Helsinki University of Technology, Networking Laboratory
Teknillinen Korkeakoulu, Tieteverkkolaboratorio

Espoo 2007 Report 4/2007

# A Hybrid Routing Approach for Ad hoc Networks

Jose Costa-Requena

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Department of Electrical and Communications Engineering, for public examination and debate in Auditorium S1 at Helsinki University of Technology (Espoo, Finland) on the 7th of December, 2007, at 12 noon.

Helsinki University of Technology
Department of Electrical and Communications Engineering
Networking Laboratory

Teknillinen Korkeakoulu
Sahko- ja Tietoliikennetekniikan Osasto
Tietoverkkolaboratorio

Distribution:
Helsinki University of Technology
Networking Laboratory
P.O. Box 3000
FIN-02015 TKK
Finland

Tel. +358 9 451 2461
Fax. +358 9 451 2474

ISBN 978-951-22-8910-3
ISSN 1458-0322

Otamedia Oy
Espoo 2007

| | |
|---|---|
| **ABSTRACT OF DOCTORAL DISSERTATION** | **HELSINKI UNIVERSITY OF TECHNOLOGY**<br>**P.O.BOX 1000, FIN-02015 HUT**<br>**http://www.hut.fi/** |

**Author** Lic. Jose Costa-Requena

**Name of the dissertation**
**A Hybrid Routing Approach for Ad hoc Networks**

| | |
|---|---|
| **Manuscript submitted** 20.8.2007 | **Manuscript revised** 20.11.2007 |

**Date of Defence** 7.12.2007

| | |
|---|---|
| ☒ **Monograph** | ☐ **Article dissertation (summary + original articles)** |

| | |
|---|---|
| **Department** | **Department of Electrical and Communications Engineering** |
| **Laboratory** | **Networking Laboratory** |
| **Field of Research** | **Networking, Ad hoc networks** |
| **Opponent(s)** | **Professor Pietro Michiardi (Eurecom)** |
| **Pre-examiners** | **Professor Timo Hamäläinen (University of Jyväskylä) and Professor Carlos Pomalaza-Raez (Indiana-Purdue University)** |
| **Supervisor** | **Professor Raimo Kantola (Helsinki University of Technology)** |

**Abstract**

**Ad hoc networking is a technology still under development and there are several proposals for defining the most suitable routing protocol. No single routing protocol proposed so far performs optimally under the kind of dynamic conditions possible in Ad hoc networks.**

**We analyse the performance of existing Ad hoc routing protocols using simulations and a test bed. Based on the results, the goal of this thesis is to design a hybrid routing approach for Ad hoc networks that we name Scalable Ad hoc Routing Protocol (SARP). A novel routing algorithm that responds to the drawbacks of existing routing protocols is analysed and implemented. However, rather than proposing another protocol, this study extends the well-known routing protocol, Ad hoc On Demand Distance Vector (AODV), with a new broadcast algorithm to accommodate the new routing design.**

**The contribution of the nodes to the routing functionality is critical for establishing Ad hoc networks. We analyse the incentives to participate in the routing functions using game theory. The Scalable Ad hoc Routing Protocol defines a novel architecture that integrates with the routing protocol a rewarding mechanism for the participating nodes. This architecture facilitates the cooperation of the nodes in the Ad hoc networks routing functionality.**

**Keywords** Ad hoc networking, routing, QoS measurements, game theory.

| | |
|---|---|
| **ISBN (printed)** 978-951-22-8910-3 | **ISSN (printed)** 1458-0322 |
| **ISBN (pdf)** 978-951-22-8911-0 | **ISSN (pdf)** 1458-0322 |
| **Language** English | **Number of Pages** 157 |

**Publisher** Networking Laboratory / Helsinki University of Technology

**Print distribution**

☒ **The dissertation can be read at http://lib.tkk.fi/Diss/2007/isbn9789512289110/**

| | |
|---|---|
| **VÄITÖSKIRJAN TIIVISTELMÄ** | **TEKNILLINEN KORKEAKOULU**<br>**P.O.BOX 1000, FIN-02015 TKK**<br>**http://www.tkk.fi/** |
| **Tekijä TKL. Jose Costa-Requena** | |
| **Vaitöskirjan nimi**<br>**Hybridireititysmenetelemä Ad hoc Verkkoja varten** | |
| **Käsikirjoituksen päivämäärä 20.8.2007** | **Korjatun käsikirjoituksen 20.11.2007** |
| **Väitöstilaisuuden ajankohta 7.12.2007** | |
| ☒ **Monografia** | ☐ **Yhdistelmäväitöskirja (yhteenveto ja erillisartikkelit)** |

| | |
|---|---|
| **Osasto** | **Sähkö- ja tietoliikenntekniikan osasto** |
| **Laboratorio** | **Tietoverkkolaboratorio** |
| **Tutkimusala** | **Adhocverkot** |
| **Vastaväittäjä** | **Professori Pietro Michiardi (Eurecom)** |
| **Esitarkastajat** | **Professori Timo Hamäläinen (University of Jyväskylä) ja Professori Carlos Pomalaza-Raez (Indiana-Purdue University)** |
| **Työn valvoja** | **Professori Raimo Kantola (Teknillinen Korkeakoulu)** |


**Tiivistelmä**

**Ad hoc verkot on vielä kehityksen alla oleva teknologia ja tarkoitukseen sopivia reititysprotokollia on ehdotettu useita. Yksikään tähän asti ehdotettu reititysprotokolla ei toimi optimaalisesti ad hoc verkkojen mahdollisesti muuttuvissa olosuhteissa.**

**Työssä analysoidaan olemassa olevia ad hoc reititysprotokollia simuloinnin ja koeympäristön avulla. Näiden tulosten perusteella tämän tutkimuksen tavoitteena on suunnitella reititysmenetelmä, jota kutsutaan skaalautuvaksi ad hoc reititysprotokollaksi (SARP, Scalable Ad hoc Routing Protocol). Työssä analysoidaan ja toteutetaan uudenlainen reititysalgoritmi, joka ratkaisee nykyisten protokollien ongelmia. Työssä ei kuitenkaan ehdoteta kokonaan uutta reititysprotokollaa, vaan uusi reititysmenetelmä toteutetaan laajentamalla AODV (Ad hoc On Demand Distance Vector)-reititysprotokollaa uudella yleislähetysmekanismilla.**

**Solmujen osallistuminen reititystoimintaan on ad hoc verkkojen muodostumisessa tärkeää. Analysoimme halukkuutta osallistua reititystoimintoihin peliteorian avulla. SARP -protokolla määrittelee uuden arkkitehtuurin, joka sisältää osallistuvia solmuja palkitsevan mekanismin. Tämä arkkitehtuuri tukee solmujen yhteistyötä reititystoiminnassa.**



| | |
|---|---|
| **Asiasanat Ad hoc verkot, reititysprotokolla, QoS liikennemittaukset, game theory.** | |
| **UDC** | **Sivumäärä 157** |
| **ISBN (painettu)… 978-951-22-8910-3** | **ISBN (pdf) 978-951-22-8911-0** |
| **ISBN (muut) 978-951-22-8911-0** | **ISSN 1458-0322** |
| **Julkaisija Tietoverkkolaboratorio / Teknillinen Korkeakoulu** | |
| **Paonetun väitöskirjan jakelu julkaisijan toimesta.** | |
| ☒ **Luettavissa verkossa osoitteessa http://lib.tkk.fi/Diss/2007/isbn9789512289110/** | |

## Acknowledgements

This work has been carried out at the Networking Laboratory of the Helsinki University of Technology and I want to thank everyone there who made me feel like at home since I arrived in February 98. My most sincere gratitude to Professor Raimo Kantola who has been there with his continuous guidance and dedication from the beginning with my Master Thesis up to this Doctoral Thesis.

I would also like to thank the people I met in the MobileMAN EU project who gave me new ideas and insights on the Ad hoc networking that helped me in the development of this work. I also want to thank professors Timo Hamäläinen and Carlos Pomalaza-Raez for their valuable comments on the review and professor Pietro Michiardi for accepting being my opponent.

Many thanks to the dozens of students I have instructed over the past years on their Master Thesis related to this work but also on different topics. Working together provided me fresh food for thoughts.

My parents, Tomas and Fina, they taught me the right values and encouraged me to search for new challenges with their lifetime hard work support. My sister and brother, Gemma and Tomas, who have been there all the way and I know they will be always there for me. My parents in law, Pedro and Concha, with their initiative made me enjoy the work in fancy places.

Queca, your unconditional support has made this possible. You helped me to keep on track and focus to get it done. This is the result of your endless patience, your reviews and our long working weekends. The unforgettable story we have behind, the effort and time we spent together to achieve this goal is part of our journey and part of our best foundation. Now, I am looking forward to enjoying our new life that has already started with Pepito. FTWTW.

