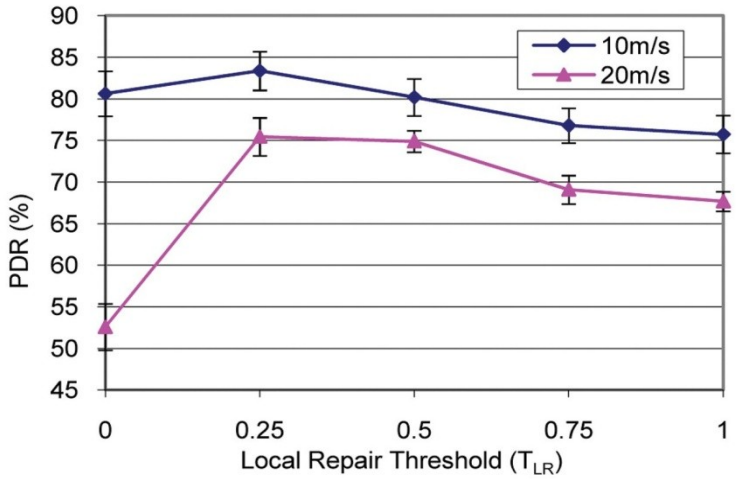
### 5.3.4 Results and Discussions

Figure 5.5 shows the packet delivery ratio (PDR) results for an increasing network load of one, three, and five 16 Kbps CBR flows, i.e. source-destination node pairs. The results are shown for average node speeds of 10m/s and 20m/s.As expected, higher node speeds lead to generally lower PDR. More interestingly, we see that for varying network loads, different route repair strategies (i.e. $T_{LR}$ values) result in the best performance. For a single 16 Kbps flow, as shown in Figure 5.5(a), a *Local Repair Threshold* of $T_{LR} = 0$, which corresponds to always performing *Local Repair*, achieves the best PDR. Increasing $T_{LR}$ monotonically decreases the PDR. The *always do Source Repair* strategy ($T_{LR} = 1$) performs worst, with a significant margin compared to $T_{LR} = 0$.

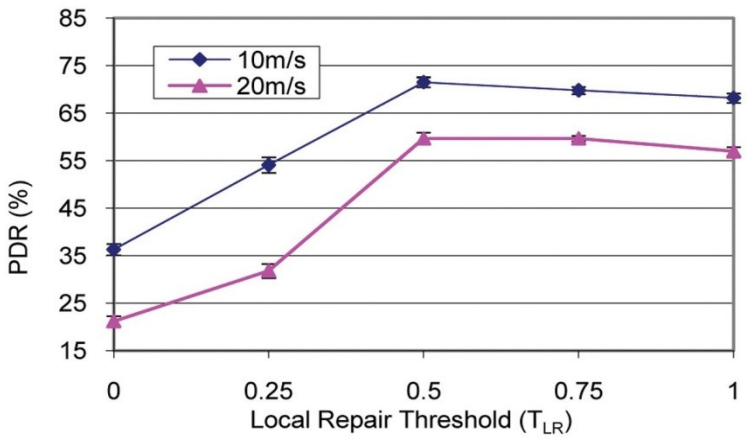
The situation changes noticeably, when the network load is increased to three and five 16 Kbps flows, as shown in Figure 5.5(b) and (c). The optimal *Local Repair Threshold* increases with increasing load, with the optimal $T_{LR} = 0.25$ for three flows and the optimal $T_{LR} = 0.5$ for five active flows. This trend is further illustrated in Figure 5.6, which shows the same results as Figure 5.5, but with a rate of 32 Kbps for the CBR flows. While for a single 32 Kbps flow, $T_{LR} = 0$ is still the best option, it turns into the worst option by a large margin for a network load of three and five flows. For both three and five flows, and both speeds of 10m/s and 20m/s, the optimal value of $T_{LR}$ is 0.75 in these scenarios.




(a) One 16 Kbps flow

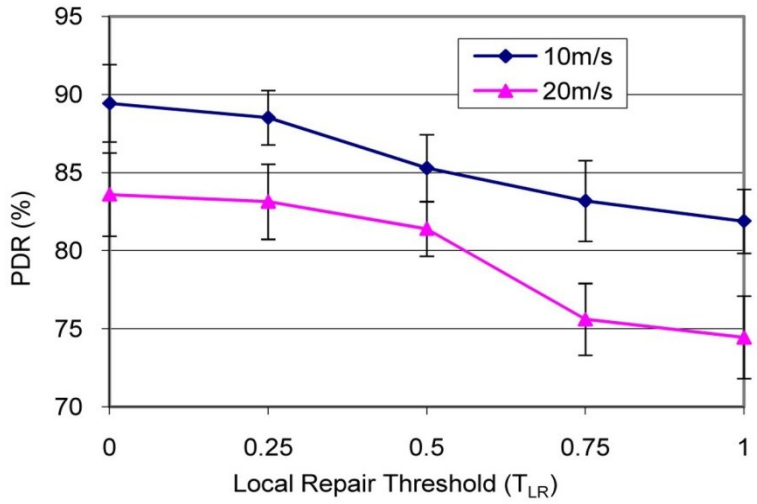



(b) Three 16 Kbps flows

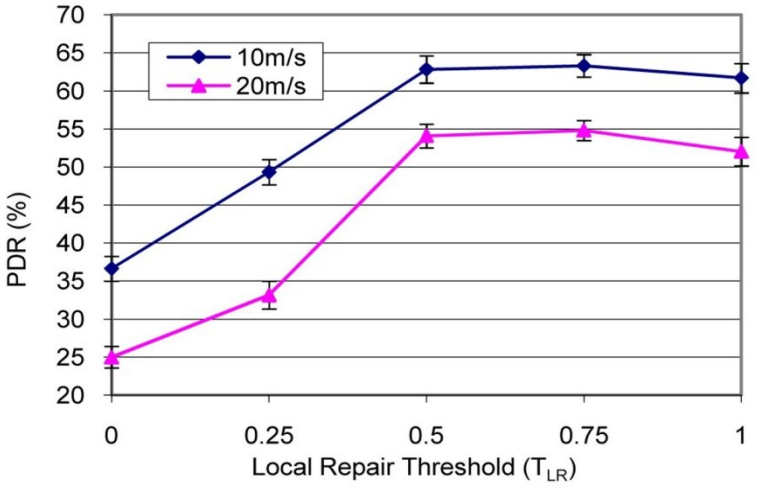



(c) Five 16 Kbps flows

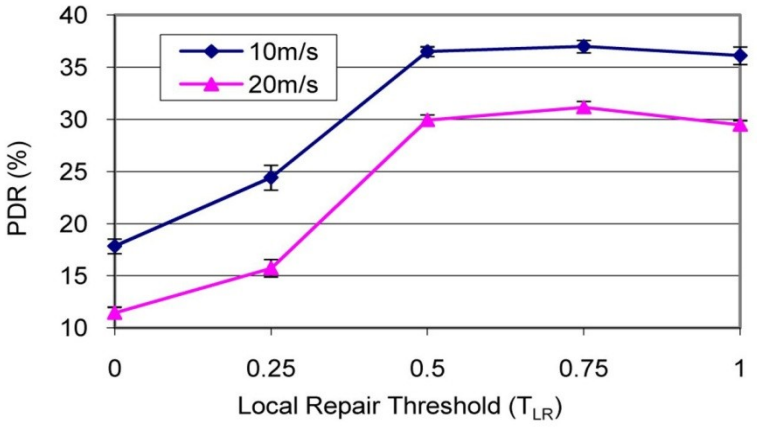
Figure 5.5: PDR vs. TLR for 16 Kbps CBR flows




(a) One 32 Kbps flow



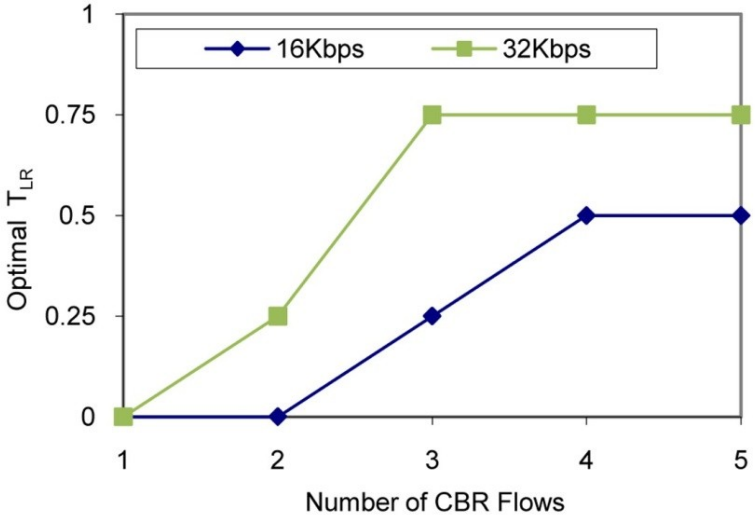

(b) Three 32 Kbps flows



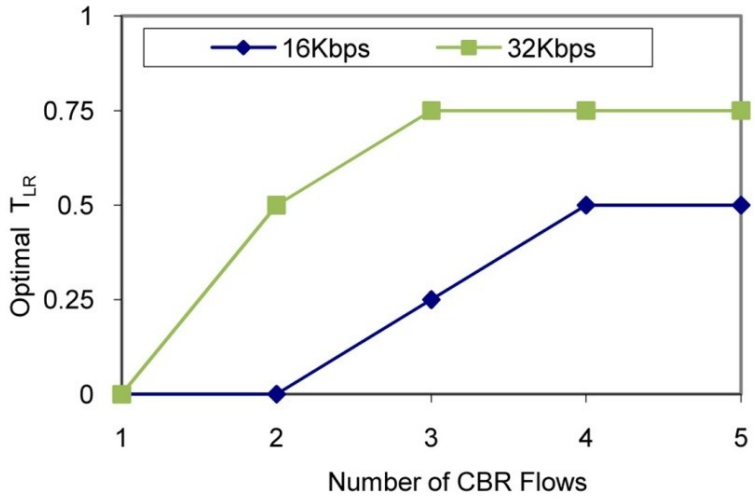

(c) Five 32 Kbps flows

Figure 5.6: PDR vs. TLR for 32 Kbps CBR Flows

From our results, there does not seem to be an obvious correlation between the level of mobility in the network and the optimal value of $T_{LR}$, and the results for 10 m/s and 20m/s are qualitatively very similar. The relationship between the level of network load and the optimal value of $T_{LR}$, in terms of the maximal achievable Packet Delivery Ratio, is summarised in Figure 5.7. As mentioned above, we consider the following set of discrete $T_{LR}$ values: {0, 0.25, 0.5, 0.75, 1}. Figure 5.7(a) shows the optimal $T_{LR}$ as a function of the number of active network flows, for an average node speed of 10m/s, and for both 16 Kbps and 32 Kbps flows. Figure 5.7(b) shows the same results, but for an average node speed of 20m/s.




(a) 10m/s average node speed

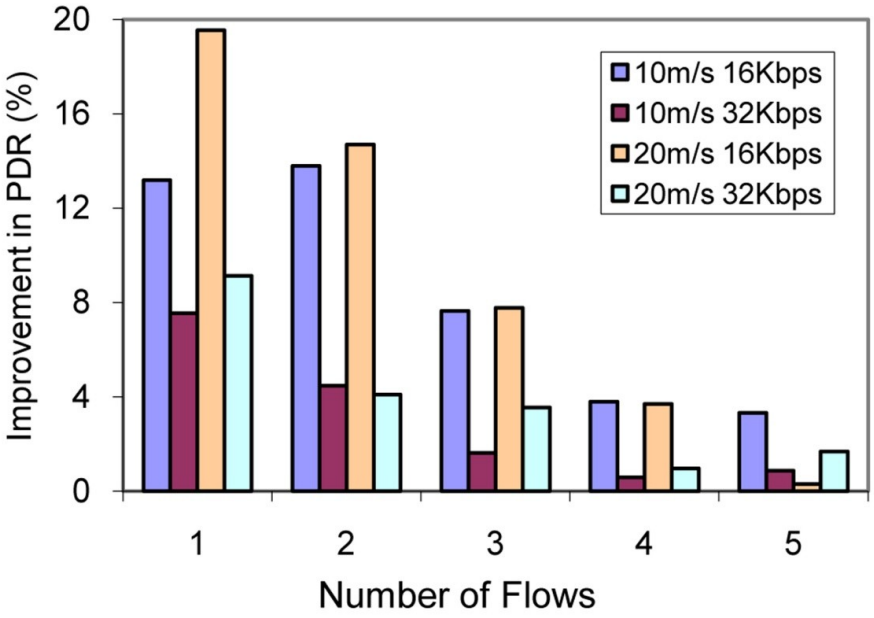



(b) 20m/s average node speed

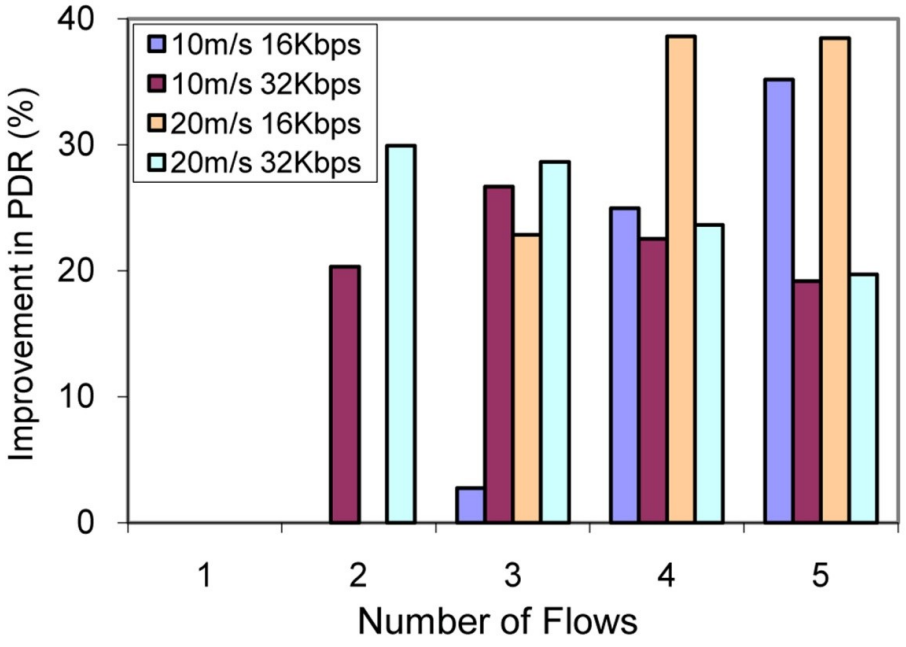
Figure 5.7: Optimal TLR for (a) 10m/s; and (b) 20m/s

We see that for a very low network load, $T_{LR}$ = 0 (*always do Local Repair*) seems to be the optimal strategy. With increasing load, the value of the optimal $T_{LR}$ increases. This means that the higher the network load, the closer to the destination the link break needs to occur, in order for *Local Repair* to be efficient. In our simulation scenarios, the optimal $T_{LR}$ never reaches the value of 1, which means the *always do Source Repair* strategy is never the best option. We are interested in the potential performance improvement that can be gained by applying an optimal, parameterised route repair strategy, i.e. by selecting the optimal Local Repair Threshold parameter $T_{LR}$. We compare the performance improvement in terms of PDR of this optimal strategy with two baseline cases in Figure 5.8. The first baseline case is the *always do Source Repair* strategy, which corresponds to a constant $T_{LR}$ = 1. This is the route repair strategy employed by the DYMO routing protocol.

Figure 5.8(a) shows the performance improvement of the optimal choice of $T_{LR}$ over the fixed choice of $T_{LR}$ = 1, for a varying network load. We see that, in the scenarios we considered, the biggest performance improvement can be achieved when the network load is low. For example, for a single 16 Kbps CBR flow at 20m/s node mobility, the optimal strategy can achieve an improvement in PDR of almost 20% (in absolute terms). The performance gain decreases with increasing network load.




(a) Optimal $T_{LR}$ versus $T_{LR}$ = 1




(b) Optimal $T_{LR}$ versus $T_{LR} = 0$

Figure 5.8: PDR gain of optimal Route Repair strategy over (a) always do Source Repair; and (b) always do Local Repair

The second baseline case to which we compare our suggested optimal route repair strategy, is the *always do Local Repair* strategy, which corresponds to a fixed $T_{LR} = 0$. As mentioned above, this is the strategy employed by standard AODV in small to medium size networks, such as that considered in our simulations. Figure 5.8(b) shows the performance gain of the optimal choice of $T_{LR}$ over a fixed choice of $T_{LR}= 0$. For a low network load (single 16 or 32 Kbps flow, or two 16 Kbps flows), the performance gain is 0, since $T_{LR}= 0$ already represents the optimal choice. As soon as the network load is increased, we see that the optimal strategy results in a significant improvement in PDR, with a maximum gain of 38% and more than 20% in most cases.

These results present a strong case for a more flexible choice of route repair strategies than is employed by current proactive MANET and Wireless Mesh Network routing protocols such as AODV.

## 5.4 Evaluation of Network Performance under different Protocol Parameter Choices

In this section, we compare the performance of four popular routing protocols; AODV, OLSR, DYMO and HWMP in terms of the Packet Delivery Ratio (PDR) metric, under a range of network scenarios with varying degrees of mobility.

We further provide an analysis of the different reasons of packet loss for the various protocols. We also investigate the potential performance improvement that can be gained by adapting critical protocol mechanisms and parameters. From our simulation results, we see that the link break detection mechanism employed by the considered protocols is critical for overall protocol performance. We therefore specifically investigated how the choice of key parameters in the link break detection mechanism affects the overall network performance. We further explore other protocol variations and features and their potential for performance improvements.

By default, OLSR, DYMO and OLSR use periodic HELLO Messages to monitor neighbour connectivity, except for HWMP which uses dedicated Peer-Link Management Protocol (PMP) [105]. Another method to detect link breaks is by using Link Layer (LL) feedback as described in [106]. For AODV, we include both version of the protocol, i.e. AODV-HELLO and AODV-LL for our comparison. For our simulations, we use AODV-UU [107], the widely used implementation of AODV by Uppsala University. For DYMO and OLSR protocol, we use the DYMO-UM and OLSR-UM version [108]. DYMO-UM utilise the Link Layer feedback mechanism while OLSR utilise the HELLO-ing technique. For HWMP, we used the ns-2 implementation from the Russian Institute for Information Transmission Problems (IITP) [110], which was the most complete and standard compliant HWMP implementation that we were able to find. However, this implementation of HWMP is very basic and does not contain the PMP link break detection mechanism. In this version, the only way to recover from a link break is to wait for the routing entry to timeout. Due to this drawback in the implementation, HWMP is shown to perform relatively badly in our evaluation, as described in the following discussion. Table 5.3 summarizes the key properties of the considered protocols.

| Routing Protocol | Type | Link Break Detection | Routing Metric | Gateway Support |
|---|---|---|---|---|
| AODV | Reactive | Hello/LL | Hop Count | No |
| DYMO | Reactive | Hello/LL | Hop Count | Yes |
| OLSR | Proactive | Hello/LL | Hop Count | No |
| HWMP | Hybrid | PMP | ALM | Yes |

Table 5.3: Key Protocol Properties

### 5.4.1 Performance Evaluation

We compare the performance of the four routing protocols in various scenarios, incorporating different cases of node mobility and traffic load. Using the ns-2 simulator, we simulated 50 nodes moving randomly over a rectangular area of size 1500m x 300m. The well known random-waypoint mobility model was used for this, in which nodes randomly choose a destination to move to, with a constant node speed that is uniformly randomly chosen in the interval [0, MAX_SPEED]. Once the node reaches the destination, it pauses for the time PAUSE_TIME, before repeating the whole process. Multiple traffic flows are generated between uniformly randomly selected pairs of nodes.

In our simulations, we measure the performance of the routing protocols in terms of the Packet Delivery Ratio (PDR) metric. PDR is defined as the ratio of the total number of data packets received at the destination node, to the total number of data packets sent from the source node. We also investigate in detail, the statistics and reason for the data packet loss in each protocol. All protocols are evaluated using the ns-2.34 simulator, with the exception of HWMP which is evaluated with the ns-2.33 simulator. Table 5.4 shows a summary of the relevant simulation parameters. In our simulations, we perform 50 runs (corresponding to 50 different random mobility patterns) for each pause time, and the results are averaged over these runs. We also report the 90% confidence interval in our results. In our 900s long

simulations, traffic flows and data tracing are only activated after 300s of "warm-up time", in order to ensure that the simulated network has reached steady state.

Figure 5.9 shows the PDR performance of the routing protocols for a traffic load of 30 flows, and with a maximum node speed of 20m/s. We have performed simulations for different number of traffic flows and maximum node speeds (as shown in Table 5.4), and the PDR performance results achieved in these scenarios are similar to that shown in Figure 5.9. For low values of pause time, which corresponds to higher levels of node mobility, we can see a very significant difference between each protocol in terms of their PDR performance. With more than 95% PDR, AODV-LL has the best performance, followed by DYMO and AODV-HELLO. We believe this is due to the AODV-LL and DYMO protocols using the Link Layer feedback mechanism, which provides immediate notification of link breaks, as soon as a packet transmission fails. On the other hand, in AODV-HELLO, nodes have to wait for two consecutive HELLO messages to be lost (corresponding to two seconds), before it can determine that the link is broken.

| | |
|---|---|
| Number of Flows | 30 flows |
| Packet Size | 64 bytes |
| Source rate (CBR traffic) | 4 packets/s |
| 802.11 MAC TX Rate | 11Mbps |
| Transmission Range | 250 metres |
| Propagation Model | Two Ray Ground |
| MAX_SPEED | 20m/s |
| PAUSE_TIME | 0, 30, 60, 120, 300, 600, 900 sec |

Table 5.4: Simulation Parameters

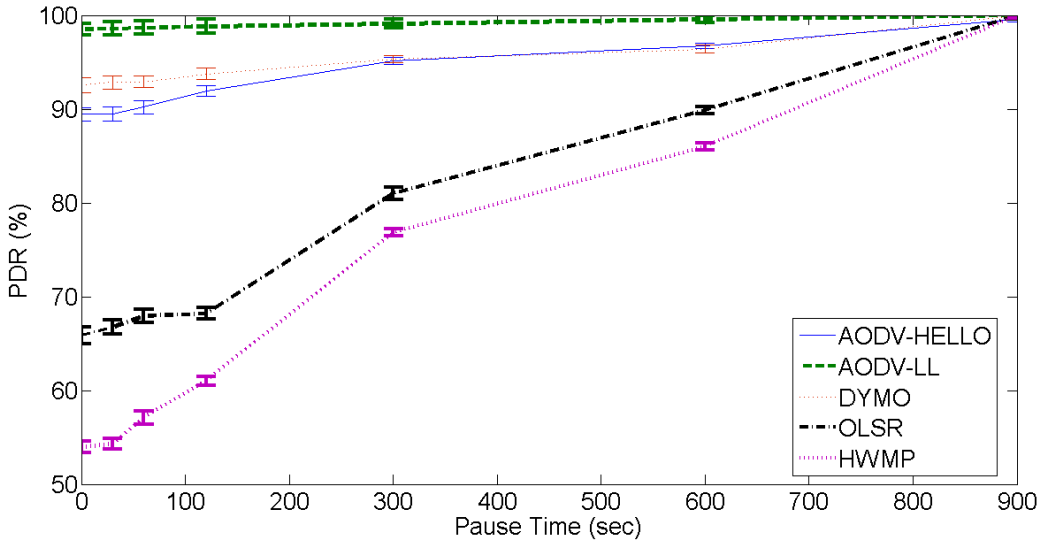



Figure 5.9: PDR vs. Pause Time for 30 flows with max speed 20m/s

OLSR and HWMP perform relatively poorly, especially at higher rates of mobility (corresponding to low pause time). At 0 sec pause time, OLSR and HWMP delivered only 66% and 55% of packets respectively. For OLSR, we believe that its poor performance is due to its slow detection of link breaks. The default OLSR implementation uses the HELLO message mechanism to detect link breaks, which leads to a high delay in the update of the routing table. For HWMP, as mentioned before, the ns-2 implementation that we used in our evaluations does not have a proper PMP link detection mechanism, and hence fails to effectively discovery link breaks. As a result, HWMP performs the worst among all the protocols that we evaluated.

Figure 5.9 also shows that as the pause time increases, the difference in the PDR performance among the four protocols decreases. This is due to the fact that lower node mobility results in a lower number of links being broken, resulting in a more stable network, and hence a higher number of successfully delivered packets.

In order to get a better understanding of these results, we investigate the reasons for packet losses in our scenario for each routing protocol. We take a detailed look at the ns-2 traces for the 0 sec pause time (30 flows) scenario for each protocol since this scenario shows the worst PDR performance. We are interested to see what the main reasons for packet loss

are in this case. The ns-2 trace file shows the following reasons: MAC transmission retries exceeded (RET), no route or invalid route error (NRTE), interface queue buffer exceeded (IFQ), routing loop (LOOP) and Time-To-Live (TTL) field reaching zero. A brief explanation on these reasons of packet drop is given in Table 5.5. Table 5.6 shows the packet drop statistics in detail for each protocol, where the values shown are the average of the 50 simulation runs, rounded to the nearest integer.

| **Types of Packet Loss** | **Explanation** |
|---|---|
| RET | After the maximum number of failed retransmission attempts, i.e. without getting an ACK, the MAC layer drops the packet and the routing layer is notified. |
| NRTE | A packet is dropped due to the fact that there is no route to the destination available. |
| IFQ | The packet is dropped since the interface queue buffer (IFQ) is full. |
| LOOP | The packet is dropped due to a detection of a routing loop. |
| TTL | The packet is dropped due to expiry of its time-to-live field. |

Table 5.5: Packet Loss Reason

We see that AODV-LL has the lowest number of packet drops. This corresponds to its high PDR performance observed in Figure 5.9. We also see that the Link Layer feedback mechanism clearly outperforms the HELLO message mechanism in detecting link breaks in the AODV protocols. This is because the number of RET and NRTE packet drops in AODV-HELLO is almost 9 times higher than that of AODV-LL.

For the DYMO protocol, Table 5.6 shows that it has the highest number of NRTE packet drops compared to the other protocols. We believe this is due to the path accumulation feature in DYMO. The path accumulation feature enables routing information of other participating node to be appended to a DYMO control message, as it passes through those

nodes on their way to the source and destination nodes during a route discovery process. This additional routing information can be used to create entries in the routing tables of nodes that process the DYMO control message, and hence can later reduce the number of route discovery attempts to those participating nodes. However, in a highly dynamic network environment, these routing entries can easily become stale or out-dated. This then causes packets to be dropped when the nodes attempt to send packets over these invalid routes.

| Protocol | RET | NRTE | IFQ | LOOP | TTL |
|---|---|---|---|---|---|
| AODV-HELLO | 6131 | 440 | 0 | 0 | 0 |
| AODV-LL | 712 | 53 | 0 | 0 | 0 |
| DYMO | 1092 | 3369 | 0 | 0 | 0 |
| OLSR | 19607 | 968 | 4 | 35 | 156 |
| HWMP | 28397 | 0 | 18 | 0 | 0 |

Table 5.6: Packet Loss Reason Statistics

For the OLSR protocol, we see that the highest number of packet drops is due to RET, i.e. the failure of the MAC layer to deliver the packet to the next hop. As mentioned before, OLSR depends on the HELLO message mechanism to detect link breaks, whereby if a node does not receive a HELLO message from its neighbour within a specific amount of time, it declares the link to its neighbour is broken, and will then invalidate the route entry corresponding to that neighbour. Therefore in OLSR, even after a link break occurs, a node may continue to send packets over the broken link until it finally determines that the link is broken due to the missing Hello messages. This is the cause of the high number of RET packet drops in OLSR. It is also interesting to note that OLSR is the only routing protocol that has packet drops due to LOOP and TTL. This is because OLSR (as a proactive routing protocol) is known to be susceptible to creating routing loops in a network where nodes are highly mobile [109].

As for the HWMP protocol, we can see that the highest number of packet drops is also due to RET. As explained before, the HWMP implementation used in our simulations does not have an effective link break detection mechanism, and solely depends on its path timeout feature to remove stale routing entries. Hence, nodes will keep on sending packets over an invalid route until the route entry expires, thereby leading to a high number of RET packet drops in the network.

### 5.4.2 Performance Enhancement

The previous section has evaluated the performance of key protocols and has tried to gain an insight into the reasons for observed level of performance. In this section, we aim to investigate if and how the performance of some of these protocols can be improved. For this, we adapt some parameters in the AODV-HELLO and DYMO routing protocols to see the impact it has on the network performance. For the OLSR protocol, a comparison is made between the Link Layer feedback and the HELLO message based link break detection mechanism.

**AODV-HELLO**

For the AODV-HELLO protocol, we identified two important parameters that control the determination of link connectivity based on the periodic HELLO messages, i.e. HELLO_INTERVAL and ALLOWED_HELLO_LOSS.

HELLO_INTERVAL is defined as the time interval between consecutive transmissions of HELLO messages, while ALLOWED_HELLO_LOSS is defined as the number of HELLO_INTERVAL periods that can lapse without receiving a HELLO message, before a node decides that the link to its neighbour is broken. In AODV [2], the default value for the HELLO_INTERVAL parameter is one second, while the default value for ALLOWED_HELLO_LOSS is two. For the link break detection mechanism, these two parameters are obviously closely related. For the purpose of your investigations, we define the Link Break Detection time parameter $L_{lb}$ as follows:

$$L_{lb} = HELLO\_INTERVAL \times ALLOWED\_HELLO\_LOSS \qquad (5)$$

The parameter identifies the time after which a link is considered broken, if no HELLO messages are received. In the following simulations, we vary the HELLO_INTERVAL and ALLOWED_HELLO_LOSS parameters according to Table 5.7, which also shows the corresponding value of $L_{lb}$.

Figure 5.10 shows the PDR performance of AODV for these 5 different parameter pairs. From the figure, we see that there is a relationship between the PDR metric and the $L_{lb}$ parameter. Protocols with a lower value of $L_{lb}$ seem to perform better than the ones with a higher value, in most cases. The difference is bigger for high mobility scenarios, i.e. scenarios with low Pause Time. This is not surprising, since a lower value of $L_{lb}$ means that nodes are quicker to detect link breaks, and therefore quicker to react to topology changes.

However, for the two cases where $L_{lb}$ = 1s, we see that the scenario with HELLO_INTERVAL = 0.5s performs somewhat better than the scenario with ALLOWED_HELLO_LOSS = 1. This may be explained as follows. The scenario where HELLO_INTERVAL = 0.5s requires the loss of two consecutive HELLO messages before a link can be declared as broken. On the other hand, in the scenario where ALLOWED_HELLO_LOSS = 1, a loss of a single HELLO message will trigger a link break and the invalidation of a route entry. We note that in our simulations, there is a total of 30 traffic flows, and due to the relatively high probability of collisions of data packets in these flows, a single HELLO message can easily be lost. Therefore, rather than the HELLO message being lost due to node mobility, it is lost due to collision. As such, in the latter case, even though the link is still existent, it is unnecessarily declared broken and causes the route entry to be invalidated, leading to a poorer PDR performance. For the same expected link break detection time $L_{lb}$, the version with a lower HELLO_INTERVAL is more robust packet loss. However, this comes at a cost of a slightly higher overhead due to the higher number of HELLO messages that are exchanged in the network.

| HELLO_INTERVAL | ALLOWED_HELLO_LOSS | $L_{lb}$ |
|---|---|---|
| 0.5s | 2 | 1s |
| 1s | 1 | 1s |
| 1s | 2 | 2s |
| 1s | 3 | 3s |
| 2s | 2 | 4s |

Table 5.7: Varying AODV-HELLO Parameters

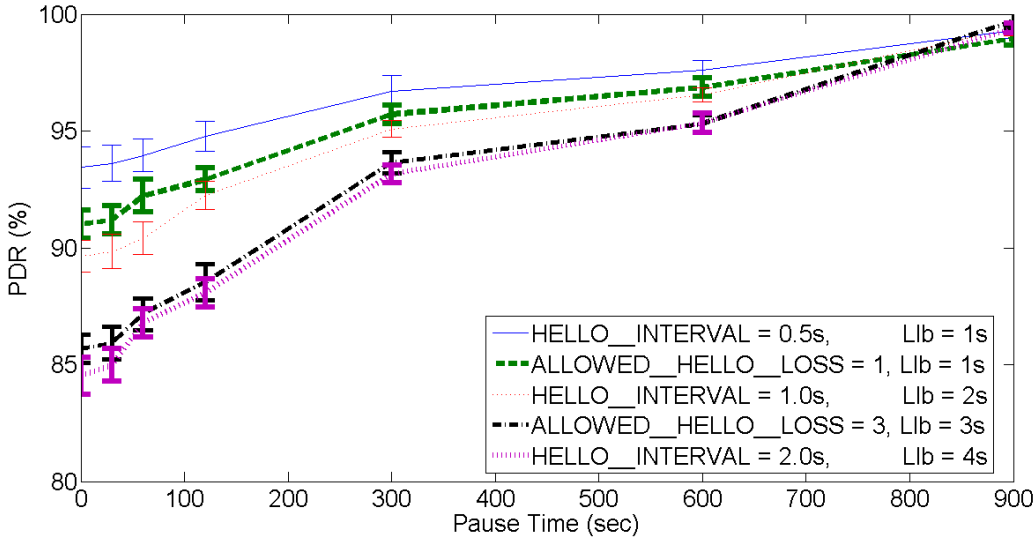



Figure 5.10: PDR vs. Pause Time for variants of AODV-HELLO

**DYMO**

For the DYMO protocol, we measure the impact of the Path Accumulation (PA) feature and the RREQ_TRIES parameter on the PDR performance. RREQ_TRIES is the parameter that determines how many times a node will try to discover a path to a destination node. In the ns-2 implementation of DYMO, the PA feature is enabled and the RREQ_TRIES parameter is set to 1. Therefore, a node will only perform the route discovery process once,

and if unsuccessful, it will assume no route is available and drop the corresponding data packets in its buffer. We evaluate four scenarios in our simulation:

- DYMO-PA: DefaultRREQ_TRIES setting with path accumulation enabled
- DYMO-noPA: Default RREQ_TRIES setting with path accumulation disabled
- DYMO-RREQ2-PA: RREQ_TRIES = 2 and with path accumulation enabled
- DYMO-RREQ2-noPA: RREQ_TRIES = 2 and with path accumulation disable

From the results in Figure 5.11, we observe that by disabling the PA feature in DYMO, DYMO-noPA achieves a PDR performance that is about 2.5% better compared to DYMO-PA in the case of 0 sec pause time. By increasing the value of RREQ_TRIES to 2, the PDR performance of DYMO-RREQ2-noPA improved further by around 4% compared to DYMO-PA. Table 5.8 shows the packet drop and routing overhead statistics, based on traces from the scenario with 0 sec pause time. The first two result columns show the number of packet drops due to RET (failed MAC layer delivery) and NRTE (no route to destination) cases.

We see that for the best setting of DYMO-RREQ2-noPA, the number of NRTE packet drops has gone down by 65% compared to the default setting of DYMO-PA. However, there is a small cost for disabling the PA feature, in terms of increased routing overhead, i.e. the number of control (Route Request) messagesthat need to be sent to discover routes. From the table, we see that when PA is disabled, the number of control messages is increased by around 10-15%. In this case, we believe that this is an acceptable trade-off in order to achieve higher PDR performance. From these results, we conclude that both the path accumulation feature and the optimal RREQ_TRIES parameter setting in the DYMO protocol are crucial to its performance improvement, and by administratively or adaptively controlling them, network performance can be significantly improved.