May, 2007 Espoo, Finland

Lic. Jose Costa-Requena

## Table of Contents

## Acronyms

| | |
|---|---|
| 3G | 3rd Generation Wireless Networks |
| AODV | Ad Hoc On Demand Distance Vector |
| AS | Autonomous System |
| BGP | Border Gateway Protocol |
| CBR | Constant Bit Rate |
| CGRS | Cluster head-Gateway Switching Routing |
| CIDR | Classless Inter Domain Routing |
| DHCP | Dynamic Host Configuration Protocol |
| DNS | Domain Name System |
| DSR | Dynamic Source Routing |
| DSDV | Destination Sequenced Distance Vector Routing |
| ERC | Equity Reciprocity and Competition Theory |
| FDVB | Fully Distributed Virtual Backbone |
| FSLS | Fuzzy Sighted Link State |
| GSM | Global System Mobile communications |
| HSLS | Hazy Sighted Link State |
| HSR | Hierarchical State Routing |
| IERP | Inter-Zone Routing Protocol |
| IETF | Internet Engineering Task Force |
| IP | Internet Protocol |
| ITU-T | International Telecommunication Union |
| IZRP | Intra-Zone Routing Protocol |
| LAR | Location Aided Routing |
| LCC | Least Cluster head Change |
| LSU | Link State Update |
| MAC | Medium Access Control |
| MANET | Mobile Ad hoc Networks |
| MPR | Multipoint Relay |
| OLSR | Optimised Link State Routing |
| OSPF | Open Shortest Path First |

| | |
|---|---|
| PAN | Personal Area Networks |
| PCM | Pulse Code Modulation |
| PDA | Personal Digital Assistant |
| POTS | Plain Old Telephony Service |
| QoS | Quality of Service |
| RREQ | Route Request |
| RREP | Route Reply |
| RTP | Real Time Protocol |
| SARP | Scalable Ad hoc Routing Protocol |
| TORA | Temporally Ordered Routing Algorithm |
| ToS | Type of Service |
| TKK | Teknillinen Korkeakoulu (Helsinki University of Technology) |
| TTL | Time To Live |
| UDP | User Datagram Protocol |
| VoIP | Voice over IP |
| WLAN | Wireless Local Area Networks |
| ZRP | Zone Routing Protocol |

## List of Tables

## List of Figures

# Chapter 1

## Introduction

Ad hoc networks are envisioned as a key technology for ubiquitous networking. It is a suitable technology for embedded network devices in multiple environments such as vehicles, mobile telephones and personal appliances. As an *infrastructure-less* technology, it will allow users to create their Personal Area Networks (PAN). The benefit of Ad hoc networks is that users can create the network automatically when needed and tear it down if it is not required anymore. The network can be created at any point in time for any communication purpose such as leisure, military or disaster situations. Ad hoc networks have an undefined lifetime since they can be up and running momentarily or permanently as long as there is a group of users that are willing to be part of the network.

Nowadays, mobile computers and personalized applications are indispensable. Users demand connectivity at any time at any place, even where the appropriate infrastructure is not available. In this kind of scenarios, it is necessary that wireless devices learn how to communicate among themselves without routers, base stations or service providers. Ad hoc networks could be the solution to fulfil these user needs but they present new challenges that have not been primary concerns in fixed networks deployment until now.

### 1.1 Networking Requirements in Ad hoc Networks

In Ad hoc networks the link state information changes whenever users move and create interferences to each other. Ad hoc networks are self-established without previous knowledge of the environment. Ad hoc nodes require a set of mechanisms to allow the devices to be autonomously integrated and configured as part of the Ad hoc network.

Network scalability is the ability to expand or reduce the number of nodes and size of the network while maintaining similar performance for each user. Ad hoc nodes have to perform the routing functionality and maintain the network topology information, while keeping track of the connection with other nodes. They must also be able to react fast to network changes and dynamically adapt to the new topology. Therefore, the overall Ad hoc network performance is affected by the size of the network, the number of nodes, their mobility and resources.

Ad hoc nodes cannot rely on a fixed server that would inform about the services available in the Ad hoc network. Therefore, each node needs its own mechanism to discover the network capabilities and configure itself to the services available in the Ad hoc network. Besides these, Ad hoc networks have to interconnect with other IP based technologies such as fixed Wireless Local Area Networks (WLAN) and 3G networks. For that reason, Ad hoc nodes have to act as routers and constantly search for the services available in the networks. The nodes that become part of Ad hoc networks contribute to the overall network performance while spending their own resources. This leads to a high energy consumption that exhausts the batteries of the nodes.

**1.2 Objectives of the Thesis**

In recent years it has been proven that there is no single protocol that accommodates different conditions in Ad hoc networks [1] [2]. Moreover, not all the nodes have the same requirements in terms of mobility and resources. Therefore, it is difficult to design a single protocol that simultaneously meets all the network variations and the different node requirements.

The objective of this thesis is to design and implement a new hybrid routing approach named Scalable Ad hoc Routing Protocol (SARP). The main purpose of

SARP is to enable Ad hoc networks scalability. This approach has to be able to meet the demands of the Ad hoc network when it reduces or increases the size and the number of nodes. Moreover, it has to be suitable for nodes with different mobility and resource constrains. Test bed results and simulations of existing routing protocols are used as the basis for SARP design. A mathematical model of Ad hoc networks is defined to evaluate SARP performance and optimize the protocol.

A protocol enabling Ad hoc networks scalability requires that some nodes spend additional resources, which may lead into unfairness. This thesis proposes a new algorithm assessed using game theory [3] that provides a rewarding mechanism for the Ad hoc nodes contributing towards network scalability. Besides that, a cross-layer architecture is designed to implement the rewarding algorithm. With this approach the Ad hoc nodes obtain a fair added value in return for their contribution to the routing functionality.

SARP is integrated with the cross-layer architecture for enabling network scalability and implementing the rewarding mechanism. The analysis of the existing protocols together with the mathematical model evaluation supported the selection of the Ad hoc On Demand Distance Vector (AODV [4]) as the basis for SARP implementation.

### 1.3 Our Contribution

We have studied the different routing protocols used in Ad hoc networks, and found that each protocol has different drawbacks and benefits depending on the network topology. We propose a network model based on the results obtained from simulations and a test bed.

Our main contribution is the following:

1. We run simulations to evaluate the performance of different Ad hoc routing protocols. The author in cooperation with other students

implemented a test bed with a voice over IP application, and the results were compared to the ones obtained in the simulations. The outcome of this work is part of the MobileMAN EU project IST-2001-38113 [5].

2. Based on the results from the simulations and the test bed, we propose a routing protocol to fix some of the drawbacks of reactive, proactive and some hybrid routing protocols. Using those results as baseline, we devise a mathematical model to evaluate the network performance of existing Ad hoc routing protocols and compare the results with the proposed routing protocol.
3. We apply game theory [3] to analyse the incentives required to deploy the proposed routing protocol. Moreover, based on the game analysis, a cross-layer architecture with a rewarding system is proposed for implementing the incentives.

The author’s original contributions can be found in this thesis and the following publications.

> The author instructed nine Master Thesis as preliminary work leading to this thesis. Preliminary results of what will be published in this thesis were reported in the respective nine Master Thesis and joint conference papers based on those Master Thesis. In particular, Master Thesis [6] includes part of the simulation results presented in Chapter 2. Master Thesis [7], [8], [9], [10], [11] and [12] develop the Ad hoc test bed, and Master Thesis [13] and [14] provide the test bed performance results partly used in Chapter 2.
>
> The early simulations and the initial hybrid routing proposal included in Chapter 2 can be found in [15]. Some of the test bed results in Chapter 2 are published in [16]. The performance metrics model based on the simulation and test bed results that are used to propose the new fully distributed virtual backbone (FDVB) algorithm is published in [17]. A subset of the implementation presented in Chapter 4 including the route

cache replication and the original proposal of the FDVB based on smart nodes is published in [18] and [19]. The architecture proposed in Chapter 4 to implement the FDVB for supporting network scalability can be found in [20] and [21]. Preliminary work including the network incentives to implement the proposed hybrid routing protocol is published in [22].

In addition to the publications directly related to Ad hoc networking, the author previously contributed to Internet addressing, numbering and IN interoperability routing research. Those are used in this work as background to analyse scalability in IP networks [23], [24] and [25].

Therefore, part of the content included in several Chapters of this thesis can be found in existing publications. However, this thesis includes improved versions of the work presented in those publications. Chapter 2 includes new propositions obtained from recent simulations. Chapter 3 contains an updated version of the performance models and simulation results not included in previous publications. Chapter 4 contributes with new conclusions obtained after reformulating the game analysis, which are not published in any previous work. The instructed Master Theses include an early protocol design that has been updated in Chapter 5 with new algorithms identified after obtaining some preliminary test results from prototype implementations. Therefore, the work published in the Master Thesis, conference papers and journals include the preliminary results used as baseline for this work. Nevertheless, this thesis presents new findings and conclusions formulated with more detail than in previous publications.

This thesis is structured as a monograph instead of an article dissertation to present a more coherent and accurate report of the work done by the author and the students working on this subject. This thesis provides a comprehensive presentation of the results and a progressive analysis of the subject. Therefore, this work starts with simulations and a test bed to provide the basic analysis that is followed by a mathematical model to evaluate the network performance. To conclude, we introduce a theoretical analysis based on game theory to describe the incentives for implementing the proposed routing protocol and support scalability in Ad hoc networks.

### 1.4 Structure of the Thesis

Chapter 2 presents the performance evaluation of existing Ad hoc routing protocols. The results demonstrate that there is no single protocol suitable for all the Ad hoc networks. This chapter also highlights the scalability limitations of some of the existing routing protocols. Based on the performance evaluation we design a novel hybrid routing approach for Ad hoc networks named Scalable Ad hoc Routing Protocol (SARP). SARP is specified as a fully distributed virtual backbone (FDVB) algorithm.

Chapter 3 defines a mathematical model to evaluate SARP performance and optimize the protocol. The results are used to specify the optimal requirements for the FDVB algorithm.

Chapter 4 presents the incentives for the nodes to participate in SARP routing functionality. In this chapter game theory [3] is applied to demonstrate that SARP requires a cross-layer architecture implementing a rewarding mechanism.

Chapter 5 describes the SARP implementation on top of a reactive routing protocol, the Ad hoc On demand Distance Vector (AODV) [4]. A novel architecture based on a cross-layer interaction with the routing protocol is studied.

Chapter 6 presents our conclusions and future work.

# Chapter 2

## Ad hoc Routing Protocols Analysis

This chapter introduces a performance evaluation of existing Ad hoc routing protocols. The performance results presented in this chapter, obtained from simulations and validated using a test bed, demonstrate that there is no single protocol suitable for all the Ad hoc networks [26]. This chapter highlights the performance of reactive, proactive and hybrid routing protocols in terms of scalability.

### 2.1 Addressing and Reachability

In Ad hoc networks, the nodes perform the addressing and routing functionalities making scalability a critical issue in large networks. Before studying the existing Ad hoc routing protocols and their performance, different addressing approaches are analysed. As baseline for our study, we briefly review the different solutions that have been implemented in fixed networks to handle the scalability problems in addressing.

Addressing is hierarchical (e.g. country code, trunk code and subscriber number) in existing fixed networks such as Plain Old Telephony Service (POTS) [27] where each switch maintains a specific numbering block. IP networks addressing was originally flat [28] but when the number of hosts connected to the network increased, a mechanism to emulate a hierarchical addressing structure dividing the addressing space into groups (i.e. address classes A, B, C and D) was established. The number of nodes kept increasing and the addresses availability was reduced.

Therefore, a more flexible hierarchical scheme, the Classless Inter-Domain Routing (CIDR) [29] was implemented for a more efficient usage of the existing address space.

Maintaining the names and IP addresses of all the hosts in the network up to date, required a continuous exchange of messages resulting in network congestion. Thus, new protocols such as the Dynamic Name Service (DNS) [30], and the Dynamic Host Configuration Protocol (DHCP) [31] were required.

In Ad hoc networks a similar approach has to be followed due to scalability issues. Most of the Ad hoc routing protocols have a flat addressing structure where each node keeps the addresses of the rest of the nodes, similarly to Internet when it was created. However, as history shows, this approach is not suitable when the number of nodes in the network is large. The nodes have to store all IP addresses in their routing tables and they have to maintain the topology information up to date. Therefore, a hierarchical addressing structure is required for scalable Ad hoc networks. The drawback is that Ad hoc networks cannot rely on a fixed entity that assigns the blocks of addresses, making the addressing a significant challenge.

In fixed IP networks moving from flat to hierarchical addressing is feasible because all the nodes are static and they can easily be grouped under sub networks. The IP address space remains flat but it is divided into blocks to emulate hierarchical addressing. Moreover, users want mobility and connectivity with their devices anywhere. DHCP [31] and Mobile IP [32] are the mechanisms for maintaining the flat addressing but still allowing the nodes mobility through different sub networks. DHCP dynamically assigns a new IP address to the nodes accessing the network. Mobile IP enables nodes to be reachable through different sub networks using their static IP address. Ad hoc networks could have applied the same mechanisms (i.e. DHCP or Mobile IP) allowing the nodes to obtain an IP address or maintain their static IP address when joining the Ad hoc network. However, due to the nature of Ad hoc networks [33], the availability of DHCP servers or Mobile IP agents cannot be guaranteed. Instead the Ad hoc nodes must acquire the IP addresses on their own and configure themselves as part of the Ad hoc network.

In fixed networks routers or gateways provide the routing and addressing functionality and the nodes only store the address of the DNS, DHCP server and gateway for routing purposes. In principle, fixed networks are made of many networks (i.e. Autonomous Systems) connected by routers or gateways as depicted in Figure 1. The routers are nodes that use routing protocols such as Open Shortest Path First (OSPF) [34] to maintain addressing information and find the routes between source and destination nodes within the same or different sub networks. The gateways are routers that maintain addressing information about sub networks they are bridging using routing protocols such as the Border Gateway Protocol (BGP) [35].

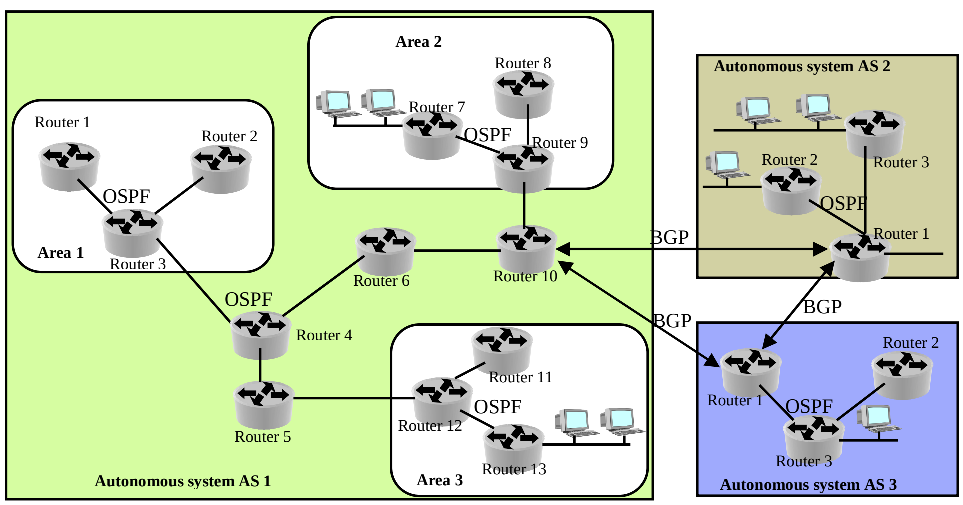


**Figure 1. Routing protocols between autonomous systems.**

When a router receives a packet, it checks the destination address looking up the longest match in the routing table and forwards it to the next router closer to the destination. If no match for the destination address is found in the router, the packet will be forwarded to the default route tied to zero in the routing table. The default route address points to the gateway that maintains addressing information of the other sub networks.

Ad hoc nodes act as routers that cannot rely on any fixed infrastructure devices such as gateways, DHCP or DNS for addressing assistance. Therefore, Ad hoc nodes have to include all necessary routing and addressing functionalities themselves. This means that they must store all routing information and need a mechanism to discover the routes to other nodes that are outside the local sub network.

Scalable Ad hoc networks require a hierarchical addressing structure, where the network is partitioned into sub networks or clusters. Figure 2 represents a cluster-based network with four clusters.

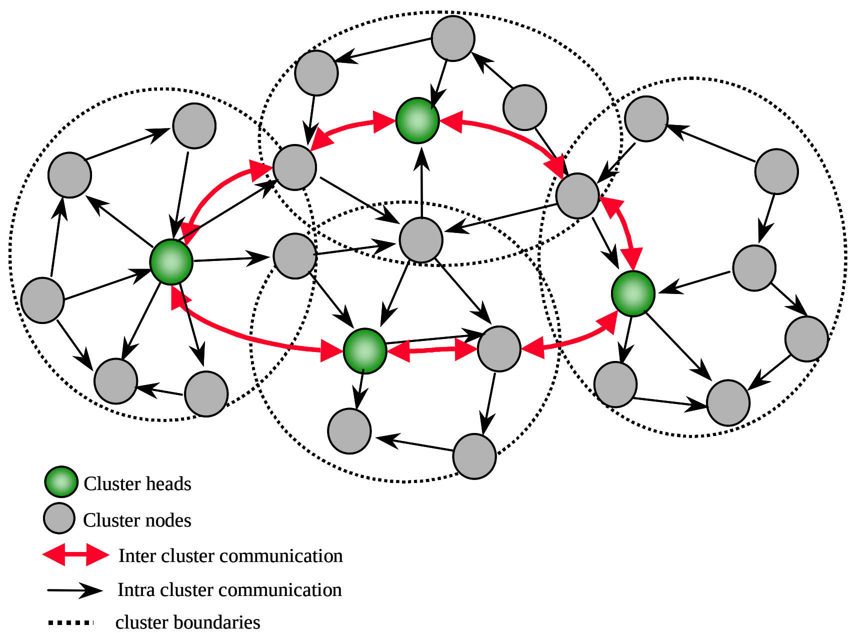


**Figure 2. Cluster-based network routing.**

A cluster-based network is a network divided into several clusters. Each cluster consists of a single cluster head and multiple cluster nodes. The cluster head is a node that performs the routing functionality assigned to gateways in fixed networks. When a cluster node needs to find a route to a destination node not located in the same cluster, it will contact the cluster head that acts as a gateway. The cluster head communicates with other cluster heads in different clusters to find the route to the destination node.

The communication between nodes in the same cluster is known as intra cluster communication. Cluster heads establish the inter cluster communication with nodes outside their own cluster. Cluster heads require additional resources to perform the gateway functionality. The cluster-based routing decreases the network reliability because the cluster head may become the bottleneck. Moreover, the algorithm for selecting the optimal cluster head among the existing cluster nodes is cumbersome.

Nevertheless, from a preliminary analysis on the evolution of the public Internet a hypothesis can be formulated; *a cluster-based routing protocol where the changes in IP addresses and route updates are localised and do not span the entire network, is required to guarantee scalability in Ad hoc networks.*

The evolution path taken in the fixed Internet to solve the scalability problem might not be valid for Ad hoc networks and there is no mathematical analysis to prove that a cluster-based routing protocol is the only solution to make Ad hoc routing scalable. Therefore, in order to verify this claim, next section describes the state of the art in some of the existing Ad hoc routing protocols and their performance. Ad hoc routing protocols can be classified into three categories reactive, proactive and hybrid [5].