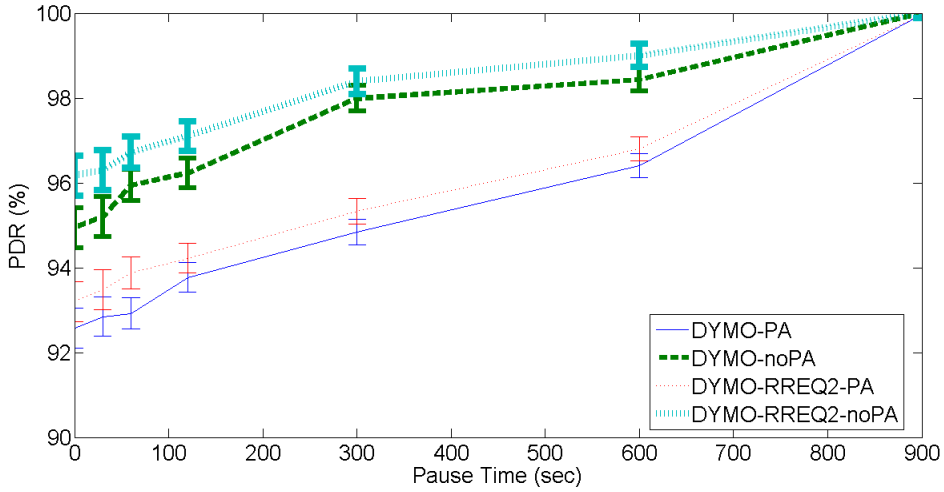



Figure 5.11: PDR vs. Pause Time for variants of DYMO

| DYMO Variants | RET | NRTE | RE Msgs. | % Rise in RE Msgs. | % Drop in NRTE |
|---|---|---|---|---|---|
| DYMO-PA | 1092 | 3369 | 1825410 | - | - |
| DYMO-noPA | 1145 | 2015 | 2093690 | 14.7% | 40% |
| DYMO-RREQ2-PA | 1097 | 2971 | 1827100 | 0.09% | 11.8% |
| DYMO-RREQ2-noPA | 1134 | 1176 | 2064130 | 13.1% | 65% |

Table 5.8: Packet Drop and Routing element (RE) Statistic for DYMO

**OLSR**

The implementation of OLSR evaluated previously (results shown in Figure 5.9) uses the HELLO message mechanism for link break detection (OLSR-HELLO). In Figure 5.12, we compare its performance with OLSR-LL, which uses the Link Layer feedback mechanism to detect link breaks. As we can see, OLSR-LL significantly outperforms OLSR-HELLO, particularly in the high mobility scenarios. For example, in the 0 sec pause time scenario, OLSR-LL achieves a PDR of 93%, compared to 66% achieved by OLSR-HELLO. The LL feedback mechanism enables a node to immediately detect a link disconnection to its neighbor

node, hence allowing it to quickly update its routing table. It also prevents the node from using the disconnected link to send packets. Table 5.9 shows the statistics for the packet drop reasons in OLSR-HELLO and OLSR-LL. We see that the number of packet drops due to failed MAC layer transmission (RET) is the main reason for packet loss. The number of RET losses decreases drastically for OLSR-LL compared to OLSR-HELLO. The large number of RET based packet losses in OLSR-HELLO, is due to the fact that the protocol keeps using stale routes for longer, and keeps sending packets across disconnected links, resulting in failed transmissions (RET).

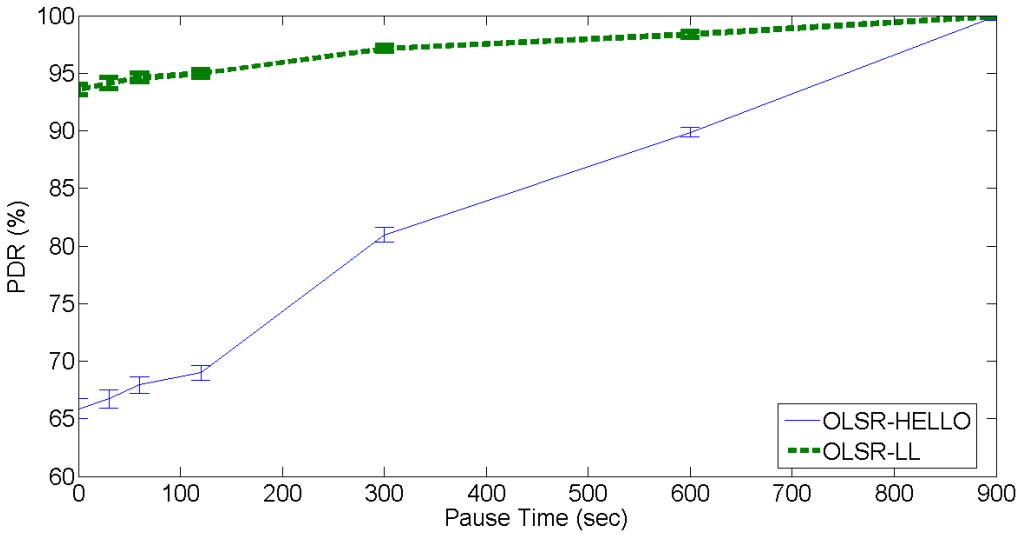



Figure 5.12: PDR vs. Pause Time for variants of OLSR

| OLSR Variants | RET | NRTE | IFQ | LOOP | TTL |
|---|---|---|---|---|---|
| OLSR-HELLO | 19607 | 968 | 4 | 35 | 156 |
| OLSR-LL | 2828 | 932 | 0 | 32 | 235 |

Table 5.9: Packet Drop Statistic for OLSR

## 5.5 Analysis of OLSR Performance in Various Topology Size

In the previous section, we have presented our results of the impact of protocol mechanism and parameter choice on network performance. We have shown that there is a potential for performance improvement if certain protocol parameters are tuned accordingly. However, the results are presented for a fixed area of 1500m x 300m, with a fixed number of nodes, resulting in a constant average node density. In this section, we want to explore protocol performance for a wider range of cases, and in particular sparser network topologies.

We will vary the topology area, but we will use the same number of 50 nodes in our simulation scenarios, resulting in different node densities and levels of connectivity. We generate our mobility scenarios from a small area (very dense network) to a bigger topology area (very sparse network). Such sparse networks can be considered as scenarios for Delay Tolerant Networks, with very limited and intermittent connectivity.

We are using OLSR for our evaluation. In the following, we will present our initial results and approach used for our simulations. We also vary OLSR's HELLO_INTERVAL parameter to see the impact on the network performance in different topology areas. Four types of network topologies with different areas are randomly generated using the ns-2 simulator, T1 (500m x 500m), T2 (1000m x 1000m), T3 (2000m x 2000m) and T4 (4000m x 4000m). For each topology type, we generate 50 random mobility patterns, and results are averaged over these 50 simulation runs.

The pause time is set to 0 sec so that the nodes are always in movement. We use 10 traffic flows (Source-Destination pairs) which are uniformly randomly selected among the nodes in the network. We use default OLSR protocol parameter settings, HELLO messages for link break detection. Further details on simulation parameters are shown in Table 5.10.

| Parameter | Value(s) |
|---|---|
| Number of nodes | 50 |
| Simulation area | 500m x 500m (T1)<br>1000m x 1000m (T2)<br>2000m x 2000m (T3)<br>4000m x 4000m (T4) |
| Mobility model | Random waypoint<br>(0s pause time) |
| Node speed (maximum) | 20m/s |
| Traffic type | Constant bit rate (CBR) |
| Traffic sending rate | 4 packets per second |
| Packet Size | 512 bytes |
| Number of flows | 10 |
| Transmission range | 250 meters |
| 802.11 MAC rate | 11Mbps |
| Simulation time | 500 seconds |
| Number of simulation runs | 50 |
| HELLO_INTERVAL | 1 second |

Table 5.10: Simulation Parameters

Figure 5.13 shows the PDR results for the four types of topologies, with different level of node density and level connectivity. Not surprisingly, we can see that there is a significant drop in PDR, for increasing topology area with a fixed number of nodes. In the case of T1, the PDR is well above 90%, while in the case of T4, the PDR approaches 0%. It is obvious that these relatively sparse networks, traditional WMN routing protocols perform extremely poorly. This provides the motivation for our investigations in the next chapter, where we explore the idea of enhancing WMN protocols with DTN store-carry-forward capabilities, in order to increase their performance in sparser network scenarios, where end-to-end connectivity cannot be assumed.

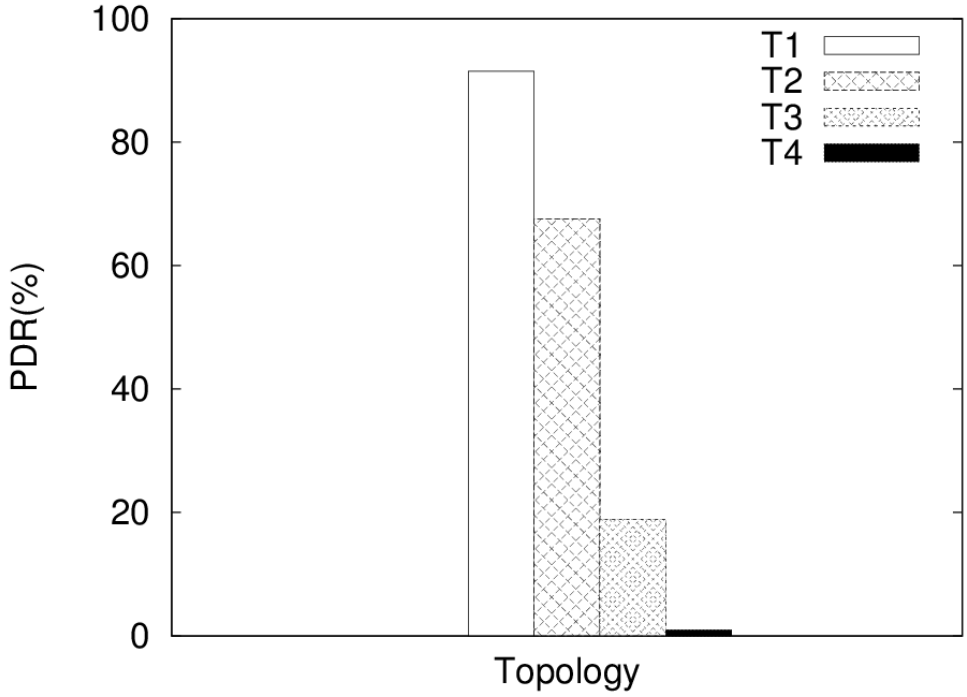



Figure 5.13: OLSR PDR performance for topologies with different node densities

## 5.6 Summary

In this chapter, we have shown that the choice of protocol parameters and protocol mechanisms, such as for link break detection and route repair, have a significant impact on network performance. Furthermore, the optimal choice of protocol parameters depends on and varies with the network characteristics, such as traffic load, mobility pattern etc.

This confirms for the potential of protocol tuning and adaptation to the characteristics of deployment scenarios. We also show the impact of various topology sizes and levels of network density on the performance of OLSR. The observed dramatic decline in performance for networks of increasing sparseness, provides the motivation for the work presented in the following chapter, where we investigate the potential of a hybrid WMN/DTN routing protocol, which combines the end-to-end routing of WMN protocols with the store-carry-forward mechanism of DTN protocols. The goal is to improve protocol performance for sparse network scenario, with intermittent connectivity.

# Chapter 6 OPPORTUNISTIC ROUTING

## 6.1 Overview

In the previous chapter, we demonstrated the impact of protocol parameters on network performance and the importance of link break detection mechanism in a routing protocol [111]. We showed that a potential network improvement is achievable if certain protocol parameters are configured accordingly. However, in some scenarios with high node mobility or with sparse network topology, tuning parameters in traditional WMN protocols may not solve the problem. As discussed in the last chapter, the performance of OLSR (in terms of PDR) degrades dramatically when the network becomes sparse. In these cases, end-to-end routes are difficult to be established. WMN protocols such as OLSR that rely on end-to-end connections will suffer and they will start to drop packets, ultimately resulting in lower PDR.

As discussed in Chapter 3, there are protocols that have been specifically designed for highly disconnected networks. These Delay Tolerant Networks (DTNs) rely on node mobility and opportunistic encounters between nodes to forward packets in a store-carry-forward fashion. These DTN or Opportunistic Routing protocols are very distinct from WMN protocols, and have been tailored for operation in networks where end-to-end routes cannot be assumed to exist at any point in time.

The goal of this chapter is to explore a protocol that can operate on networks with a wide range of connectivity levels, ranging from highly connected networks (traditional WMN scenario), to a medium level of connectivity, as well as highly sparse and disconnected networks (DTN scenario). The goal is to design a protocol that is simple and backwards compatible with standard WMN routing. Another critical goal is to not require any change to the packet format or the introduction of any special signalling in the network, in order to reduce complexity and overhead.

The hybrid protocol presented in this paper is based on OLSR [3], with an extension that provides a store-carry-forward mechanism that is inspired by the Spray-and-Wait DTN protocol [80]. We call this protocol *OLSR-OPP*, for OLSR with Opportunistic routing and

forwarding extension. The term Opportunistic has a double meaning in OLSR-OPP. Firstly, it applies to the opportunistic aspect of store-carry-forward routing in DTN protocols such as Spray-and-Wait. The second meaning represents the capability of OLSR-OPP to switch opportunistically between standard WMN routing and store-carry-forward mode, depending on the level of connectivity available in the network.

This chapter presents the OLSR-OPP and describes its operation. The performance of OLSR-OPP is systematically evaluated over a wide range of topologies, from dense to very sparse, and compared to the performance of standard OLSR as well as the Spray-and-Wait protocol.

## 6.2 The OLSR-OPP Protocol

### 6.2.1 OLSR Key Features Revisited

Since OLSR-OPP is built on OLSR as the base protocol, we first provide a brief summary of the key mechanisms of OLSR. As discussed in Chapter 2, OLSR is a table driven proactive routing protocol that is widely used in wireless ad-hoc networks. Topology information is disseminated via the use of HELLO and Topology Control (TC) messages. Due to its proactive nature, OLSR provides topology information to all participating nodes in the network, and global topology information and routes are maintained at all times. As we will see, this is an advantage for opportunistic routing, compared to reactive routing protocols. OLSR supports link break detection via both HELLO messages as well as Link Layer feedback. In case OLSR does not have a route for a packet, it simply drops it. There is no need and no point in trying to repair the route, since in contrast to reactive protocols such as AODV, we can assume that OLSR has a global topology view anyway, and would know about a route to the destination, if it existed.

### 6.2.2 OLSR-OPP Concept

The overall goal of OLSR-OPP is to increase the packet delivery ratio of basic OLSR in networks with intermittent connectivity. As mentioned in the previous section, if OLSR does not have a route to the destination of data a packet it simply drops it. The basic idea behind OLSR-OPP is to buffer these otherwise dropped packets, and to attempt to deliver

them via a store-carry-forward approach. We refer to this special buffer as *OppQueue*. If at any point in time, a route to the destination node of any of the packets stored in the OppQueue is established, they can be delivered directly via standard end-to-end routing. The switching between the two modes of communication can happen dynamically and transparently, without any special signalling. Figure 6.1 shows a basic example, to illustrate the concept.

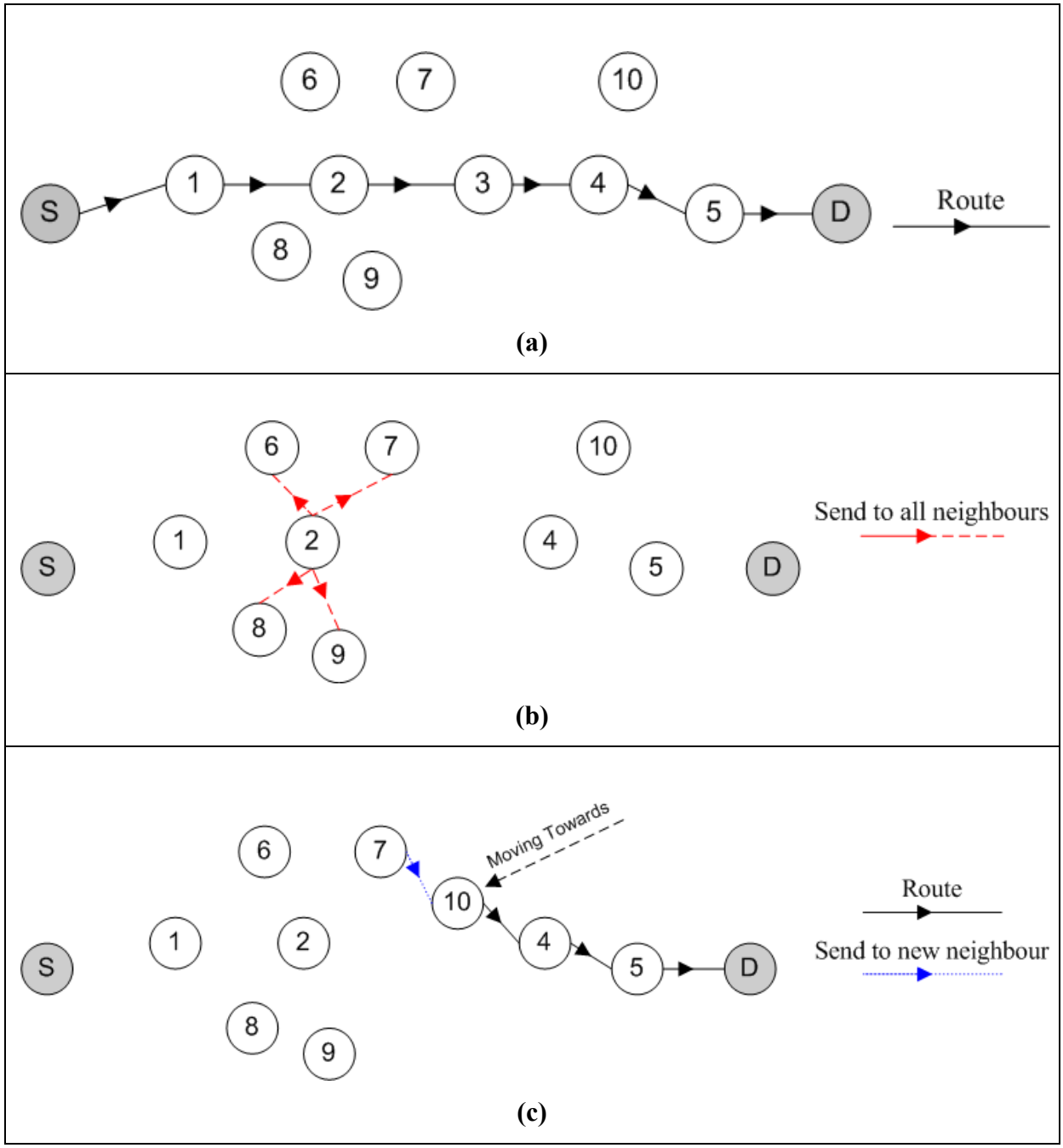



Figure 6.1: An example of packet routing in OLSR-OPP

In Figure 6.1(a), a source node S has a route to destination D via route S-1-2-3-4-5-D. In this kind of scenario, the network will operate in normal end-to-end routing mode, as supported by standard OLSR. We consider the scenario where the route from node S to D breaks, such as due to the disappearance/failure of node 3, as shown in Figure 6.1(b). The standard behaviour of OLSR in this case would be to simply drop any packets at node 2 destined to node D. In contrast, OLSR-OPP will store these packets in the OppQueue at node 2, in the hope of being able to deliver them later, either via store-carry-forward or via end-to-end routing, in case the route is re-established later on.