## 2.2 Reactive Ad hoc Routing Protocols

Reactive Ad hoc routing protocols determine a path on-demand only, meaning that they search for a single path when a message needs to be delivered. In this section we briefly describe the Ad hoc On Demand Distance Vector (AODV) [4], the Dynamic Source Routing (DSR) [36] and the Temporally Ordered Routing Algorithm (TORA) [37] as the most widely used reactive Ad hoc routing protocols.

In AODV the originating node initiates a Route Request (RREQ) message that is flooded through the network to the destination. The intermediate nodes in the route record the RREQ message. A Route Reply (RREP) unicast message is sent back to the originating node as the acknowledgement following the reverse routes established by the received RREQ message. The intermediate nodes in the route also record the RREP message in their routing table for future use. Each node keeps the most recently used route information in its cache. Therefore, AODV is a simple protocol and does not require excessive resources on the nodes. However, the routing information available in the nodes is limited, and the route discovery process may take too much time. The initial RREQ is sent with TTL=1 and if no RREP is received within a certain time, the TTL is incremented and a new RREQ is sent. Thus, if the destination node is not close enough, the network is flooded several times during the RREQ process before a route is found or an error is notified.

DSR is similar to AODV where RREQ and RREP messages are also used for discovering the route to the destination. The main difference is that in this case, these messages also include the entire path information (i.e. addresses of the intermediate nodes). The drawback is that the route information generates an overhead that can be excessive when the number of hops or node mobility increases.

TORA is a reactive routing protocol with some proactive enhancements where a link between nodes is established creating a Directed Acyclic Graph (DAG) of the route from the source to the destination. The routing messages are distributed to a set of nodes following the graph around the changed topology. TORA provides multiple routes to a destination quickly with minimum overhead. In TORA the optimal routes are of secondary importance versus the delay and overhead of discovering new routes.

## 2.3 Proactive Ad hoc Routing Protocols

The proactive protocols are the traditional routing protocols used in fixed IP networks. These protocols maintain a table with the routing information, and perform periodic updates to keep it consistent. In this section we will introduce the Destination Sequenced Distance Vector Routing (DSDV) [38] and the Optimised Link State Routing (OLSR) [39] as the most representative proactive Ad hoc routing protocols.

DSDV looks for the optimal path using the Bellman-Ford algorithm [40]. It uses a full dump or incremental packets to reduce the traffic generated by the routing updates in the network topology. However, it creates an excessive overhead because it constantly tries to find the optimal path.

OLSR defines Multipoint Relay (MPR) nodes for exchanging the routing information periodically. The nodes select the local MPR node that will announce the routing information to other MPR nodes in the network. The MPR nodes calculate the routing information for reaching other nodes in the network.

## 2.4 Hybrid Ad hoc Routing Protocols

This section introduces a hybrid model that combines reactive and proactive routing protocols but also a location assisted routing protocol.

The Zone Routing Protocol (ZRP) [41] is a hybrid routing protocol that divides the network into zones. The Intra-Zone Routing Protocol (IZRP) implements the routing within the zone, while the Inter-zone Routing Protocol (IERP) implements the routing between zones. ZRP provides a hierarchical architecture where each node has to maintain additional topological information requiring extra memory.

The Location Aided Routing (LAR) [42] is a location assisted routing protocol that uses location information for the routing functionality. LAR works similarly to DSR but it uses location information to limit the area where the route request is flooded. The originating node knows the neighbours location and based on that selects the closest nodes to the destination as the next hop in the route request.

### 2.5 Ad hoc Routing Protocols Evaluation

We have described different routing protocols and based on the basic characteristics of reactive and proactive routing protocols we can formulate a set of propositions. The propositions will consider the impact of system variables such as used routing protocol type, node mobility and number of nodes (i.e. node density) on performance measures such as routing overhead, percentage of packet loss, end to end packet delay and percentage of optimal routes. At this stage we are not able to indicate whether there is a linear or polynomial relationship between the system variables and the performance measures.

AODV, DSR and OLSR, TBRF are the experimental protocols standardized in the IETF as reactive and proactive routing protocols. The routing protocols under consideration in this evaluation are AODV and OLSR as the most representative of reactive and proactive categories.

In our propositions we assume that the following conditions do not change: bit rate, number of flows and size of the Ad hoc network. Let us now formulate the set of propositions using the notations introduced in Table 1 and Table 2.

**Table 1. System variables.**

| Proactive routing protocol | Proactive routing protocol and UDP flows | Proactive routing protocol and TCP flows | Reactive routing protocol | Reactive routing protocol and UDP flows | Reactive routing protocol and TCP flows | Number of nodes in the network or node density | Node mobility |
|---|---|---|---|---|---|---|---|
| P | Pu | Pt | R | Ru | Rt | N | M |

**Table 2. Performance metrics.**

| Routing overhead | End to end packet delay | Percentage of packet loss | Percentage of optimal routes |
|---|---|---|---|
| Ω | D | L | Π |

**Proposition 1. Routing overhead increases with node mobility in both proactive and reactive routing protocols.**

**P1.1** For $M_1>M_2$, $\Omega_P(M_1)>\Omega_P(M_2)$

**P1.2** For $M_1>M_2$, $\Omega_R(M_1)>\Omega_R(M_2)$

**P1.3** For $M>M_{threshold}$, $\Omega_P(M)>\Omega_R(M)\geq 0$

$M_1$ and $M_2$ represent different values for mobility. The derivatives $\mathbf{\Omega}_P'(M)\geq 0$ and $\mathbf{\Omega}_R'(M)\geq 0$ are used to demonstrate that overhead function increases with mobility, and they will be applied for the mathematical analysis in the rest of the chapter.

The routing overhead increases with node mobility due to the extra route discovery transactions generated in reactive protocols and the route updates required in proactive routing protocols. We expect that the routing overhead of proactive routing protocols increases more than the routing overhead of reactive protocols because the route updates need to span all nodes when links break due to mobility. We assume that the routing overhead of reactive routing protocols is lower than the routing overhead of proactive protocols because only the existing routes need to be re-established during a link break.

**Proposition 2. End to end packet delay increases with node mobility in both proactive and reactive routing protocols.**

**P2.1** For $M_1>M_2$, $D_P(M_1)>D_P(M_2)$

**P2.2** For $M_1>M_2$, $D_R(M_1)>D_R(M_2)$

**P2.3** For $M>M_{threshold}$, $D_P(M)>D_R(M)\geq 0$

$M_1$ and $M_2$ represent different values for mobility. The derivatives $D_P'(M)\geq 0$ and $D_R'(M)\geq 0$ are used to demonstrate that delay function increases with mobility, and they will be applied for the mathematical analysis in the rest of the chapter.

In proactive routing protocols, the end to end packet delay increases when there is network congestion because of the increment in the number of transactions required to exchange topology information with all the nodes. The end to end packet delay increases with node mobility in reactive routing protocols because of the increment of route discovery transactions. We expect that the packet delay in reactive routing protocols is lower than in proactive protocols because the route information is fresh since it is acquired right before starting the flow. We assume that the packet delay in proactive routing protocols is higher than in reactive protocols because the routing information may be stale when starting the packet flow, and the link breaks due to mobility create additional traffic increasing the congestion in all nodes.

**Proposition 3. Percentage of packet loss increases with node mobility in both proactive and reactive protocols.**

**P3.1** For $M_1>M_2$, $L_P(M_1)>L_P(M_2)$

**P3.2 F**or $M_1>M_2$, $L_R(M_1)>L_R(M_2)$

**P3.3** For $M>M_{threshold}$, $L_P(M)>L_R(M)>0$

$M_1$ and $M_2$ represent different values for mobility. The derivatives $L_P'(M)\geq 0$ and $L_R'(M)\geq 0$ are used to demonstrate that packet loss function increases with mobility, and they will be applied for the mathematical analysis in the rest of the chapter.

When mobility increases, links are more frequently broken and percentage of packet loss increases. We expect the mobility will increase the link breaks that in proactive protocols will result in additional traffic and congestion in all nodes. The reactive protocols have more fresh routing information when starting the packet flow that will result in lower packet loss than in proactive protocols.

**Proposition 4. Percentage of optimal routes decreases in both proactive and reactive routing protocols when node mobility increases.**

**P4.1 F**or $M_1>M_2$, $\Pi_P(M_1)<\Pi_P(M_2)$

**P4.2 F**or $M_1>M_2$, $\Pi_R(M_1)<\Pi_R(M_2)$

$M_1$ and $M_2$ represent different values for mobility. The derivatives $\Pi_P´(M)\leq 0$ and $\Pi_R´(M)\leq 0$ are used to demonstrate that optimal routes function decreases with mobility, and they will be applied for the mathematical analysis in the rest of the chapter.

When the nodes move new shorter routes may appear and it takes time for a routing protocol to discover those optimal routes. This problem occurs more often when node mobility increases.

**Proposition 5. Percentage of optimal routes obtained with proactive routing protocols is higher than with reactive protocols.**

**P5.1** $\Pi_P(M)>\Pi_R(M)$

The routing protocols obtain the network topology based on periodic routing updates (i.e. proactive) or on demand route discovery (i.e. reactive). The proactive routing protocols apply an additional algorithm over the discovered routes to select the most optimal route (e.g. lower number of hops). As a consequence, proactive routing protocols obtain a higher percentage of optimal routes compared to the routes obtained with reactive routing protocols. When mobility increases, the routes obtained become stale due to frequent link brakes.