OLSR-OPP invokes the store-carry-forward routing for packets in the OppQueue at node 2 by sending them to its immediate neighbours. This is illustrated in Figure 6.1(b), where the packet (in its original format) is sent to nodes 6, 7, 8 and 9.

It is most likely that these nodes do not have a route to D, so they will also store the packet in their respective OppQueue, and further send them to their immediate neighbours, except for the node from which the packet was received from. In order to limit the overhead of this flooding based forwarding, a cap (called *copy_count*) is imposed on the number of neighbours to which the packet is forwarded.

In this fashion, the buffered packet will be disseminated further in the network. The packet can be delivered to the destination node D when any of the nodes that receive the packet have a route to D, either at the time when they receive the packet, or at any later point in time. Packets can also be delivered if the carrier node comes into direct contact with the destination node, as in traditional DTN routing. However, we can consider this simply as a special case of the above scenario, since coming into direct contact means having a route to the destination, even if it is only a one-hop route. An example of this is shown in Figure 6.1(c), where mobile node 10 moves into range of node 7. Due to standard OLSR Topology Control and HELLO messages, node 7 learns about the newly established route to D via node 10. Since node 7 has previously received a copy of the packet(s) destined to D, it can now deliver them via normal end-to-end routing. In that case, the buffered packets are then removed from the nodes OppQueue.

By doing so, OLSR-OPP can improve the packet delivery ratio by delivery packets

that would otherwise be dropped. In the following, we will provide further details of OLSR-OPP and its packet handling mechanism, in particular the two key events that trigger a special action in OLSR-OPP, the case when a packet is to be dropped due to the unavailability of a route, the encounter of a new neighbour, and the discovery of a new route.

### 6.2.3 OLSR-OPP Packet Handling

**Handling Packet Drops**

In Section 6.2.1, we discussed the two ways for detecting link failures in OLSR; that is, by periodic exchange of HELLO messages and by the link layer feedback mechanism. Irrespective of which approach is used, we need to handle the case where packets are to be dropped by OLSR, and extend the functionality of the protocol in that case. Figure 6.2 shows the basic behaviour of OLSR-OPP in the case where OLSR is to drop a packet due to the unavailability of a route. In case of a packet drop scenario, OLSR-OPP will store the packet in a special buffer or queue called Opportunistic Queue (*OppQueue*).

OLSR-OPP will also send the packet to the node's immediate neighbours (or a randomly chosen subset), as determined by the OLSR Neighbour Set. There are a maximum number of copies of each packet to be disseminated by a node, determined by the initial value of the *copy_count* parameter. The *copy_count* parameter is decremented whenever a copy of a packet is sent to an immediate neighbour of a node. The node stops forwarding packets when all neighbours have received a copy, or if the *copy_count* reaches 0. When the *copy_count* parameter reaches 0, the packet is no longer forwarded to neighbour nodes, but it remains in the *OppQueue*.

Additional information will be added to each packet when it is stored in the *OppQueue*: The current value of the *copy_count* parameter, and the *opp_TTL* value, which is time based Time-To-Live parameter in seconds, which indicates the local time that a packet can stay in the OppQueue of a node, before it is purged. A higher *copy_count* parameter achieves a higher packet delivery probability, but at a cost of higher dissemination overhead. We will evaluate this trade-off later in this chapter. The *opp_TTL* parameter allows configuring the protocol to cater for different amounts of packet delivery delay can be tolerated.

At the end of the packet drop handling scenario shown in Figure 6.2, the protocol returns to normal OLSR operation.

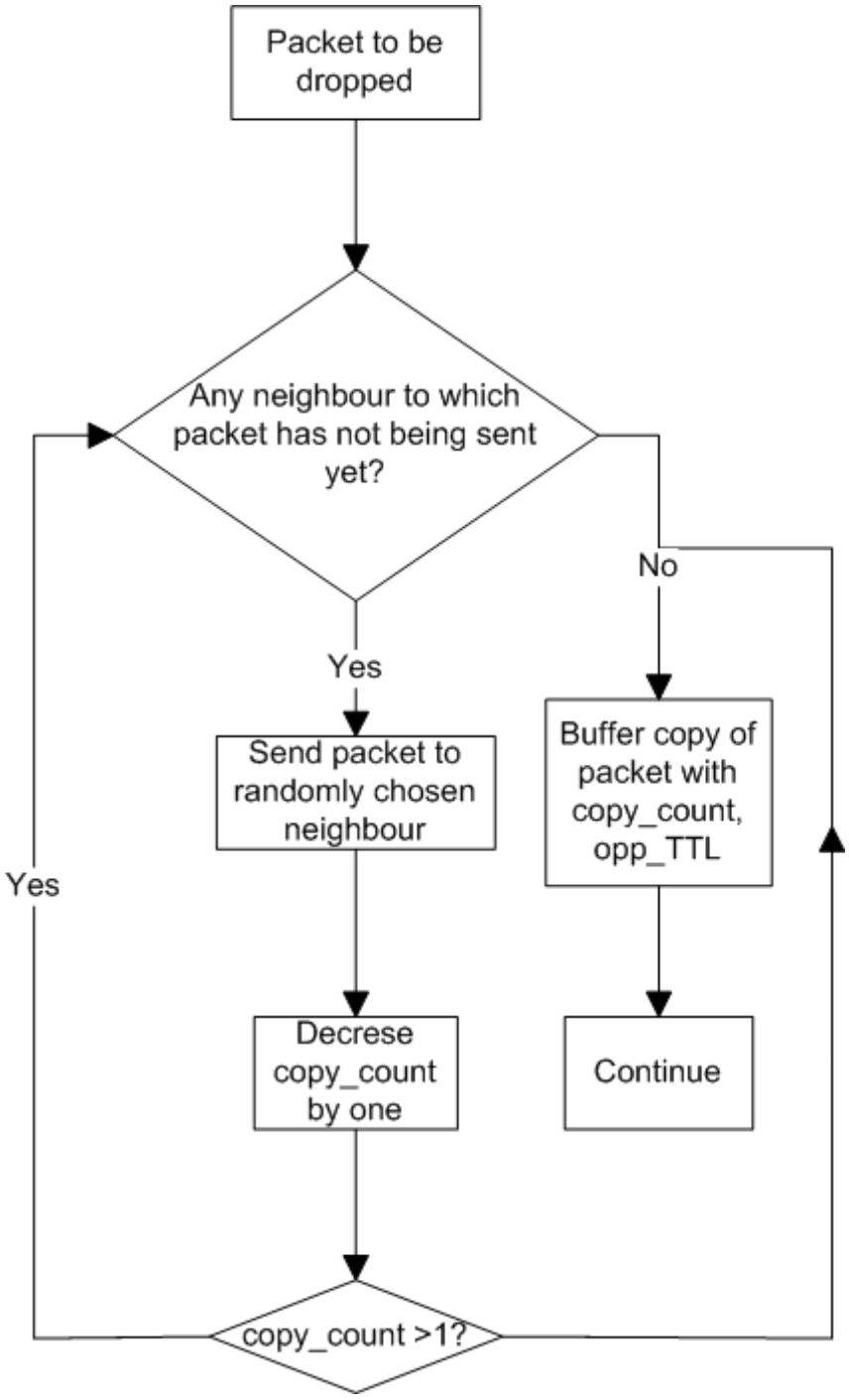



Figure 6.2: Handling Packet Drops

**Handling Detection of New Neighbour Node**

As a node moves around in the network, it might encounter other nodes through direct contact. As shown in Figure 6.3, whenever a node detects a new symmetrical link to another node, it will check for each packet in the OppQueue if the new neighbour is the packet's destination. If that is the case, OLSR-OPP will deliver the packet to the neighbour and remove the packet from the OppQueue.

If the new neighbour is not the destination, the protocol checks if the *copy_count* value is greater than 1, and if that is the case, forwards a copy to the neighbour in store-carry-forward mode.

The corresponding *copy_count* parameter is decremented by 1. If the *copy_count* parameter is not greater than 1, which means that the maximum number of packet copies has already been disseminated, the protocol will return to its normal operations.

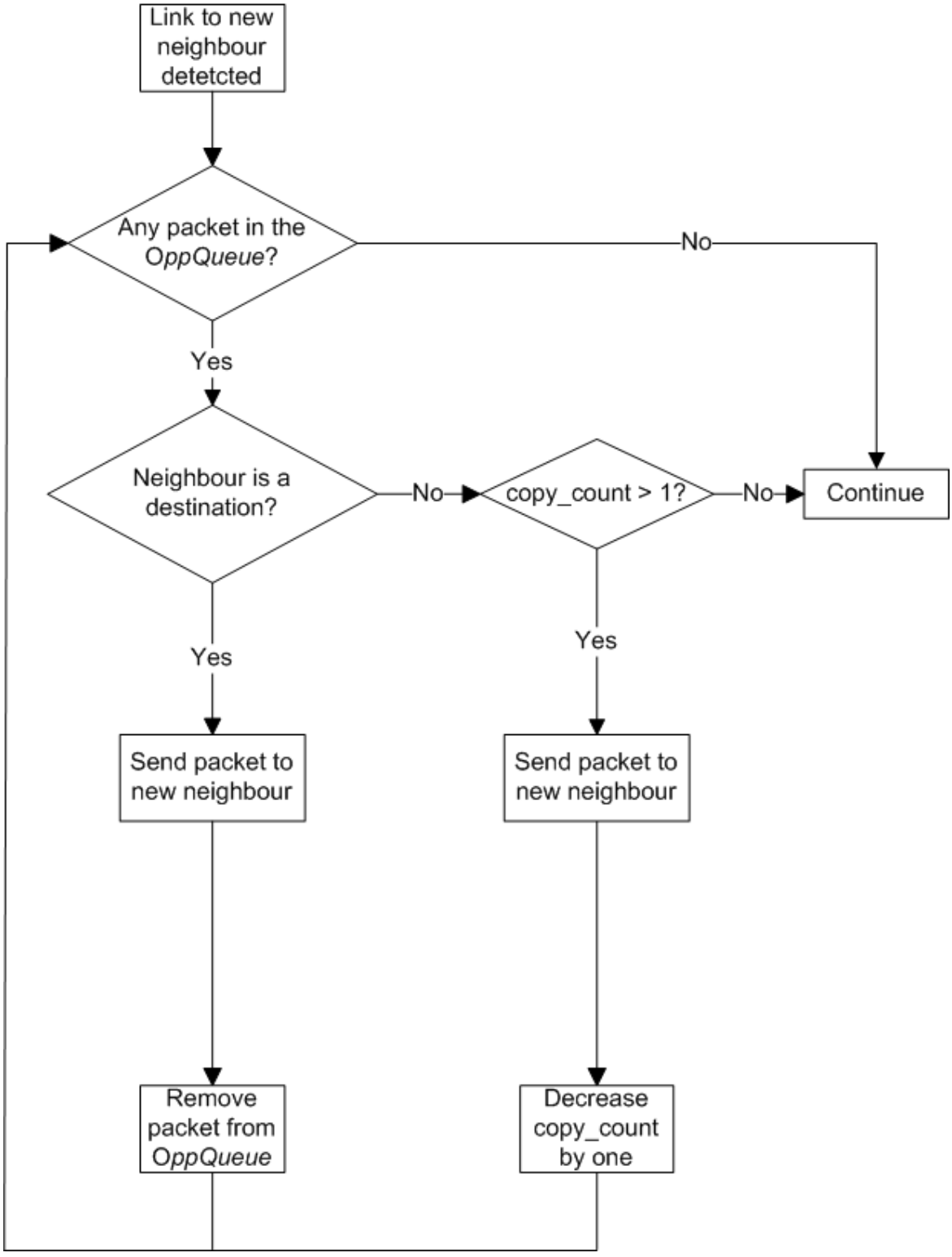



Figure 6.3: New Neighbour Node Encountered

**Handling Detection of New Route**

Due to the continuous exchange of Topology Control message, OLSR nodes learn about topology changes and discover new routes. In such an event, there is a chance that a new end-to-end route to a destination of a packet in the *OppQueue* is discovered. Therefore, anytime there is a change made to the routing table by OLSR, OLSR-OPP triggers this check, as shown in Figure 6.4. The protocol simply checks if the new routing information provides a route for the destination of any of the packets in the *OppQueue*, and if that is the case, it simply sends the packets towards the next hop of this route, and removes the packet from the *OppQueue*. While the availability of a route in the routing table generated by OLSR does not guarantee successful delivery of the packet, the probability is high enough to warrant the removal of the packet from the *OppQueue.* This is a reasonable trade off in terms of delivery probability and resource efficiency.

In this context, the ability of OLSR to continually learn about new routes is a big advantage over reactive protocols, where routes are only discovered as a result of route discovery triggered by a source node. This allows OLSR-OPP to automatically and transparently switch from store-carry-forward routing back to the much more efficient end-to-end routing approach. If a similar approach is to be applied to a reactive routing protocol such as AODV, additional overhead and complexity is required to discover new routes that can potentially be used for the delivery of packets stored in DTN mode [112].

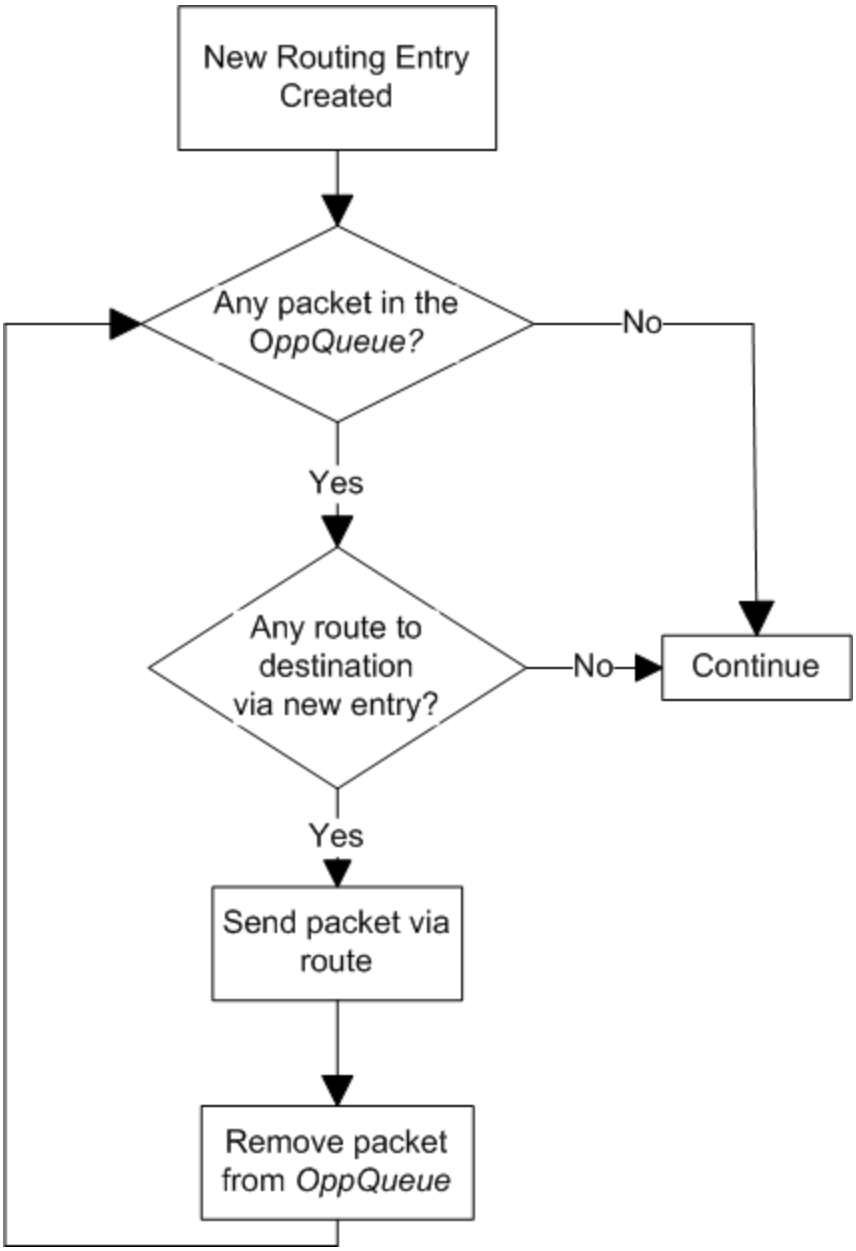



Figure 6.4: New Routing Entry

## 6.3 OLSR-OPP Implementation

In this section we will discuss a few relevant implementation details of the OLSR-OPP protocol in ns-2.As a basis, the OLSR implementation from the University of Murcia (UM-OLSR) was used [108], which is compliant with the OLSR RFC 3626 [3].

The buffer (OppQueue) in which packets in the store-carry-forward mode, is implemented as a linked list. In addition to the packets, additional meta-information is stored for each packet, in particular the *copy_count* and *opp_TTL* parameters.

These parameters are initialised when a packet is stored in the buffer. We will explore a range of values for the *copy_count* parameters later in this chapter. We use 400 seconds as the default value for the *opp_TTL* parameter. A process in OLSR-OPP regularly checks if for any of the packets in the OppQueue the *opp_TTL* has expired, and if that is the case, the packet is purged from the buffer. Packets are buffered in the OppQueue data structure in the event a link break is detected, in which case OLSR would normally drop the packet.

## 6.4 Basic Validation Test

In order to verify the correctness of our OLSR-OPP implementation, we have performed a set of validation tests, in a set of small scale simulation experiments.

### 6.4.1 Simulation Setup and Scenarios

The validation tests are conducted in ns-2 simulator (ns-2.34). Figure 6.5 shows the six nodes "diamond" topology that was used. Node **S** is the source node which generates the traffic. Node **R** is the receiver or sink of the traffic. None of the other nodes act as traffic sources or sinks, but simply as forwarders.

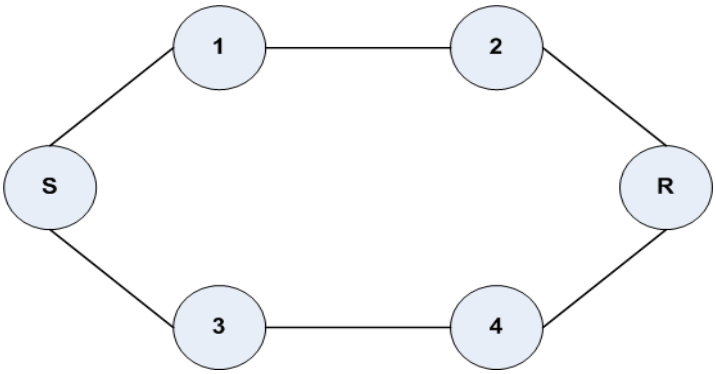



Figure 6.5: Validation Test Topology

The injected traffic consisted of a flow of CBR traffic, using UDP packets. In the simulation, the traffic flow is activated after 10 seconds of the 'warm-up time' in order to allow the OLSR proactive topology discovery mechanism to do its job and establish the network topology at each node.

After the node S stops sending, we wait for another 50 seconds before we terminate the experiment. This is to allow clearing out any packets in OppQueue at different nodes to be delivered to their destination. Since we were not interested packet loss due to buffer overflows, we chose a sufficiently large size of relevant buffers such as the OppQueue, to prevent tail drop packet loss. The relevant simulation parameters are listed in Table 6.1.

| | | |
|---|---|---|
| OLSR-OPP Parameters | copy_count | 10 |
| | opp_TTL | 400 seconds |
| Traffic Parameters | Traffic Type | CBR |
| | Traffic Start Time | 10s |
| | Traffic end time | 120s |
| | Data rate | 4 pkts/s |
| | Packet size | 512 bytes |
| Network Parameters | Transmission Range | 250m |
| | IFQ length | 1000pkts |
| | Simulation Time | 300 sec |
| | 802.11 MAC Tx Rate | 11 Mbps |
| | Propagation Model | Two Ray Ground |
| | RTS/CTS | Enabled |
| | Queue Type | Drop Tail |
| | Simulation Area | 1500 x 1500 $m^2$ |

Table 6.1: OLSR-OPP Simulation Parameters

The objective of these validation tests is to verify the basic operation of the OLSR-OPP protocol. We have defined 4 different set of test scenarios. Each scenario aims to test one specific aspect of OLSR-OPP.

*Case 1: Static case*

This scenario simply aims to establish that the extensions made to the OLSR protocol do not incur any additional overhead or any unanticipated packet loss. In a completely static case, it is expected that OLSR-OPP behaves exactly as OLSR. From our results, we see that this is the case. The PDR of OLSR-OPP is the same as for OLSR, i.e. 100%, as shown in Figure 6.6. We also observe that no packets are buffered in the OppQueue in any of the nodes, also as expected.

*Case 2: Route break and re-routing*

The aim of the second scenario is to test the ability of OLSR-OPP to handle short disruptions in connectivity. In this scenario, traffic is initially delivered via the S-1-2-R path. After 50 seconds, we move Node 2 out of transmission range of all the other nodes, which will create the route to be disrupted. However, there exists an alternative route via Nodes 3 and 4, and the protocol will use this alternative route, as soon as the link break is detected, which is via HELLO messaging in this implementation.

Looking at the traces, we see that only two packets are affected by the disruption. In the case of OLSR, both of these are dropped. In OLSR-OPP, we expect both packets to be buffered and delivered as soon as the route is re-established. However, we notice that only one of the two packets is delivered, and we still have one, somewhat unexpected packet loss. This is reflected in Table 6.2. After some investigation, we found that this is not an error, but that the packet was drop due to the routing loop detection mechanism. The mechanism checks if a packet is to be sent to a next-hop node from which it was received previously, and if that is the case, drops the packet.

*Case 3:Route break, no re-routing*

This scenario is similar to Case 2 except that the alternative route S-3-4-R is not available here, since we remove Node 3. Therefore, the route is disrupted, but without any route repair or any possibility to deliver packets in store-carry-forward fashion. We expect

OLSR-OPP to perform the same way as OLSR. While OLSR-OPP buffers packets that are dropped by OLSR, it has no opportunity to deliver them. This is confirmed in Figure 6.6 and Table 6.2, where we see that a total of 74 packets are buffered in the OppQueue in total, but none of them are delivered.

*Case 4:Route break and re-establishment after some time*

This scenario aims to verify the ability of OLSR-OPP to handle route disruption for an extended period of time, and to switch dynamically between end-to-end and store-carry-forward mode. This scenario is a variation of Case 3. The difference here is that after an absence of 50 seconds, Node 3 is returned to its position between Nodes S and 4, to repair the route. Here we see the typical scenario where OLSR-OPP can deliver a benefit. Figure 6.6 shows that it achieves a PDR of 100%, in contrast to the corresponding value of OLSR of only just over 80%. In Table 6.2, we see that all of the packets buffered in OppQueue are eventually delivered to Node D, as we expected.

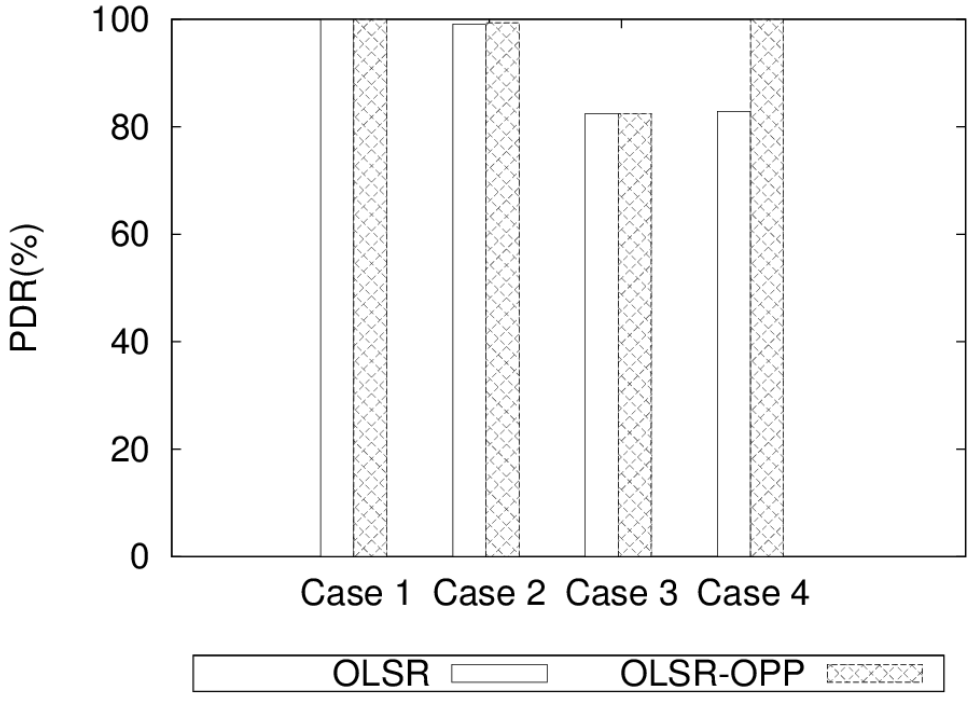



Figure 6.6: PDR values for Validation Test Scenarios

| Scenario | Number of packets buffered in OppQueue | Number of packets lost |
|---|---|---|
| Case 1 | 0 | 0 |
| Case 2 | 1 | 1 |
| Case 3 | 74 | 74 |
| Case 4 | 74 | 0 |

Table 6.2: OLSR-OPP packet handling statistics

After having confirmed the basic behaviour of the OLSR-OPP implementation, we now move on to do a more extensive performance evaluation in the following section.

## 6.5 Performance Evaluation of OLSR-OPP

The aim in this section is to evaluate the performance of OLSR-OPP in regards to its packet delivery capability, across the entire spectrum of network connectivity levels, from highly disconnected, to highly connected networks. The performance of OLSR-OPP is compared to OLSR, and for some scenarios with the Spray-and-Wait DTN protocol.

### 6.5.1 Experiment Scenarios

Traditionally, wireless multi-hop networks are considered either as largely connected, the WMN case, or largely disconnected; the DTN case. In contrast, we aim to systematically study the performance of OLSR-OPP in the entire range of connectivity scenarios, including all the in-between cases. For this, we need a parameter that describes the level of connectivity in the network. While the node degree parameter of a topology, which indicates the average number of neighbours per node, provides some concept of network denseness, it does not give any direct information about the availability of routes between node pairs.

For our purpose, we use the Partitioning Degree (PD) parameter, discussed in [104]. This simple parameter is defined as the ratio of the number of node pairs that are NOT connected via a route, to the total number of node pairs in the given network topology. It can also be interpreted as the probability of two randomly selected nodes not having a route. One nice property of the PD parameter is that it normalises the degree of connectivity, or dis-connectivity rather, in the range of 0 to 1. (This is also in contrast to other metrics, such as node degree.) A PD value of 0 means a completely connected network topology, with a route between any source destination node pair. A PD value of 1 means the opposite, i.e. a network where no node pair has a route.

We use the BonnMotion [104] network topology and mobility pattern generator to generate all our topology scenarios. Since our aim is evaluate the performance of OLSR-OPP across the whole range of network connectivity levels, we therefore generate network topology scenarios with a Partitioning Degree from 0 to 1. We differentiate between 3 levels of connectivity, with the following corresponding PD values: PD low [0-0.33], PD medium ]0.33-0.66], and PD ]0.66-1]. These three categories correspond to high connectivity, medium connectivity, and low connectivity.

We initially generated 2,000 random network scenarios, all for 50 nodes, using the Random Waypoint mobility model. We randomly chose the (rectangular) size of the simulation area, by randomly choosing the length and the width. In this way, we can generate scenarios with varying density, connectivity, and therefore varying PD values. We then randomly chose 100 scenarios from each of the three categories (low, medium and high PD value) as the basis of our simulations. Figure 6.7 shows the cumulative density function (CDF) of the Partitioning Degree. Given that uniform distribution has a linear CDF, and that the CDF in Figure 6.7 can be approximated by a straight line, we see that our 300 scenarios have a fairly uniform PD distribution, in the range between 0 and 1, covering all levels of connectivity.

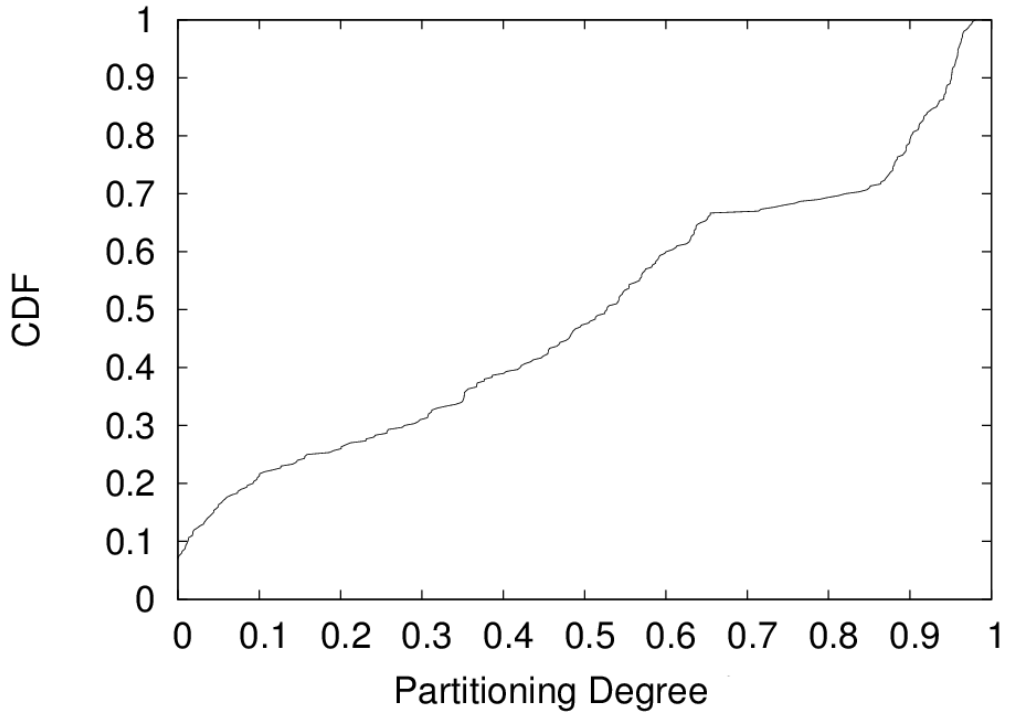



Figure 6.7: CDF of Partitioning Degree

For all our simulations, we used 50 mobile nodes and 10 concurrent traffic flows between uniformly randomly chosen source-destination pairs. Individual traffic flows have duration of 10 seconds. A summary of key simulation parameters is shown in Table 6.3.

| | | |
|---|---|---|
| OLSR-OPP Parameters | copy_count | 10 |
| | opp_TTL | 400s |
| Traffic Parameters | Traffic Type | CBR |
| | Concurrent traffic flows | 10 |
| | Traffic flow duration | 10 s |
| | Data rate | 4 pkts/s |
| | Packet size | 512 bytes |
| Network Parameters | Transmission Range | 250m |
| | Number of Nodes | 50 |
| | IFQ length | 50 pkts |
| | Simulation Time | 500 s |
| | 802.11 MAC Tx Rate | 11 Mbps |
| | Propagation Model | Two Ray Ground |
| | RTS/CTS | Enabled |
| | Queue Type | Drop Tail |
| | Mobility Model | Random Waypoint |
| | Simulation Time | 500 s |

Table 6.3: OLSR-OPP Simulation Setting

### 6.5.2 OLSR-OPP Packet Delivery Performance Evaluation

We aim to evaluate the performance of OLSR-OPP in terms of Packet Delivery Ratio (PDR) for the range of topologies discussed in the previous section. For each of the 300 considered scenarios with varying Partitioning Degrees (PD), we get a PDR value from the ns-2 simulation. In Figure 6.8, we plot these 300 data points, i.e. we plot the PDR value versus the corresponding PD value. For comparison, we also plot the corresponding PDR values of the OLSR protocol. To better visualise the difference in performance, we also perform curve fitting for these data points using a second degree polynomial function, for each of the data point sets.

We observe that OLSR-OPP achieves a significant (up to 50%) performance improvement over OLSR-OPP in a very wide range of the PD parameter, with the only

exception of the very extreme ends of the PD range. It is clear that no major improvement can be achieved in the extreme corner case of when PD is close to 0, since in that case of (almost) perfect connectivity, OLSR already performs at close to 100% PDR. In the other extreme case of PD close to 1, the connectivity is extremely limited with a close to 0 probability of a route between node pairs. In that case, no problem can be expected to perform well.

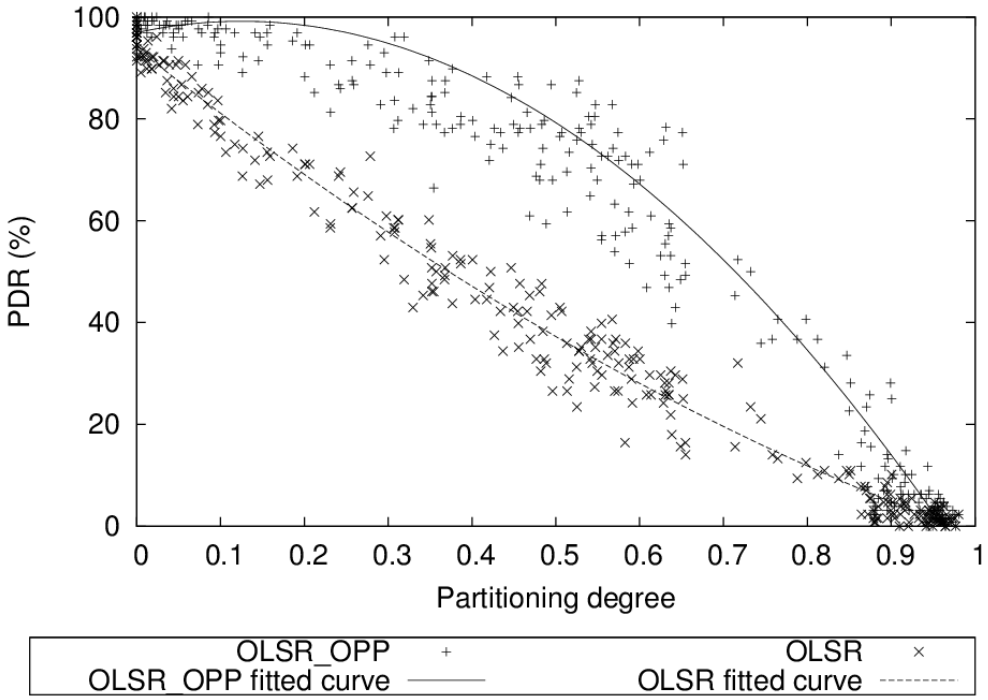



Figure 6.8: PDR *Performance of OLSR and OLSR-OPP*

Figure 6.9 shows the Cumulative Distribution Function (CDF) of the PDR gain OLSR-OPP achieves over OLSR, for our three PD scenarios: low, medium and high. We see that for the medium PD range case, the improvement is most significant, followed by the low PD range, and the high PD range. For the medium PD case, we see that the minimum PDR gain is 20%, and that in 50% of the scenarios, the PDR improves by more than 40%. This information is also shown in Figure 6.10 in a different way. The figure shows the average performance gain of OLSR-OPP over OLSR in terms of PDR, for each of the 3 categories. The figure confirms what we have already observed in Figure 6.8, i.e. the highest performance gain is achieved for topologies with a medium level PD value, i.e. with moderately intermittent connectivity.