**Proposition 6. Routing overhead increases with the number of nodes in both proactive and reactive routing protocols.**

**P6.1** For $N_1>N_2$, $\Omega_P(N_1)>\Omega_P(N_2)$

**P6.2** For $N_1>N_2$, $\Omega_R(N_1)>\Omega_R(N_2)$

$N_1$ and $N_2$ represent different values for the number of nodes. The derivatives $\Omega_P´(N)\geq 0$ and $\Omega_R´(N)\geq 0$ are used to demonstrate that routing overhead function increases with the number of nodes, and they will be applied for the mathematical analysis in the rest of the chapter.

The proactive routing protocols have to share the routing information with all the other nodes in the network, which increases the routing information per node as a function of the total number of nodes in the network. The reactive routing protocols have to increase the TTL in the route request to reach all the nodes in the network. Therefore, when the node density increases the route requests are sent by higher number of nodes but few of the messages are reaching new nodes, thus decreasing the route discovery efficiency.

**Proposition 7. For the same number of nodes and mobility conditions the routing overhead is higher in proactive than in reactive protocols.**

**P7.1** $\Omega_P(M,N)\geq\Omega_R(M,N)$

The routing overhead increases with the number of nodes due to additional topology information required in proactive protocols, and the additional route requests forwarded by each of the intermediate nodes in reactive protocols.

**Proposition 8. End to end packet delay increases with the number of nodes in both proactive and reactive routing protocols.**

**P8.1** For $N_1>N_2$, $D_P(N_1)>D_P(N_2)$

**P8.2** For $N_1>N_2$, $D_R(N_1)>D_R(N_2)$

$N_1$ and $N_2$ represent different values for the number of nodes. The derivatives $D_P´(N) \geq 0$ and $D_R´(N) \geq 0$ are used to demonstrate that delay function increases with the number of nodes, and they will be applied for the mathematical analysis in the rest of the chapter.

In this proposition, N denotes both the density and the number of nodes on the end to end path.

**Proposition 9. Percentage of packet loss increases with the number of nodes in both proactive and reactive routing protocols.**

**P9.1** For $N_1 > N_2$, $L_P(N_1) > L_P(N_2)$

**P9.2** For $N_1 > N_2$, $L_R(N_1) > L_R(N_2)$

$N_1$ and $N_2$ represent different values for the number of nodes. The derivatives $L_P'(N) \geq 0$ and $L_R'(N) \geq 0$ are used to demonstrate that packet loss function increases with the number of nodes, and they will be applied for the mathematical analysis in the rest of the chapter.

When the number of nodes increases, the network gets congested because of the additional signalling, causing an increment of the packet delay and the percentage of packet loss. According to Proposition 1, the routing overhead increases with mobility, therefore the throughput will decrease reducing the available bandwidth and increasing the percentage of packet loss.

**Proposition 10. Percentage of optimal routes obtained with proactive and reactive routing protocols decreases with the number of nodes.**

**P10.1** For $N_1 > N_2$, $\Pi_P(N_1) < \Pi_P(N_2)$

**P10.2** For $N_1 > N_2$, $\Pi_R(N_1) < \Pi_R(N_2)$

$N_1$ and $N_2$ represent different values for the number of nodes. The derivatives $\Pi_P'(N) \leq 0$ and $\Pi_R'(N) \leq 0$ are used to demonstrate that optimal routes function decreases with the number of nodes, and they will be applied for the mathematical analysis in the rest of the chapter.

When calculating the optimal routes, increasing the number of nodes will decrease the efficiency of the protocols because of the additional topology information collected from all the nodes that has to be processed.

## 2.6 Proactive versus Reactive Simulation Comparison

In previous section we have formulated a number of propositions based on our qualitative understanding of the behaviour of ad hoc routing protocols. In this section, we include results from a large set of simulations and in section 2.8 we provide the measurements obtained from our test bed to seek confirmation of the accuracy of our propositions. In order to make the transformation from quantitative numeric results obtained from simulations to qualitative statements we fit the simulation results into parametric equations that minimize approximation error.

The purpose of the parametric equations is not to reflect the behaviours of all Ad hoc networks under certain conditions. However, the goal is to explore the behaviour of Ad hoc networks under different routing protocols qualitatively in order to have a good understanding of the design tradeoffs of routing protocols. Therefore, we use both simulations and measurements to study the behaviour. Based on our own experience, we consider that too many simulation results have been published that fit poorly to the measured behaviour gained from a test bed or a real network. The limitation of measurements, on the other hand, is that generalizing the results is difficult. Therefore, we do not believe it would be possible to propose a grand theory and verify it with the means in our disposal. However, our aim is to improve on routing protocol design and justify design choices without having such a theory by using both measurements and simulations, by explaining the differences between the two and thus verifying our work on a qualitative level.

In this section, simulation results justifying the advantages and drawbacks of the reactive and proactive Ad hoc routing protocols will be presented [15]. The routing protocols comparison has been done using ns-2 simulator [43] version 2.27 with standard IEEE 802.11 MAC protocol, which is used in the simulations and test bed included in this thesis. We also verify some of the propositions introduced in section 2.5

The results are obtained from the average of three simulations rounds performed continuously in order to reduce any possible effect due to initialization process of the simulator. In the simulations we consider the following parameters:

- Simulation area: 1500m x 300m.
- Simulation time: 900 seconds.
- Traffic flows:
  1. Constant Bit Rate (CBR) with UDP transport: 20 IP unidirectional flows.
  2. Traffic with TCP transport: 20 IP unidirectional flows.
- Connection rate: 8 packets/second.
- Packet size: 65 bytes.
- Number of nodes: 50 nodes using random waypoint mobility pattern.
- Pause time between node movements: 0, 30, 60, 120, 300, 600 and 900 seconds.

In the simulations we consider the mobility as the average speed of the node during the simulation.

$M = \frac{M_{\max} t_{moving} + 0 t_{pause}}{t_{simulation}} = \frac{M_{\max} t_{moving}}{t_{simulation}}$ where $M = M_{\max}\big|_{t_{moving} = t_{simulation}}$ and $M = 0\big|_{t_{moving} = 0}$.

We run simulations with the same parameters but using either UDP or TCP as transport protocol for the traffic flows to compare the effect of congestion and reliable traffic control mechanisms.

The literature shows that different mobility patterns affect Ad hoc networks performance results [44]. Ad hoc networks will be deployed under different mobility patterns and the routing protocols have to perform in different environments. Therefore, in the simulations, the nodes follow a different mobility pattern after each waiting time as characterised in the random waypoint model[1] [45].

[1] It has been demonstrated that the random waypoint model is not the most accurate mobility pattern but we will use it for simplicity assuming that it is good enough.

The simulations are made considering that the network is handling the traffic generated by 20 active connections transmitting 8 packets/second. The simulations reflect the performance of Ad hoc networks with real time applications under different mobility conditions and using different routing and transport protocols. The simulations last for 900 seconds, thus a pause time of 900 seconds is equivalent to static nodes that do not move during the simulation.

Both reactive (i.e. AODV, TORA, DSR) and proactive routing protocols (i.e. DSDV, OLSR) are covered in the simulations. The simulation results presented in this section are inaccurate due to the random behaviour of the nodes. Therefore, a deeper analysis will be made extracting from each simulation the associated equation for the most representative reactive (i.e. AODV) and proactive (i.e. OLSR) routing protocols and specific transport protocol (i.e. TCP or UDP).

The simulation results can be associated with an equation that can be linear $f(x) = cx + b$, polynomial $f(x) = b + c_1 x + c_2 x^2 + ... + c_n x^n$, logarithmic $f(x) = c \ln x + b$ or exponential $f(x) = ce^{bx}$. The constants $c$ and $b$ of these equations are adjusted using the r-squared value $r^2 = 1 - \frac{\sum (Y_i - \hat{Y}_i)^2}{\sum Y_i^2 - \frac{(\sum Y_i)^2}{n}}$, where $Y_i$ represents the value obtained in the simulation and $\hat{Y}_i$ represents the estimated value from the associated equation. The r-squared value represents the approximation error, thus it tends to 1 when the values from the simulation and the associated equation match. In following sections each simulation is associated with the equation that provides the lowest approximation error $r^2$.

### 2.6.1 Simulation Results on Mobility

Figure 3 shows the routing overhead generated by reactive and proactive routing protocols during the simulation time versus node mobility with UDP traffic flows.

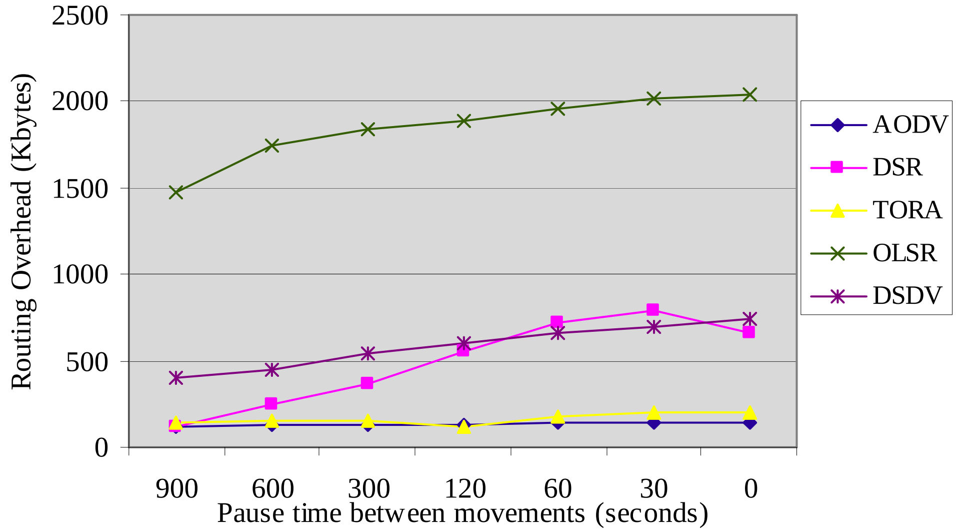


**Figure 3. Routing overhead versus node mobility.**

Proactive protocols have a higher routing overhead than reactive protocols, which can be caused by the additional topology information they exchange. In particular, AODV generates less routing overhead compared to OLSR in similar conditions.