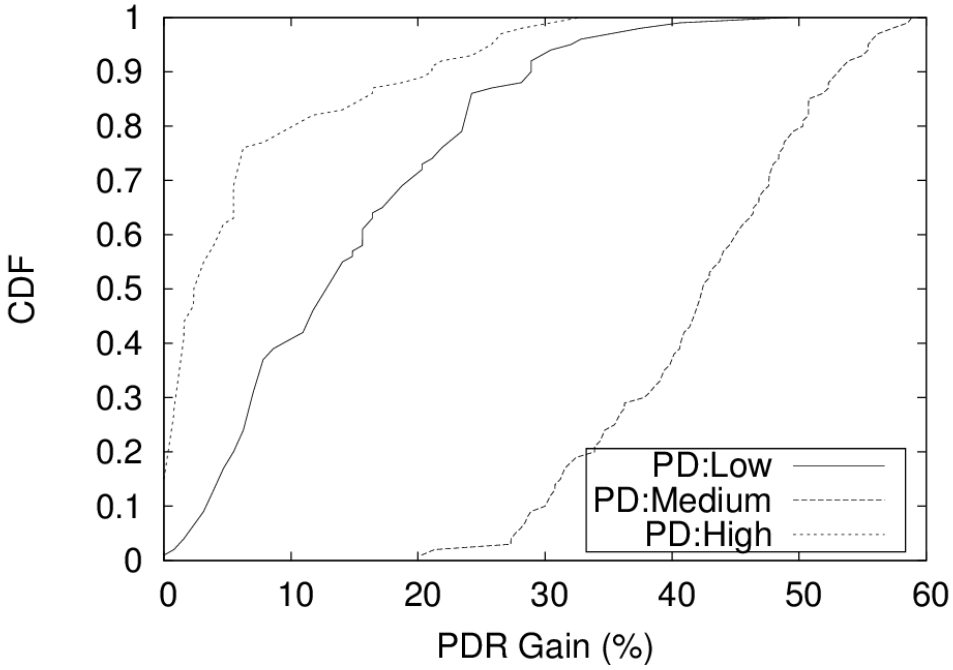



Figure 6.9: CDF of PDR gain of OLSR-OPP over OLSR-OPP

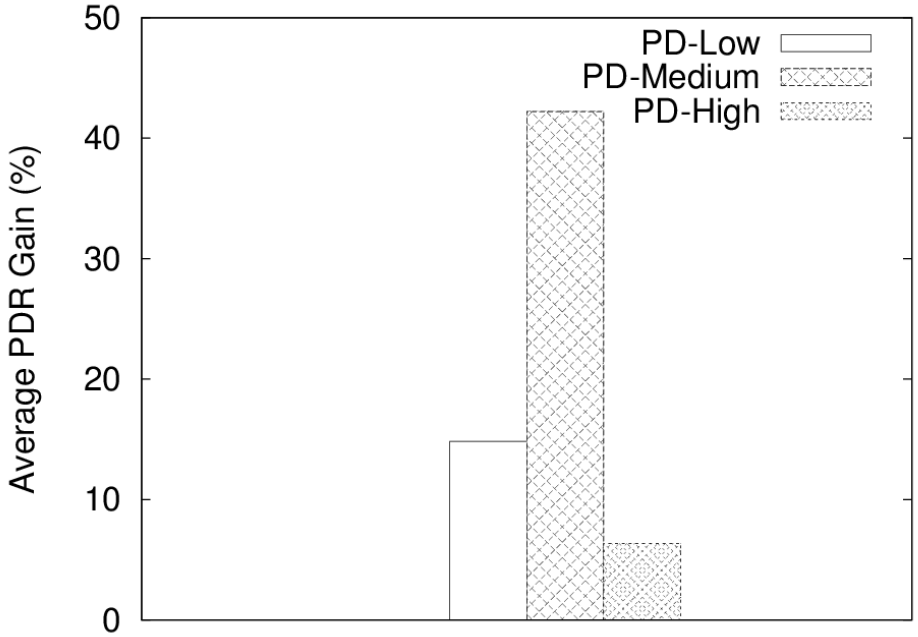



Figure 6.10: Average PDR gain for each PD range

### 6.5.3 End-to-end Delay

OLSR-OPP is able to achieve significant improvements in terms of PDR over OLSR by opportunistically switching between story-carry-forward routing and end-to-end routing, depending on the available connectivity. It is clear that packets experience a much higher delay in store-carry-forward routing, due to the extended buffering time. As in DTN routing, the trade off is an increased PDR, at a cost of an increased end-to-end delay.

Figure 6.11 shows the averaged end-to-end delay of packets for the 300 different scenarios for both OLSR-OPP and OLSR. As expected, the delay for OLSR is consistently close to 0. Also as expected, we see that OLSR-OPP has increased end-to-end delay, mostly for scenarios with a high Partitioning Degree. In these cases, we observe end-to-end delay values above 60 seconds. For highly connected networks with a low PD value, the store-carry-forward mode is rarely used, and therefore the end-to-end delay of OLSR-OPP is in a more similar range to OLSR.

We compute the average end-to-end delay for the 100 scenarios in each of our different PD categories and the results are shown in Table 6.4. We see that OLSR has a consistently low delay, irrespective of the level of connectivity. Again, this is as expected, since OLSR operates in a binary fashion, if there is an end-to-end route, packets are delivered, if not, they are dropped. For OLSR-OPP, the store-carry-forward delay increases for more disconnected networks, i.e. for the medium and high PD ranges.

It is important to note that a higher end-to-end delay is not a reflection on the quality of the OLSR-OPP, it simply follows from the fact that the protocol can trade off better PDR for increased delay.

The maximum tolerable end-to-end delay obviously depends on the type of application, and some applications are more delay tolerant than others. In OLSR-OPP, the end-to-end delay can be controlled (indirectly) via the *opp_TTL* parameter, which determines the maximum time a packet is buffered by an individual node. In our simulations, we set the *opp_TTL* parameters to a high value of 400 seconds, which aims to maximise PDR, at the cost of a higher delay.

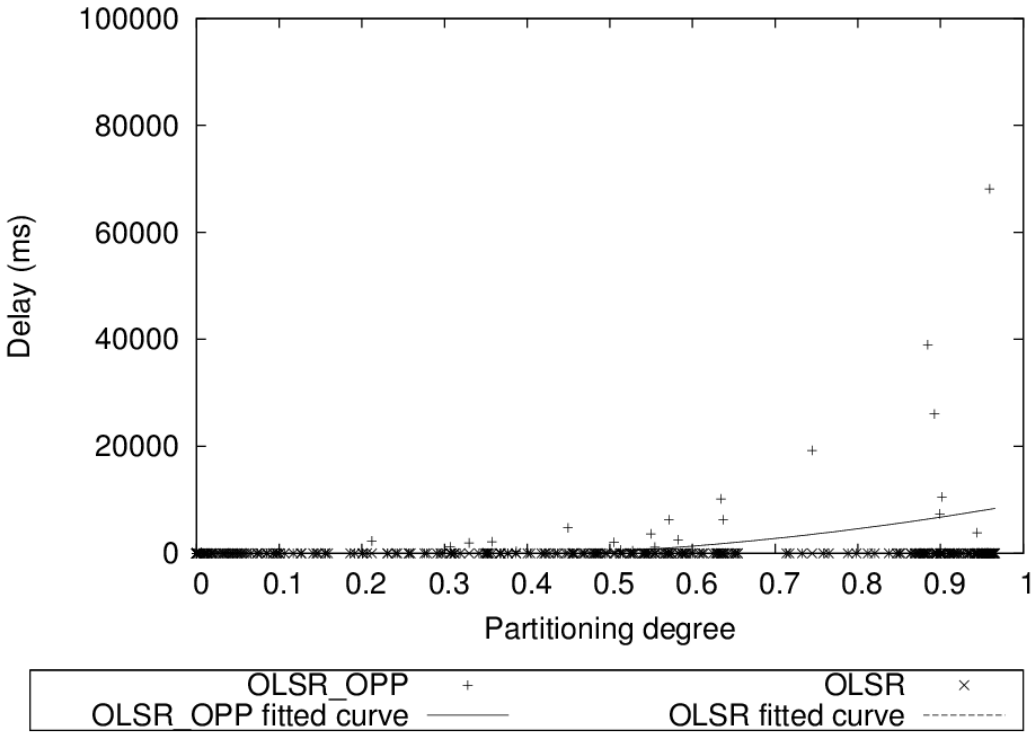



Figure 6.11: End-to-end Delay versus PD

| PD Range | OLSR | OLSR-OPP |
|---|---|---|
| PD-Low | 5.5 | 60.1 |
| PD-Medium | 6.7 | 412.4 |
| PD-High | 2.7 | 6968.4 |

Table 6.4: Average end-to-end Delay Comparison (in milliseconds)

### 6.5.4 Trading off PDR and Forwarding Overhead

In OLSR-OPP, a node will try to forward a buffered in the *OPP_Queue* for a maximum number of times, determined by the *copy_count* parameter. It is clear that the higher the number of copies disseminated in the networks, the higher the probability of packet delivery. In the extreme case, flooding is very effective in that regard, but it also has the highest overhead. OLSR-OPP limits the overhead of flooding, while still maintaining a very high PDR. In this section, we explore how changing the OLSR-OPP *copy_count* parameter affects both the PDR, as well as the forwarding overhead.

We define the *forwarding overhead* $O_f$ as the total number of one-hop packet transmissions, divided by the total number of successfully delivered packets. For this, we only consider packets sent in store-carry-forward, and exclude packets that are delivered by normal end-to-end routing only.

$$O_f = \frac{Number\ of\ Packet\ Transmissions}{Number\ of\ Packets\ Delivered} \qquad (6)$$

Figure 6.12 shows the performance of OLSR-OPP for different *copy_count* values (2, 4, 6, 8, 10), across the entire PD range. For example, the label *OLSR-OPP-2* refers to the OLSR-OPP with *copy_count=2, etc.* We again fit a second order polynomial for each of the data sets, to better visualise the results. As expected, we see that a higher *copy_count* value results in higher PDR. However, the increase is relatively small.

This is also shown in Figure 6.13, which shows the average PDR gain of OLSR-OPP over OLSR, as a function of the *copy_count* value. (The average of the gain is taken over all 300 topology scenarios.) We see that for an increase of the *copy_count* parameter from 2 to 10, the increment in PDR gain is around 4%. The biggest gain is from *copy_count* of 2 to 4. Beyond that, the marginal increase in PDR is very minimal.

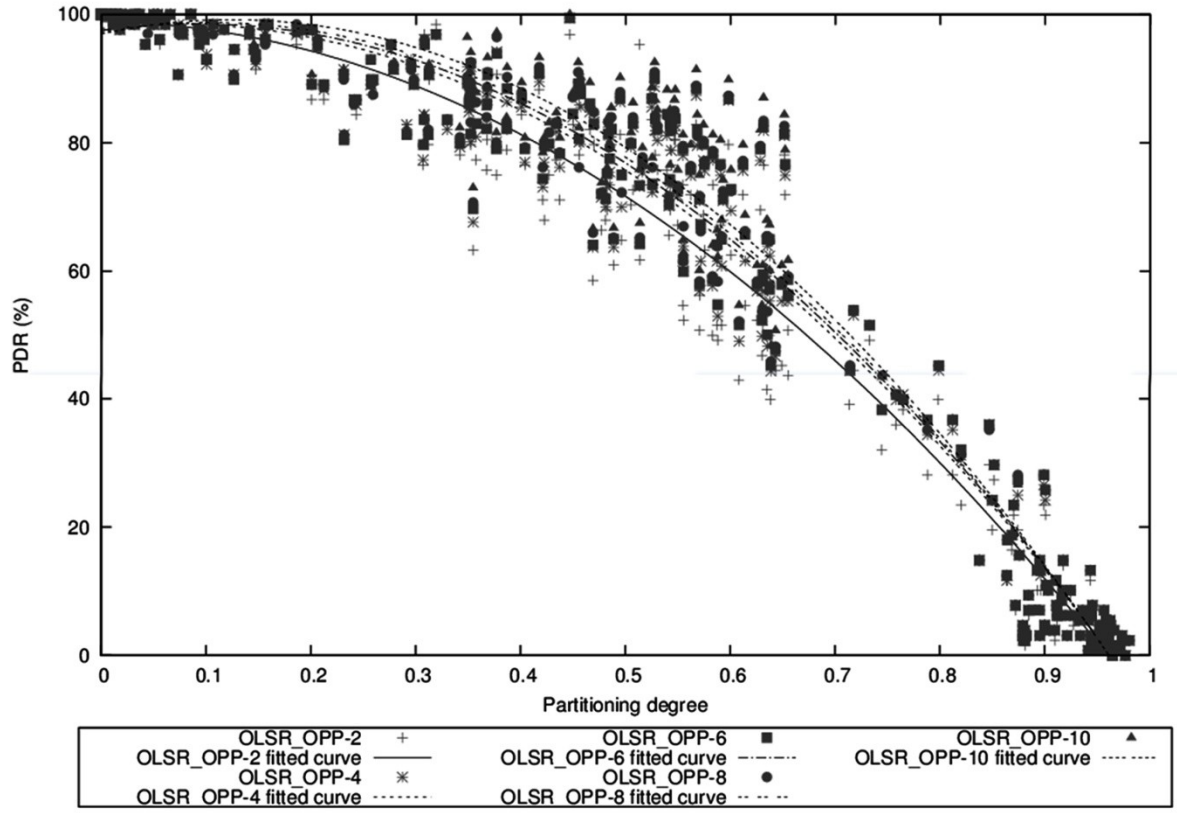



Figure 6.12: PDR vs. PD for different values of *copy_count*

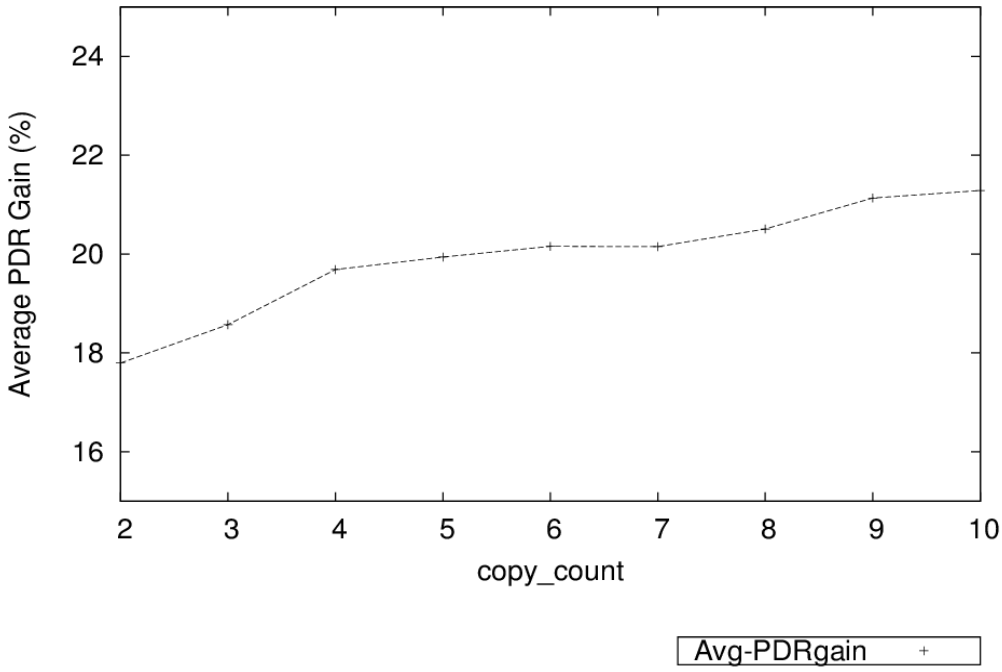



Figure 6.13: Average PDR gain vs. *copy_count*

While a *copy_count* value of 10 results in the best PDR performance of all the considered cases, it also has the highest overhead. Figure 6.14 shows the forwarding overhead $O_f$ across all 300 scenarios, for OLSR-OPP with the *copy_count* parameter ranging from 2 to 10. As expected, a higher *copy_count* value results in a larger number of packet transmissions and therefore a higher overhead.

Figure 6.15 shows the average forwarding overhead (averaged over all 300 scenarios), versus different *copy_count* parameter values. While there is a sharp increase in overhead for low *copy_count* values, the increase tapers off for higher values. This is due to the fact that the network is close to 'saturation', where all node that can receive a packet, would have received a copy, and further duplicates are no longer forwarded.

Choosing the optimal value of the *copy_count* parameter, and the corresponding trade-off between PDR and overhead, depends on the application and network deployment requirements. In some cases, it might be desired to achieve a slightly higher PDR, at a significant cost in terms of additional overhead.

From looking at the PDR gain (Figures 6.12 and 6.13) and the forwarding overhead (Figures 6.14 and 6.15) for different choices of the *copy_count* parameter, we see that a low value of 2-4 looks like a reasonable choice, combining good performance, with limited overhead. Any further increase in PDR will come at a significantly higher overhead cost.

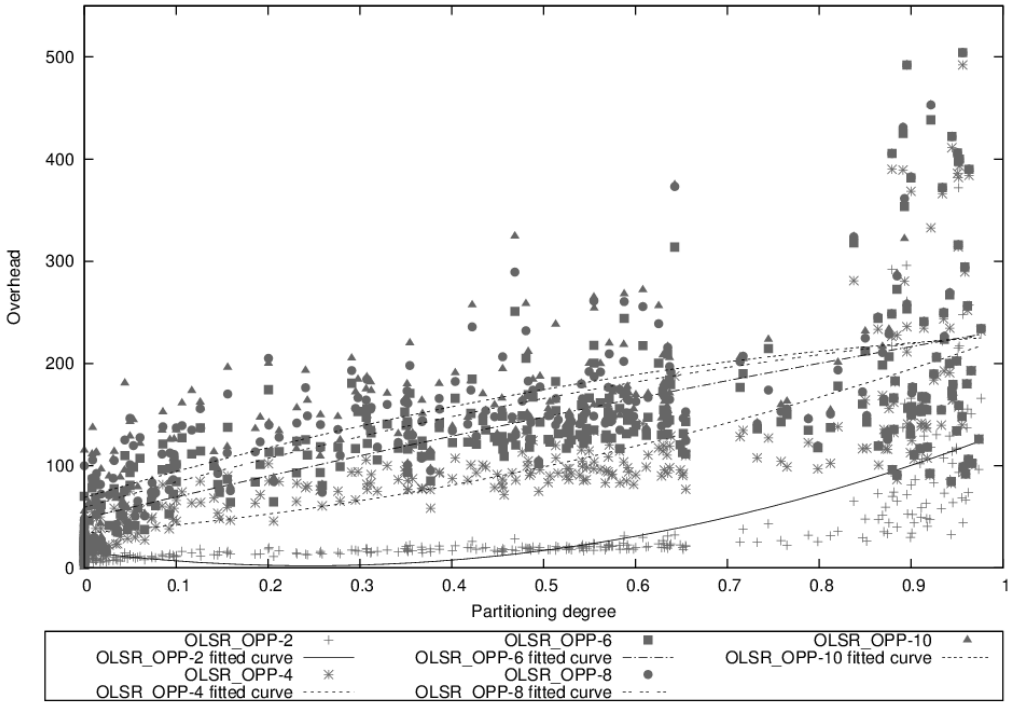



Figure 6.14: Overhead vs. PD for different number of *copy_count*

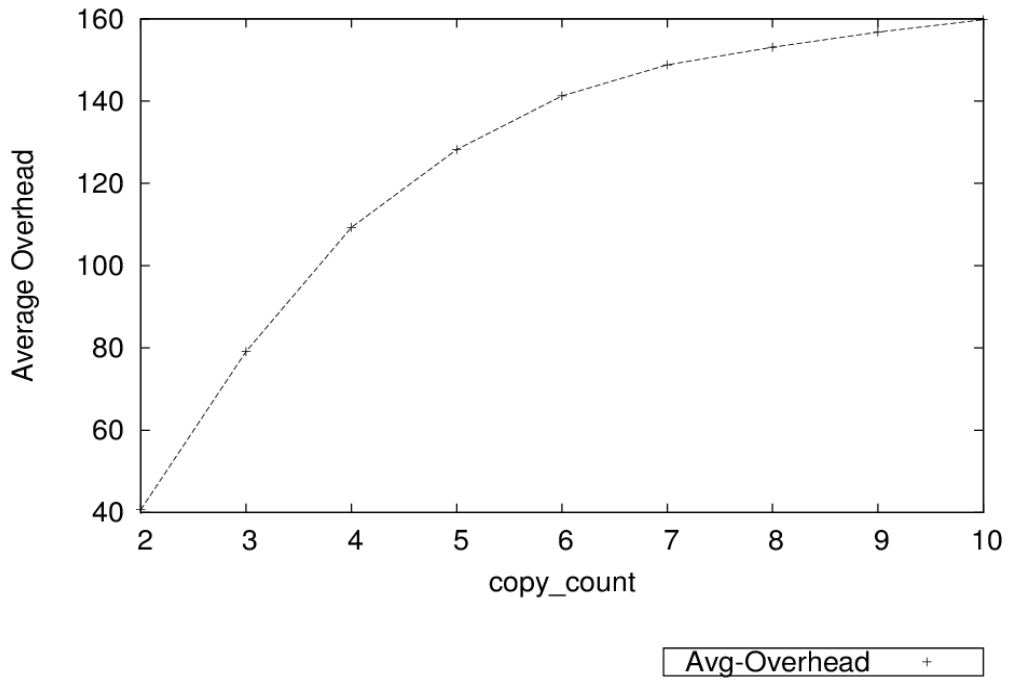



Figure 6.15: Average Overhead

### 6.5.5 Comparison of OLSR-OPP with Spray-and-Wait (SAW)

In the previous subsection, we have studied the PDR performance of OLSR-OPP and compared it to OLSR, a traditional WMN routing protocol which relies on end-to-end routing. Simulation results showed a significant performance improvement of OLSR-OPP compared to basic OLSR. In this section, we compare the performance of OLSR-OPP with Spray-and-Wait (SAW) [80], one of the key DTN routing protocols, which was briefly discussed in Chapter 3. The basic idea is simple. SAW has two phases, a spray phase and a wait phase. In the spray phase, for every message originating at a source node, a fixed number (L) of message copies are spread in the network to L different "relay nodes". If the message is not delivered in the spray phase, the wait phase simply waits for nodes to directly encounter the destination node, where it is delivered via direct transmission.

For our simulations, we use the SAW protocol implementation developed by Linkoping University [113]. In this version, Binary Spray-and-Wait is used as the transmission scheme [80]. In this scheme, a node that has n (n>1) copies of a message will pass on half of these copies to a new node that is encountered, and keeps the other half. This continues, until the node only has a single copy, which is the beginning of the wait phase.

Each SAW node will regularly send a beacon signal (which is similar to HELLO message) to determine when a new node is in the neighbourhood. We will call this node as a *beacon node*. If the beacon messaged is received by a node that carries a message, i.e. the *carrier node* or *querying node*, it will send a query message to the beacon node for the purpose of connection establishment, as illustrated in Figure 6.16. The beacon node will reply back with the response message to the carrier node. Afterwards follows the exchange of messages between the nodes.

The SAW protocol has a HELLO interval parameter, which determines the frequency in which HELLO messages or beacons are exchanged. The parameter is defined as a [min/max] range, from which the actual interval is chosen uniformly randomly. We use a range of [0.75/1.25], which corresponds to an average HELLO or beacon interval of 1 second, which corresponds to the HELLO interval of 1 second used our OLSR-OPP implementation.

The entire signalling mechanism of the SAW protocol, as shown in Figure 6.16, is relatively complex and resource intensive, in contrast to the minimal approach of OLSR-OPP.

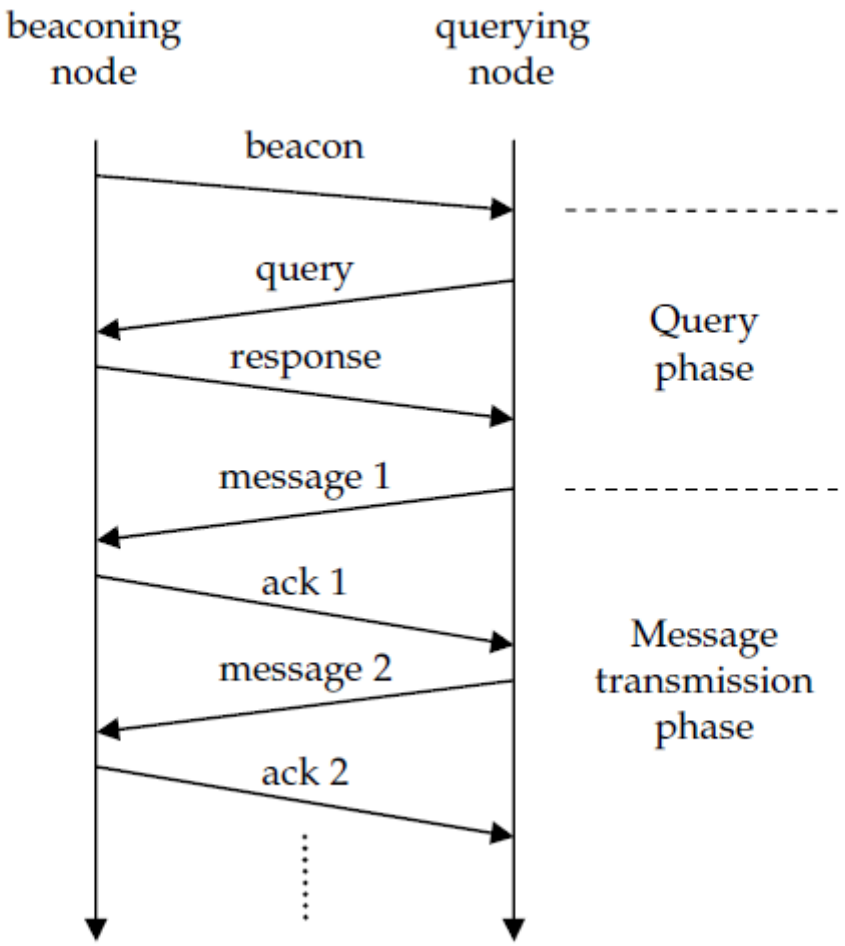



Figure 6.16: Packet Exchange Process in SAW [114]

For the comparison with SAW, we used the same 300 topology and mobility patterns for our ns-2 simulation, as used in the previous section.

The results of our simulations are shown in Figure 6.17, which shows the Packet Delivery Ratio for both OLSR-OPP and SAW. We also included the results for OLSR for comparison. From Figure 6.17, we see that both OLSR-OPP and OLSR outperform the SAW protocol for all but the most disconnected network scenarios, i.e. PD values of close to 1.

The relative poor performance of SAW across a wide PD range can be explained via the relatively involved connection establishment and message exchange process shown in Figure 6.16, where a relatively large number of message need to be successfully exchanged over wireless links with very intermittent connectivity. In contrast, OLSR-OPP has a very lightweight approach, where packets are simply exchanged between nodes, without any connection establishment or other required signalling.

Overall, we observe that OLSR-OPP significantly outperforms both the native WMN routing protocol OLSR, as well as the native DTN protocol Spray-and-Wait.

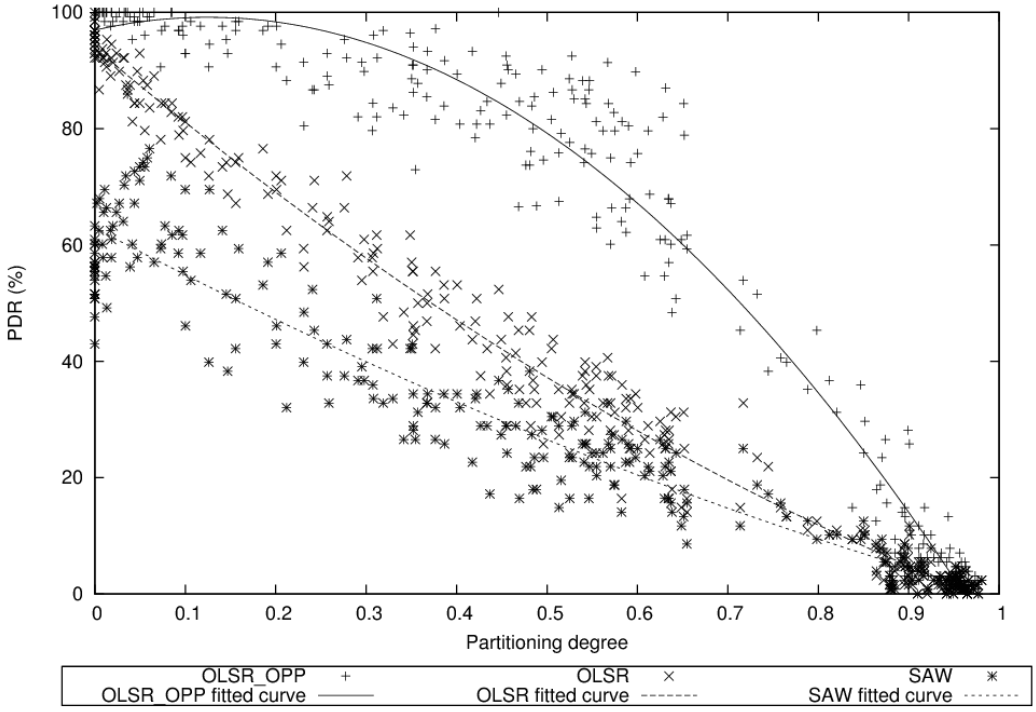



Figure 6.17: PDR vs. PD, for OLSR-OPP, OLSR and SAW

## 6.6 Summary

In this chapter, we have introduced OLSR-OPP, a simple hybrid protocol, which combines end-to-end routing with the store-carry-forward approach of DTN protocols in a very light weight manner. OLSR-OPP chooses the routing and forwarding mode opportunistically and transparently, based on the available network connectivity. This is one

of the main differences to related works such as of [97], where the forwarding mode change needs to be initiated by the source node, for the entire flow.

OLSR-OPP chooses the right forwarding mode opportunistically, and transparently, with zero overhead cost in terms of extra signalling. We verified the basic validity of our ns-2 implementation of OLSR-OPP, and performed extensive performance evaluations, across a systematically selected range of topologies and mobility scenarios, ranging from very high to very minimal network connectivity. We further explored the trade off between protocol performance and overhead via the tuning of the *copy_count* OLSR-OPP protocol parameter. Our performance comparisons of OLSR-OPP with OLSR and Spray-and-Wait (SAW) have shown a significant improvement in the Packet Delivery Ratio over both OLSR and SAW, across a wide range of topology scenarios, which is promising for future works in this direction.

# Chapter 7 CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

In this thesis, we have addressed the problem of routing in wireless multi-hop networks. Given the extremely broad range of deployment scenarios and related network characteristics of these networks, e.g. in terms of network connectivity, node mobility, traffic load and pattern, etc., the aim was to investigate how protocols can be tailored and adapted to such varying scenarios, with the goal of improving network performance.

Towards this goal, by means of extensive simulation experiments, we have investigated how different protocol mechanisms such as route repair and different routing protocol parameters, such as the HELLO interval, has an impact the performance of wireless multi-hop networks, for different traffic load and mobility scenarios.

We further investigated the performance of key Wireless Mesh Network routing protocols such as AODV, DYMO, OLSR and HWMP, in a wide range of scenarios, and analysed the reasons for packet loss. We found that the performance of standard WMN routing protocols decreases rapidly in networks with reduced levels of connectivity, i.e. in sparser network topologies.

This problem was addressed in this thesis by the introduction of OLSR-OPP, a new hybrid routing protocol which extends the traditional end-to-end routing approach of WMN protocols with the store-carry-forward mechanism of Delay Tolerant or Opportunistic networks. The extension is very light-weight, and OLSR-OPP is backwards compatible with OLSR. It allows to opportunistically and transparent switching between forwarding mechanisms, adapted to the particular level of connectivity that is available in the network.

Using network simulation, the performance of OLSR-OPP has been extensively evaluated across a large number of topologies and mobility scenarios. We used 300 scenarios with a (almost) uniform distribution of network connectivity, as expressed in terms of the Partitioning Degree parameter. This allowed a systematic evaluation of the protocol in a wide

range of networks, from highly connected to highly disconnected, with all the intermediary levels.

Our results have shown that OLSR-OPP manages to significantly improve the performance of OLSR in terms of Packet Delivery Ratio (PDR), across the entire range of topologies. The same qualitative result was achieved in a comparison with the Spray-and-Wait DTN routing protocol.

We also evaluated the trade-off in terms of forwarding overhead versus PDR, as controlled via the *copy_count* protocol parameter, and have seen that for a low value of the parameter, most of the performance improvements can be gained, with a relatively limited network overhead.

A store-carry-forward mechanism such as used in OLSR-OPP trades off increased PDR for increased end-to-end delay. We evaluated and quantified the increased delay experienced by packets.

In conclusion, the OLSR-OPP protocol is practical, since it provides a simple, light-weight, backwards-compatible and low overhead extension to OLSR, and manages to significantly increase the packet delivery performance across a wide range of networks with different levels of connectivity, in particular for networks with a Partitioning Degree between 0.1 and 0.8.

Directions for future work include more extensive evaluations of OLSR-OPP, in particular using test-bed experiments. Unfortunately, that was not possible in the context of this thesis, due to resource constraints.

It would also be interesting to investigate the optimal choice of the *opp_TTL* protocol parameter, depending on the level of delay tolerance of the application or network.

While this thesis has made steps towards exploring more adaptive and tailored routing protocols for wireless multi-hop networks, and the presented results are promising, a lot

remains to be done towards the goal of having protocols that are truly adaptive and perform optimally across the wide range of scenarios in which wireless multi-hop networks are deployed.

# REFERENCES


[1] I.F. Akyildiz and X. Wang, "A Survey on Wireless Mesh Networks," *IEEE Communications Magazine*, vol 43(9), pp. 23-30, 2005.

[2] C.E Perkins, E. Belding-Royer and S. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing" *RFC3561*, 2003.

[3] T. Clausen, and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," *RFC 3626,* 2003

[4] C. Richard, C. E. Perkins, and C. Westphal, "Defining an Optimal Active Route Timeout for the AODV Routing Protocol," *IEEE SECON*, 2005.

[5] W. Al-Mandari, K. Gyoda, and N. Nakajima, "Ad-Hoc On demand Distance Vector (AODV) Performance enhancement with Active Route Time-Out Parameter," *WSEAS Transactions on Communications*, vol. 7, pp. 912-921, 2008.

[6] C. Gomez, et al., "Evaluating performance of real ad-hoc networks using AODV with hello message mechanism for maintaining local connectivity," *IEEE PIMRC*, 2005.

[7] C. Gomez, D. Garcia, and J. Paradells, "An OLSR parameter based study of the performance of real ad-hoc network environments," *11th European Wireless Conference,* 2005.

[8] C. Gomez, et al., "Improving performance of a real ad-hoc network by tuning OLSR parameters", *IEEE ISCC*, 2005.

[9] Y. C. Huang, S. Bhatti, and D. Parker, "Tuning OLSR", *IEEE PIMRC*, 2006

[10] S. Demers and L. Kant, "MANETs: Performance Analysis and Management," *IEEE MILCOM*, 2006.

[11] S. Suhaimi, S. R. Azzuhri, and K. D. Wong, "Enhancing the 'Willingness' on the OLSR Protocol to Optimize the Usage of Power Battery Power Sources Left", *International Journal of Engineering (IJE)*, 2008.

[12] L. Chao and H. Aiqun, "Reducing the Message Overhead of AODV by using Link Availability Prediction," *3rd International Conference on Mobile Ad-Hoc and Sensor Networks*, 2007.

[13] L. Qin and T. Kunz, "Adaptive MANET Routing: A Case Study," *ADHOC-NOW*, 2008.

[14] V. Ramasubramanian, Z. J. Haas, and E. Gün Sirer, "SHARP: a hybrid adaptive routing protocol for mobile ad hoc networks," *ACM MOBIHOC*, 2003.

[15] Laura Sanders, "Slime Mold Grows Network Just Like Tokyo Rail System" http://www.wired.com/wiredscience/2010/01/slime-mold-grows-network-just-like-tokyo-rail-system/, Oct. 1 2010 [Nov 1 2012].

[16] D. Johnson, et al. "Building a Rural Wireless Mesh Network: A do-it-yourself guide to planning and building a Freifunk based mesh network", *Wireless Africa Manual*, 2007.

[17] B. Raman and K. Chebrolu, "Experiences in using WiFi for rural Internet in India", *IEEE Communications Magazine*, vol. 45, pp. 104-110, 2007.

[18] J. Shmael, S. Bury, D. Pezaros, and N. Race, "Deploying Rural Community Wireless Mesh Networks," *IEEE Internet Computing*, vol. 12, pp. 22–29, 2008.

[19] D. Anurag, R. Siuli, and B. Somprakash, "AGRO-SENSE: Precision Agriculture Using Sensor-based Wireless Mesh Networks," *Proc. Of Innovations in NGN: Future Network and Services (K-INGN)*, 2008.

[20] Wireless Mesh Networks for Public Safety. White Paper, Bel Air Networks, 2007

[21] MeshDynamics Cooperation, "Mobile Mesh Networks for Military, Defence and Public Safety" http://www.meshdynamics.com/military-mesh-networks.html, May 21 2009 [Nov 2 2012].