From the different equations that can be associated with the results of the AODV routing overhead with UDP traffic flows, the one with the lowest approximation error $r^2 = 0.976$ is Eq 1.

**Eq 1.** $\Omega_{Ru}(M) = 120.9e^{0.025M}\ (Kbytes)$

The first derivative is $\Omega_{Ru}'(M) = \frac{d(\Omega_{Ru})}{dM} = 3.02e^{0.025M} = \frac{3.02|_{M\to 0}}{+\infty|_{M\to\infty}} \geq 0$, proving P1.2

The associated equation to the OLSR routing overhead simulation results with UDP traffic flows and the lowest approximation error $r^2 = 0.835$ is Eq 2.

**Eq 2.** $\Omega_{Pu}(M) = 1521e^{0.047M}\ (Kbytes)$

The first derivative is $\Omega_{Pu}'(M) = \frac{d(\Omega_{Pu})}{dM} = 71.4e^{0.047M} = \frac{71.4|_{M\to 0}}{+\infty|_{M\to\infty}} \geq 0$, proving P1.1.

Figure 4 shows the routing overhead in AODV and OLSR using a transport protocol that includes reliability and congestion mechanisms such as TCP. The routing overhead increases in both AODV and OLSR compared to UDP traffic flows.

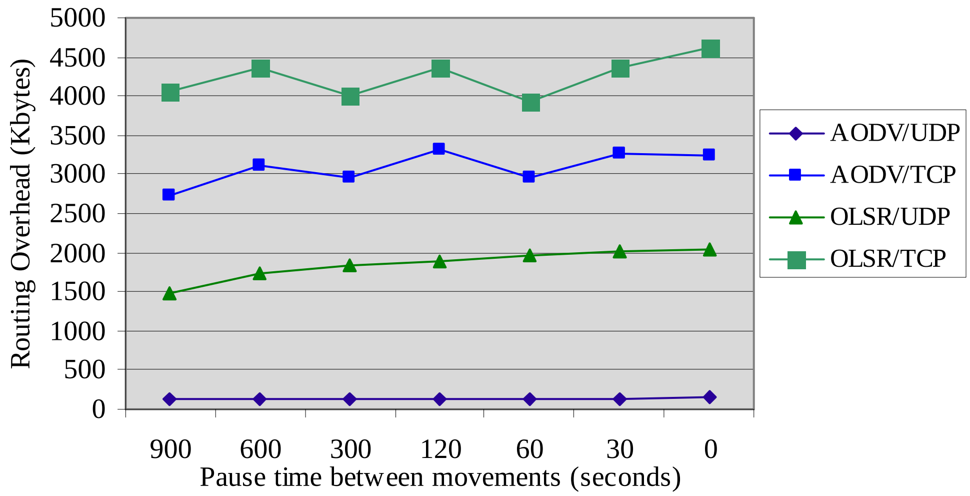


**Figure 4. Routing overhead versus node mobility and transport protocol.**

From the different equations that can be associated with the results of the AODV routing overhead with TCP traffic flows, the one with the lowest approximation error $r^2 = 0.456$ is Eq 3.

**Eq 3.** $\Omega_{Rt}(M) = 2813.1e^{0.022M}\ (Kbytes)$

The first derivative is $\Omega_{Rt}'(M) = \frac{d(\Omega_{Rt})}{dM} = 61.88e^{0.022M} = \begin{matrix} 61.88|_{M\to 0} \\ +\infty|_{M\to\infty} \end{matrix} \geq 0$, proving P1.2.

The associated equation to the OLSR routing overhead simulation results with TCP traffic flows and the lowest approximation error $r^2 = 0.244$ is Eq 4.

**Eq 4.** $\Omega_{Pt}(M) = 4014.7e^{0.013M}\ (Kbytes)$

The first derivative is $\Omega_{Pt}'(M) = \frac{d(\Omega_{Pt})}{dM} = 52.19e^{0.013M} = \begin{matrix} 52.19|_{M\to 0} \\ +\infty|_{M\to\infty} \end{matrix} \geq 0$, proving P1.1 and P1.3.

The associated equations to AODV and OLSR using UDP are more accurate than the same equations when using TCP (i.e. higher r-squared value) and they show that proactive protocols have higher routing overhead than reactive protocols under similar conditions, as stated in P1.3.

Figure 5 shows the end to end packet delay generated by reactive and proactive routing protocols during the simulation time versus node mobility with UDP traffic flows. In high mobility conditions, proactive routing protocols such as OLSR present higher delay than reactive routing protocols as stated in P2.3. In case of low mobility, performance of reactive and proactive routing protocols is similar.

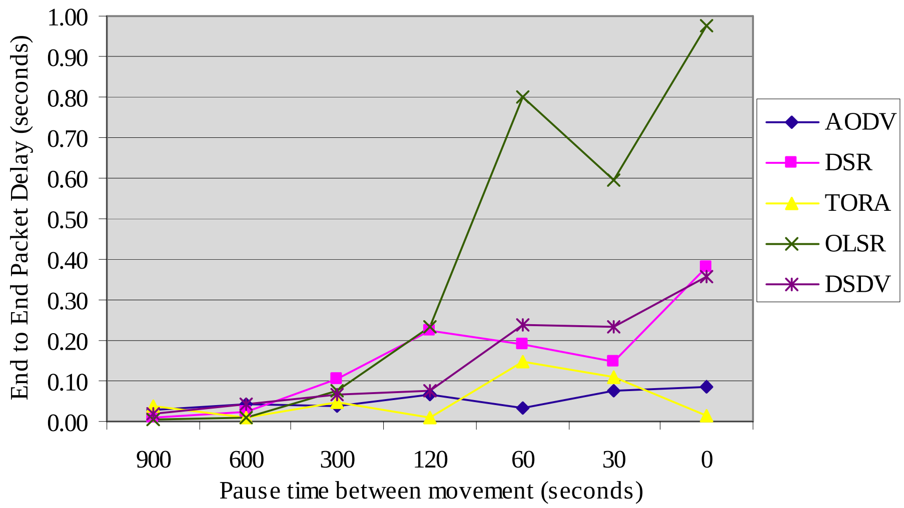


**Figure 5. End to end packet delay versus node mobility.**

Node mobility affects the end to end packet delay because of different reasons such as network congestion and loss of connectivity. Network congestion increases with mobility due to the link breaks that generate new topology updates in proactive protocols, and additional route requests initiated in reactive protocols. The connectivity is immediately re-established after the link break by reactive protocols but the same is performed after a periodic route update in proactive protocols.

The associated equation to the AODV end to end packet delay simulation results with UDP traffic flows and the lowest approximation error $r^2 = 0.625$ is Eq 5.

**Eq 5.** $D_{Ru}(M) = 0.008M + 0.021(s)$

The first derivative is $D_{Ru}'(M) = \frac{d(D_{Ru})}{dM} = 0.008 \geq 0$, proving P2.2.

The associated equation to the OLSR end to end packet delay simulation results with UDP traffic flows and the lowest approximation error $r^2 = 0.851$ is Eq 6.

**Eq 6.** $D_{Pu}(M) = 0.172M - 0.302(s)$

The first derivative is $D_{Pu}'(M) = \frac{d(D_{Pu})}{dM} = 0.172 \geq 0$, proving P2.1**.**

In Eq 6 when M=0 we obtain a negative value for the end to end packet delay $D_{Pu}(0) = -0.302$ representing an approximation error.

Figure 6 shows that the end to end packet delay is reduced using TCP as transport protocol. This can be due to the fact that with TCP both ends maintain a connection state, thus they will notice a link break immediately and either trigger a route update earlier than the normal periodic update, or they will recalculate an alternative route in the routing table. The difference in reactive protocols when using either UDP or TCP is minor because reactive protocols do not maintain routing tables. They do not have alternative routes available to re-route the traffic and they just issue a route request when needed. The reactive protocols have similar behaviour with UDP and TCP because they detect the link break immediately and initiate the route discovery to provide an alternative path.

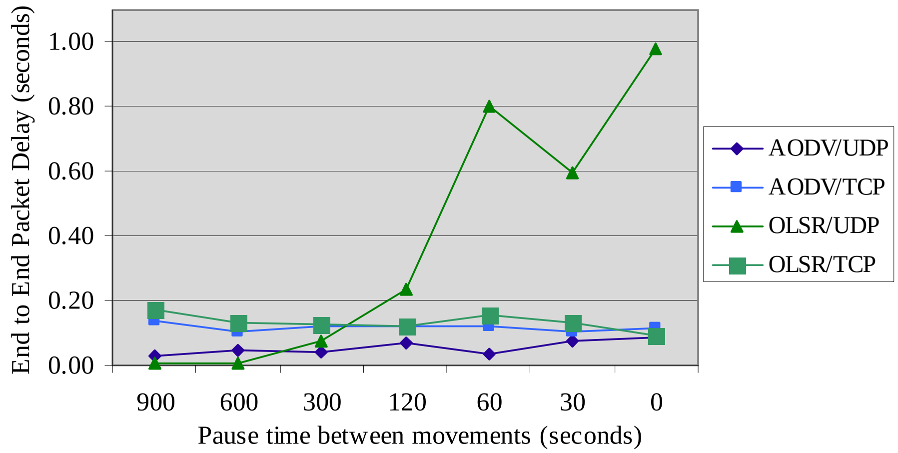


**Figure 6. End to end packet delay versus node mobility and transport protocol.**

The associated equation to the AODV end to end packet delay simulation results with TCP traffic flows and the lowest approximation error $r^2 = 0.26$ is Eq 7.

**Eq 7.** $D_{Rt}(M) = 0.0025M + 0.127(s)$

The first derivative is $D_{Rt}'(M) = \frac{d(D_{Rt})}{dM} = 0.0003 \geq 0$, proving P2.2.

The associated equation to the OLSR end to end packet delay simulation results with TCP traffic flows and the lowest approximation error $r^2 = 0.44$ is Eq 8.

**Eq 8.** $D_{Pt}(M) = 0.0076M + 0.1619(s)$

The first derivative is $D_{Pt}'(M) = \frac{d(D_{Pt})}{dM} = 0.0012 \geq 0$, proving P2.1.

In proactive protocols, the connection control in the traffic flow decreases the delay compared to non reliable connections when using UDP as transport protocol. The accuracy of the associated equations for UDP traffic flows is higher than the equations for TCP flows, but still they show that the end to end packet delay is higher in proactive routing than in reactive routing protocols as stated in P2.3.

Figure 7 shows the percentage of packet loss generated when reactive or proactive routing protocols are used during the simulation time versus node mobility with UDP traffic flows.

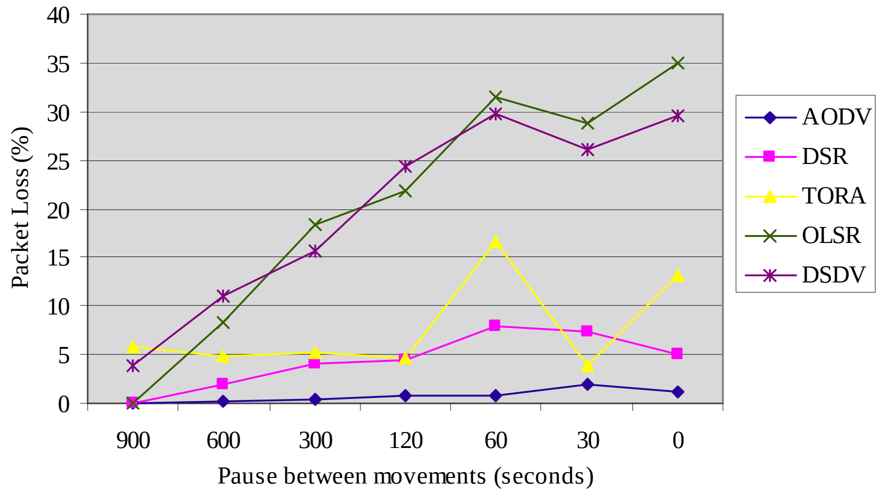


**Figure 7. Percentage of packet loss versus node mobility.**

We measured the packet loss as the percentage of packets that did not reach the destination from the total number of packets sent. The percentage of packet loss is higher in case of proactive routing protocols than in case of reactive routing protocols and increases with mobility as stated in Proposition 3.

The associated equation to the AODV percentage of packet loss simulation results with UDP traffic flows and the lowest approximation error $r^2 = 0.881$ is Eq 9.

**Eq 9.** $L_{Ru}(M) = 0.083e^{0.455M}$ (%)

The first derivative is $L_{Ru}'(M) = \frac{d(L_{Ru})}{dM} = 0.038e^{0.455M} = \frac{0.038|_{M\to 0}}{+\infty|_{M\to\infty}} \geq 0$, proving P3.2.

The associated equation to the OLSR percentage of packet loss simulation results with UDP traffic flows and the lowest approximation error $r^2 = 0.56$ is Eq 10.

**Eq 10.** $L_{Pu}(M) = 0.225e^{0.89M}$ (%)

The first derivative is $L_{Pu}'(M) = \frac{d(L_{Pu})}{dM} = 0.2e^{0.89M} = \begin{matrix} 0.2|_{M\to 0} \\ +\infty|_{M\to\infty} \end{matrix} \geq 0$, proving P3.1.

Figure 8 shows that packet loss is reduced using a transport protocol with connection control in the traffic flows (i.e. TCP).

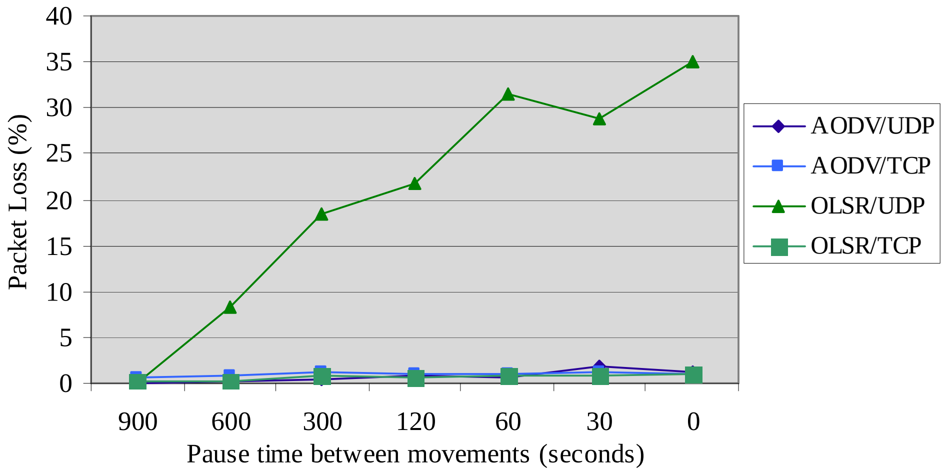


**Figure 8. Percentage of packet loss versus node mobility and transport protocol.**

The associated equation to the AODV end to end packet delay simulation results with TCP traffic flows and the lowest approximation error $r^2 = 0.488$ is Eq 11.

**Eq 11.** $L_{Rt}(M) = 0.773e^{0.062M}$ (%)

The first derivative is $L_{Rt}'(M) = \frac{d(L_{Rt})}{dM} = 0.048e^{0.062M} = \begin{matrix} 0.048|_{M\to 0} \\ +\infty|_{M\to\infty} \end{matrix} \geq 0$, proving P3.2.

The associated equation to the OLSR end to end packet delay simulation results with TCP traffic flows and the lowest approximation error $r^2 = 0.779$ is Eq 12.

**Eq 12.** $L_{Pt}(M) = 0.2418e^{0.221M}$ (%)

The first derivative is $L_{Pt}'(M) = \frac{d(L_{Pt})}{dM} = 0.053e^{0.221M} = \begin{matrix} 0.053|_{M\to 0} \\ +\infty|_{M\to\infty} \end{matrix} \geq 0$, proving P3.1.

TCP includes a connection control mechanism that reduces the end to end packet delay as we can see comparing Eq 6 with Eq 8 and it reduces packet loss as we can deduce from Eq 10 and Eq 12. Lower slopes in Eq 11 than in Eq 12 demonstrate that reactive protocols present shorter end to end packet delay than proactive routing protocols, proving P3.3.

Figure 9 shows the percentage of optimal routes obtained by reactive and proactive routing protocols during the simulation time versus node mobility. Proactive routing protocols perform better than reactive routing protocols when obtaining the optimal routes. Proactive routing protocols maintain the routing information up to date and apply appropriate routing algorithms (e.g. Shortest Path [40]). The percentage of optimal routes decreases in both reactive and proactive protocols with node mobility as stated in Proposition 4.

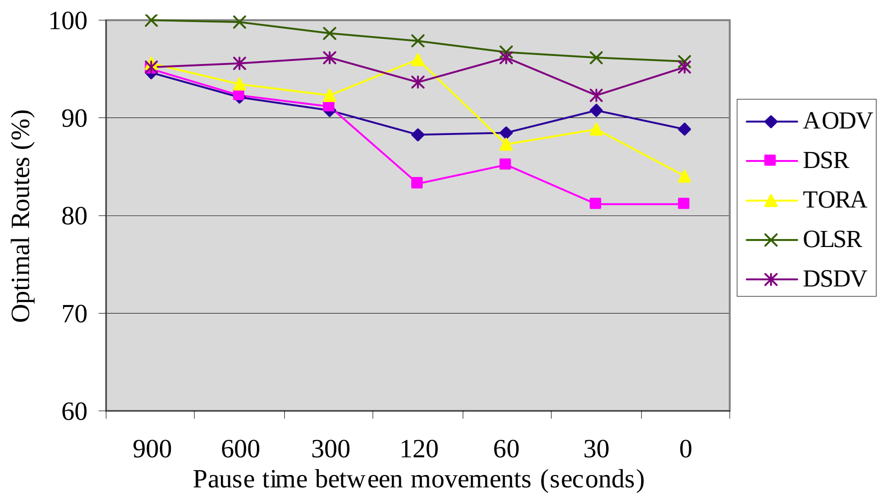


**Figure 9. Percentage of optimal routes versus node mobility.**

The associated equation to the AODV percentage of optimal routes simulation results with UDP traffic flows and the lowest approximation error $r^2 = 0.729$ is Eq 13.