[22] L. Hardy, and M. Garfen "A new highly-synchronized wireless mesh network model in use by the Electric Company to switch to automatic meter reading: Case study," *5th International Conference on Networked Sensing Systems (INSS)*, 2008.

[23] M. Li and Y. Liu, "Underground coal mine monitoring with wireless sensor networks," *ACM Transactions on Sensor Networks*, 2009

[24] B. Henderson**, "**Miners Give a Nod to Nodes," *Mission Critical Magazine*, 2008.

[25] S. J Lee, W. Su, and M. Geral, "Ad hoc wireless networks with mobility prediction," *IEEE ICCCN*, 1999.

[26] B. Raffaele, M. Conti, and E. Gregori, "Mesh networks: Commodity Multihop Ad Hoc Networks," *IEEE Communication Magazine*, 2005.

[27] C.E Perkins and P. Bhagwat, "Highly Dynamic destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers," *ACM SIGCOMM*, 1994.

[28] I. Chakeres and C. Perkins, "Dynamic MANET On Demand (DYMO) Routing," *draft-ietf-manet-dymo-17.txt*, 2009.

[29] D. Johnson, D. Maltz, Y. Hu, and J, Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," *RFC 4728*, 2007.

[30] VD Park and MS Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," *IEEE INFOCOM*, 1997.
[31] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," *ACM MOBICOM*, 2004.
[32] D. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A High Throughput Path Metric for Multi-Hop Wireless Routing," *ACM Mobicom*, 2003.
[33] J. Jun and M. L. Sichitiu, "The nominal capacity of wireless mesh networks," *IEEE Wireless Communications Magazine, Special Issue on: Merging IP and Wireless Networks,* Oct 2003.
[34] C. E. Perkins, S. Karim, W. Cedric, and S. Mahesh, "Better Plumbing for Reduced Flooding," *Proceedings of 68th IETF*, 2007.
[35] A. Bourkeche, "Performance Evaluation of Routing Protocols for Ad Hoc Networks," *Mobile Networks and Applications*, vol 9, pp. 333-342, 2004.
[36] D. Broch, A. Maltz, D.B. Johnson, Y.C Hu, and J. Jetcheva, "A Performance Comparison of Multihop Ad Hoc Network Routing Protocols," *ACM MOBICOM*, 1998.
[37] S. R. Das, C. E. Perkins, and E. M. Royer, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks," *IEEE INFOCOM*, 2000.
[38] IEEE Standards Department, Wireless LAN medium Access Control (MAC) and Physical Layer (PHY) specifications, IEEE standard 802.11, 1997
[39] H.X Tan, W.K.G Seah, "Dynamically Adapting Mobile Ad Hoc Routing Protocols to Improve Scalability," *International Conference on Communication Systems and Network (CSN),* 2004
[40] C. Gomez, A. Cuevas, J. Paradells, "AHR: A Two-State Adaptive Mechanism for Link Connectivity Maintenance in AODV," *ACM REALMAN,* 2006
[41] S. Ahn, A.U Shankar, "Adapting to Route-Demand and Mobility," *9th International Conference on Network Protocols (ICNP)*, 2001
[42] R. Boppana, S. Konduru, "An Adaptive Distance Vector Routing Algorithm for Mobile Ad-Hoc Networks," *IEEE INFOCOM 2001*
[43] O. Stanze, M. Zitterbart, C. Koch, "Mobility Adaptive Self-Parametrization of Routing Protocols for Mobile Ad-Hoc Networks," *IEEE WCNC,* 2006
[44] T. Ramrekha, E. Panaousis, G. Millar, C. Politis, "Chameleon (CML): A Hybrid and adaptive routing protocol for Emergency Situations," *IETF MANET Working Group Draft,* 2010
[45] Rendong Bai, Mukesh Sighal, "DOA: DSR over AODV Routing for Mobile Ad-Hoc Networks," *IEEE Transactions on Mobile Computing,* Volume 5, No. 10, Oct 2006
[46] J. Boleng, "Exploiting Location Information and Enabling Adaptive Mobile Ad-Hoc Network protocols," *PhD Thesis, Colorado School of Mines,* 2002
[47] H.X Tan, W.K.G Seah, "Dynamic Topology Control to Reduce Interference in MANETs," *2nd International Conference on Mobile Computing & Ubiquitous Networking (ICMU),* 2005
[48] L. Qin, T. Kunz, "Mobility metrics to enable adaptive routing in MANET," *2nd IEEE International Conference on Wireless & Mobile Computing, Networking & Communications (WiMob),* 2006
[49] T. Ramrekha, E. Panaousis, G. Millar, C. Politis, "An Adaptive QoS Routing Solution for MANET Based Multimedia Communications in Emergency Cases," *ICST International Conference on Mobile Lightweight Wireless Systems (MOBILIGHT),* 2009
[50] N. Pereira and R.M De Mores, "A Comparative Analysis of AODV Route Recovery Mechanisms in Wireless Ad-Hoc Networks", *IEEE Latin American Conference on Communication*, 2009, pp 1-6
[51] Saaidal R. Azzuhri, M. Portamn, W.L Tan, "Evaluation of Parameterised Route Repair in AODV", *4th International Conference On Signal Processing and Communication Systems (ICSPCS)*, 2010, pp 1-7
[52] J. Liu and Chun-Hung Richard Lin, "RBR: refinement-based route maintenance protocol in wireless ad hoc networks". *Computer Communications*, vol. 28, no. 8, pp. 908-920, May 2005.
[53] A. Al-Shanyour and U. Baroudi, "Bypass AODV: Improving Performance of Ad hoc On-Demand Distance Vector (AODV) Routing Protocol in Wireless Ad Hoc Networks," *Proceedings of the 1st international conference on Ambient media and systems (Ambi-sys)*, Nov 2008.
[54] C. Sengul, R. Kravets, "Bypass Routing: An on-demand local recovery protocol for ad-hoc networks," *Ad Hoc Networks*, vol. 4, no. 3, pp. 380-397, May 2006.
[55] Z. Kai; W. Neng and L. Ai-fang, "A new AODV based clustering routing protocol". *Proceedings of Wireless Communications, Networking and Mobile Computing (WCNM)*, Sept. 2005
[56] Y. Noishiki, H. Yokota and A. Idoue, "Efficient on-demand route establishment methods for dense ad-hoc networks". *Proceedings of Autonomous and Decentralized Systems (ISADS)*, April 2005
[57] Kevin Fall, Kanna Varadhan, "The ns manual" http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf. Sept 9 2009 [Nov 4 2011].

[58] D.B. Johnson and D.A. Maltz, “Dynamic Source Routing in Ad Hoc Wireless Networks,” in T. Imelinski & H. Korth, eds. Mobile Computing, Kluwer Academic Publisher, 1998.

[59] T. Clausen, C. Dearlove, J. Dean, the OLSRv2 Design Team, and the MANET Working Group, “MANET Neighborhood Discovery Protocol (NHDP),” Internet-Draft, draft-ietf-manet-nhdp-00.txt, June 2006. [Online].

[60] F. J. Ros and P. M. Ruiz. "Implementing a new manet unicast routing protocol in NS2". Dept. of Information and Communications Engineering University of Murcia, December 2004.

[61] CMU Monarch project, Computer Science Department, Canergie Mellon University, Pittsburgh. "The CMU Monarch project's wireless and mobility extensions to ns", August 1999

[62] Haas Z.J.; Pearlman M.R.; Samar P. *The Zone Routing Protocol (ZRP) for Ad Hoc Networks*, IETF Internet Draft, July 2002.

[63] Y. Yang, J. Wang, and R. Kravets, “Designing routing metrics for mesh networks,” in IEEE Workshop on Wireless Mesh Networks (WiMesh), Sept. 2005.

[64] G.R Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, Walke, B, “IEEE 802.11s: The WLAN Mesh Standard,” in IEEE Wireless Communication Magazine, Feb. 2010

[65] P. Juang, H. Oki, Y. Wang, et al, “Energy Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet,” In Proceeding of ASPLOS-X, Oct 2002.

[66] A. Doria, M. Udon, and D. P. Pandey, “Providing connectivity to the saami nomadic community,” In Proc. 2nd Int. Conf. on Open Collaborative Design for Sustainable Innovation, Dec. 2002.

[67] K. Fall, “A delay-tolerant network architecture for challenged Internets,” in ACM SIGCOMM, (Karlsruhe, Germany), Aug. 2003.

[68] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, “Delay-tolerant networking architecture.” RFC 4838, Apr. 2007.

[69] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, “Delay-tolerant network architecture: The evolving interplanetary internet.” Internet Draft: draft-irtf-ipnrg-arch-01.txt, Feb. 2003.

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

[71] Feng, Zhenxin, and Kwan-Wu Chin. "A Survey of Delay Tolerant Networks Routing Protocols." *arXiv preprint arXiv:1210.0965* (2012).

[72] A. Tovar, T. Friesen, K. Ferens and B. McLeod, “A DTN Wireless Sensor Network for Wildlife habitat monitoring”, Proceeding of the Electrical and Computer Engineering, May 2010.

[73] Araki, Mohammad Zarafshan. *TrainNet: A Novel Transport Infrastructure for Non Real-Time Data Delivery*. University of Wollongong, 2009.

[74] K. Scott and S. Burleigh, “Bundle Protocol Specification”, *IETF RFC 5050,* 2007.

[75] Forrest Warthman, “Delay Tolerant Networks Tutorial”, http://www.dtnrg.org/docs/tutorials/warhman-1.1.pdf, May 2 2003 [Aug. 4 2012].

[76] Harris Abdullah, “A DTN Study: Analysis of Implementation and Tools”, *Master`s Thesis, TKK / Informaatio- ja luonnontieteiden tiedekunta, 2010.*

[77] Z.Zhang, “Routing in intermittently connected MANET and DTN: Overview and Challenges”, *IEEE Communications Surveys & Tutorials,* Volume 8, Issue 1, p.p 24-37, 2006.

[78] A. Balasubramaniam, B.N Levine, A. Venkataramani, “DTN routing as a resource allocation problem”, ACM *SIGCOMM*, p.p 373-384, 2007.

[79] Vahdat, Amin, and David Becker. *Epidemic routing for partially connected ad hoc networks*. Technical Report CS-200006, Duke University, 2000.

[80] Spyropoulos, Thrasyvoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. "Spray and wait: an efficient routing scheme for intermittently connected mobile networks." *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. ACM, 2005.

[81] Spyropoulos, Thrasyvoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility." *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops' 07. Fifth Annual IEEE International Conference on*. IEEE, 2007.

[82] Burgess, J., et al. "Maxprop: Routing for vehicle-based delay-tolerant networks." *IEEE Infocom*. 2006.

[83] Lindgren, Anders, Avri Doria, and Olov Schelen. "Probabilistic routing in intermittently connected networks." *Service Assurance with Partial and Intermittent Resources*. Springer Berlin Heidelberg, 2004. 239-254.

[84] Lindgren, Anders, and Avri Doria. *Probabilistic routing protocol for intermittently connected networks, draft-lindgren-dtnrg-prophet-03*. IETF Internet-Draft, 2007.

[85] A. Parker, A. Kansal, A. A. Somasundara, D. D. Jea, M. B. Strivastava, and D. Estrin. "UCLA DTN Sensor Networks Update" http://www.dtnrg.org/wiki, Aug 3 2004 [May 3 2012].

[86] Uden, Maria, and Avri Doria. "Technology producers meeting indigenous users: the case of Sami network connectivity." *International Journal of Agricultural Resources, Governance and Ecology* 6.6 (2007): 693-705.

[87] European Union Research Group "Networking for Communications challenged Communities (N4C)" http://www.n4c.eu/, Nov. 2 2006 [June 3 2012].

[88] Balasubramanian, Aruna, Brian Levine, and Arun Venkataramani. "DTN routing as a resource allocation problem." *ACM SIGCOMM Computer Communication Review*. Vol. 37. No. 4. ACM, 2007.

[89] Blazevic, Ljubica, J-Y. Le Boudec, and Silvia Giordano. "A location-based routing method for mobile ad hoc networks." *Mobile Computing, IEEE Transactions on* 4.2 (2005): 97-110.

[90] R. Poor. "Gradient routing in ad hoc networks." (2000): 10-3.

[91] Hui, Pan, Jon Crowcroft, and Eiko Yoneki. "Bubble rap: social-based forwarding in delay tolerant networks." *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2008.

[92] de Oliveira, Etienne CR, and Célio VN de Albuquerque. "NECTAR: a DTN routing protocol based on neighborhood contact history." *Proceedings of the 2009 ACM symposium on Applied Computing*. ACM, 2009.

[93] Saaidal R Azzuhri, Marius Portmann, and Wee Lum Tan. "Evaluation of parameterised route repair in AODV." *Signal Processing and Communication Systems (ICSPCS), 2010 4th International Conference on*. IEEE, 2010.

[94] Musolesi, Mirco, and Cecilia Mascolo. "Car: context-aware adaptive routing for delay-tolerant mobile networks." *Mobile Computing, IEEE Transactions on* 8.2 (2009): 246-260.

[95] M. Basagni, M. Conti, S. Giordano and I. Stojmenovic, "Mobile Ad-Hoc Networking", *WILEY-IEEE*, New Jersey, 2004.

[96] J. Whitback and V. Conan. "HYMAD: Hybrid DTN-MANET routing for dense and highly dynamic wireless networks." *Computer Communications* 33.13 (2010): 1483-1492.

[97] J. Ott, D. Kutscher and C. Dwertmann, "Integrating DTN and MANET Routing", *in Proceeding of MSWim,* Turkey, 2010.

[98] P. Holliday, "NOMAD A Mobile Ad Hoc and Distruption Tolerant Routing Protocol for Tactical Military Networks." *Distributed Computing Systems Workshops, 2009. ICDCS Workshops' 09. 29th IEEE International Conference on*. IEEE, 2009.

[99] P. Holliday, "SWARMM-a mobility modelling tool for tactical military networks." *Military Communications Conference, 2008. MILCOM 2008. IEEE*. IEEE, 2008.

[100] Delosieres, Laurent, and Simin Nadjm-Tehrani. "BATMAN store-and-forward: The best of the two worlds." *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*. IEEE, 2012.

[101] Kretschmer, Christian, Stefan Ruhrup, and Christian Schindelhauer. "Dt-dymo: delay-tolerant dynamic manet on-demand routing." *Distributed Computing Systems Workshops, 2009. ICDCS Workshops' 09. 29th IEEE International Conference on*. IEEE, 2009.

[102] Jani Lakkakorpi , Mikko Pitkanen , Jorg Ott, "Adaptive routing in mobile opportunistic networks," *Proceedings of the 13th ACM international conference on Modeling, analysis, and simulation of wireless and mobile systems*, October 17-21, 2010, Bodrum, Turkey.

[103] Francisco J. Ros, "UM-OLSR" http://masimum.inf.um.es/fjrm/?page_id=116, May 21 2009 [Aug. 23 2011].

[104] University of Bonn, "Bonnmotion" http://sys.cs.uos.de/bonnmotion, March 3 2009 [June 3 2012]

[105] IEEE P802.11s™/D1.06, draft amendment to standard IEEE 802.11™: Mesh Networking. IEEE, May 2007.

[106] I. Chakeres and E. Belding-Royer. "AODV Implementation Design and Performance Evaluation," *International Journal of Wireless and Mobile Computing (IJWMC),* Issue 2/3, 2005.

[107] Uppsala University, "AODV-UU" http://freecode.com/projects/aodvruu March 26 2011 [April 23 2012].

[108] Francisco J. Ros, "UM-DYMO" http://masimum.inf.um.es/fjrm/?page_id=42 May 21 2009 [Aug. 23 2011].

[109] C. R. Aponte and S. Bohacek, "OLSR and approximate distance routing: Loops, black holes, and path stretch," *4th International Conference on Communication Systems and Networks (COMSNETS)*, 2012.

[110] Institute of Information Transmission Problems of the Russian Academy of Sciences, "Hybrid Wireless Mesh Protocols (HWMP) for ns-2" https://forge.iitp.ru/ns2/hwmp/ May 21 2008 [Feb. 23 2012].

[111] Saaidal R. Azzuhri, M. Portamn, W.L Tan, "Evaluating the performance impact of protocol parameters on ad-hoc network routing protocols", *Australasian Telecommunication Networks and Application Conference (ATNAC)*, 2012, pp 1-6.

[112] R. Pathak, Peizhao Hu, J. Indulska, M. Portamann, Wee Lum Tan, "Towards efficient opportunistic communications: a hybrid approach." *In IEEE International Conference on Pervasive Computing and Communications (PERCOM) Workshops*, 2013.

[113] Erik Kuiper, "Spray and Wait" http://www.ida.liu.se/labs/rtslab/code/LAROD-LoDiS/, Nov 1 2010 [Jan. 8 2013].

[114] Eric Kuiper, "Geographic Routing in Intermittently-connected Mobile Ad Hoc Networks: Algorithms and Performance Models", *PHD Thesis, Linkoping Studies in Science & Technology, Linkoping University Electronic Press,* 2012.

[115] A Neumann, C Aichele, M Lindner, S Wunderlich – Better Approach To Mobile Ad-Hoc Network (BATMAN),IETF **draft**, http://tools.ietf.org/html/draft-wunderlich-openmesh-manet-routing-00, Oct 10 2008 [Jan. 3 2013].

[116] Fahim Maan, Nauman Mazhar. "Manet routing protocols vs mobility models: A performance evaluation." *In IEEE Third International Conference on Ubiquitous and Future Networks (ICUFN),* 2011.

[117] Haerri, Jerome, Fethi Filali, and Christian Bonnet. "Performance comparison of AODV and OLSR in VANETs urban environments under realistic mobility patterns." In *Proc. of 5th IFIP Mediterranean Ad-Hoc Networking Workshop (Med-Hoc-Net-2006), Lipari, Italy*. 2006.

[118] Huhtonen, Aleksandr. "Comparing AODV and OLSR routing protocols." In *Seminar on Internetworking, Sjkulla*, pp. 26-27. 2004.

[119] Hassan, Yasser Kamal, Mohamed Hashim Abd El-Aziz, and Ahmed Safwat Abd El-Radi. "Performance Evaluation of Mobility Speed over MANET Routing Protocols." *IJ Network Security* 11, no. 3 (2010): 128-138.