**Eq 13.** $\Pi_{Ru}(M) = 94.028 - 2.864\ln(M)(\%)$

The first derivative is $\Pi_{Ru}'(M) = \frac{d(\Pi_{Ru})}{dM} = -\frac{2.864}{M} = \begin{matrix} -\infty|_{M\to 0} \\ -0|_{M\to\infty} \end{matrix} \leq 0$, proving P4.2.

The associated equation to the OLSR percentage of optimal routes simulation results with UDP traffic flows and the lowest approximation error $r^2 = 0.902$ is Eq 14.

**Eq 14.** $\Pi_{Pu}(M) = 100 - 2.381\ln(M)(\%)$

The first derivative is $\Pi_{Pu}'(M) = \frac{d(\Pi_{Pu})}{dM} = -\frac{2.381}{M} = \begin{matrix} -\infty|_{M\to 0} \\ -0|_{M\to\infty} \end{matrix} \leq 0$, proving P4.1.

Figure 10 shows that the percentage of optimal routes has increased in reactive and proactive routing protocols when using a transport protocol with connection control in the traffic flows such as TCP.

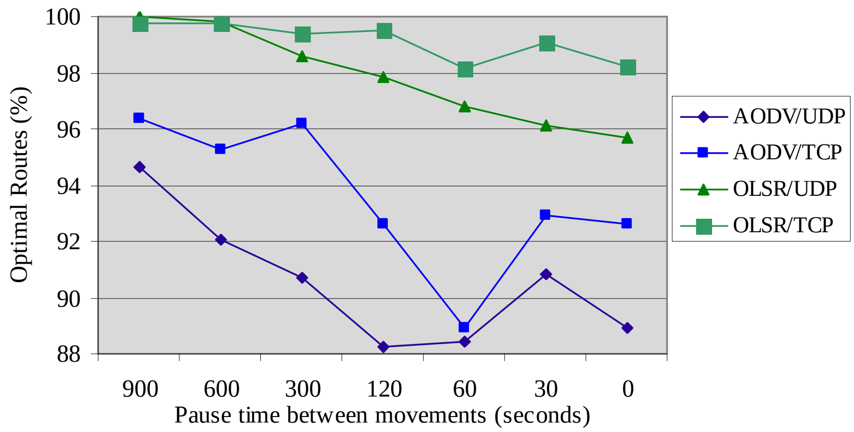


**Figure 10. Percentage of optimal routes versus node mobility and transport protocol.**

The associated equation to the AODV percentage of optimal routes simulation results with TCP traffic flows and the lowest approximation error $r^2 = 0.504$ is Eq 15.

**Eq 15.** $\Pi_{Rt}(M) = 96.85 - 2.708\ln(M)(\%)$

The first derivative is $\Pi_{Rt}'(M) = \frac{d(\Pi_{Rt})}{dM} = -\frac{2.708}{M} = \begin{matrix} -\infty|_{M\to 0} \\ -0|_{M\to\infty} \end{matrix} \leq 0$, proving P4.2.

The associated equation to the OLSR percentage of optimal routes simulation results with TCP traffic flows and the lowest approximation error $r^2 = 0.591$ is Eq 16.

**Eq 16.** $\Pi_{Pt}(M) = 100 - 0.7653\ln(M)(\%)$

The first derivative is $\Pi_{Pt}'(M) = \frac{d(\Pi_{Pt})}{dM} = -\frac{0.7653}{M} = \begin{matrix} -\infty|_{M\to 0} \\ -0|_{M\to\infty} \end{matrix} \leq 0$, proving P4.1.

The associated equations show that 100% of the routes obtained with the proactive protocol can be optimal in case of zero node mobility compared to the case of reactive protocol where with similar conditions only 94% of the routes obtained are optimal, which proves Proposition 5. We can see that using a connection control transport protocol increases the percentage of optimal routes in reactive ( Eq 13, Eq 15) and proactive (Eq 14, Eq 16) protocols. When the connection control detects a link break, it triggers either a route recalculation in proactive protocols or a route discovery in reactive protocols. However, proactive protocols obtain a higher percentage of optimal routes than reactive protocols as stated in P5.1.

### 2.6.2 Simulation Results on Scalability

We have verified some of the propositions based on the results from the simulations but the scalability effect on the routing protocols when increasing the number or density of nodes remains to be demonstrated. The simulator has some limitations in terms of number of nodes (i.e. max number of nodes is 100). Therefore, in order to study the impact on the performance results when increasing the number of nodes, new simulations were performed with 25, 50 and 100 nodes keeping the same value for the rest of the parameters. We select TCP as the transport protocol for these simulations because it provides similar results for proactive and reactive protocols regarding end to end packet delay and packet loss. However, we have to consider that the connection control mechanism in TCP creates additional overhead.

Simulation results presented in Figure 11 show that the routing overhead increases with the number of nodes in both proactive and reactive routing protocols as stated in Proposition 6.

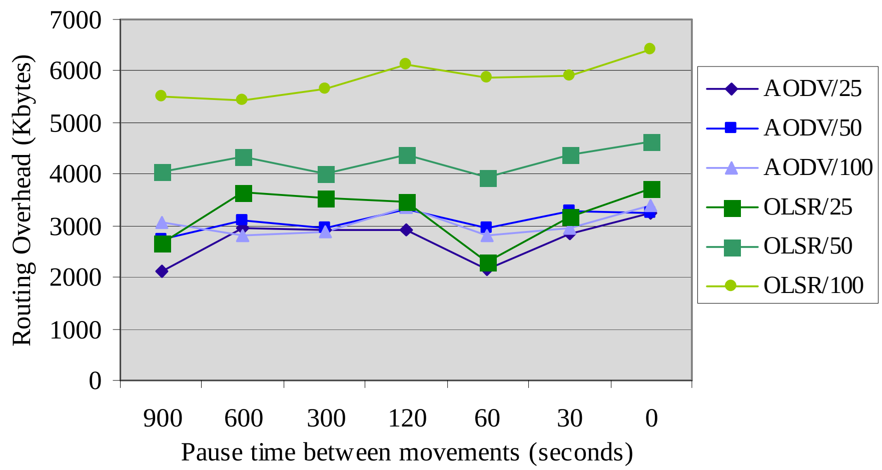


**Figure 11. Routing overhead in reactive and proactive routing with 25, 50 and 100 nodes.**

The associated equations with the AODV routing overhead simulation results for the different number of nodes with TCP traffic flows and the lowest approximation error $r^2 = 0.45$ are Eq 17, Eq 18 and Eq 19.

**Eq 17.** $\Omega_{Rt}(M, N = 25) = 2378.2e^{0.032M}\ (Kbytes)$

**Eq 18.** $\Omega_{Rt}(M, N = 50) = 2813.1e^{0.022M}\ (Kbytes)$

**Eq 19.** $\Omega_{Rt}(M, N = 100) = 2880.2e^{0.013M}\ (Kbytes)$

In reactive routing protocols the routing overhead increases with the number of nodes as stated in Proposition 6**.** The simulation results could be associated with linear equations but it has a higher approximation error than the exponential equation. A major increase of the routing overhead takes place when incrementing from 25 (i.e. 2378.2Kbytes) to 50 (i.e. 2813.1Kbytes) nodes, while the values for 50 and 100 nodes are similar.

Next, we define a generic equation that includes both mobility and the number of nodes as variables. We take the equations obtained from simulations for 25, 50 and 100 nodes, with mobility as the only variable, and we associate them with an equation that can be linear, polynomial, logarithmic or exponential depending on the associated error. The generic equation associated to the AODV routing overhead with TCP traffic flows is drawn up taking the equations Eq 17, Eq 18, Eq 19 and obtaining the associated equation for the bases (i.e. 2378.2, 2813.1 and 2880.2) and the slope factors (i.e. 0.032, 0.022 and 0.013) with the lowest approximation error resulting in Eq 20.

**Eq 20.** $\Omega_{Rt}(M,N) = (2188 + 251N)e^{(0.04+0.009N)M}\ (Kbytes)$

When comparing Eq 18 and Eq 1. $\Omega_{Ru}(M) = 120.9e^{0.025M}\ (Kbytes)$ obtained to model the routing overhead for 50 nodes using TCP and UDP respectively, we see that the results are different. This is due to the additional overhead in TCP compared to UDP. To model the routing overhead using UDP considering as variables the mobility and the number of nodes, we take Eq 20 and Eq 18 as reference to estimate the generic equation associated to the AODV routing overhead with UDP. The base of the equation with TCP changes from 2188 in Eq 18 to 2813.1 in Eq 20 which means an increment of 28.57% so we can estimate that for UDP it will be $\Omega_{Ru}(M,N) = 155.4e^{0.025M}\ (Kbytes)$. The slope of the equation changes from 0.022 in Eq 18 to 0.04 in Eq 20 which means an increment of 81.82% so we estimate that for UDP it will be $\Omega_{Ru}(M,N) = 155.4e^{0.045M}\ (Kbytes)$. The slope we obtain with UDP is similar to the one in Eq 20 so we could extend the factor associated with N for UDP with the same value for TCP as in Eq 20. We estimate that for UDP the final slope is $\Omega_{Ru}(M,N) = 155.4e^{(0.045+0.009N)M}\ (Kbytes)$. The base of Eq 18 for TCP is 2813.1 which is 23.27 times bigger than the base of Eq 1 for UDP. Therefore, we use the factor associated with N for TCP in Eq 20 as reference (i.e. 120N) to estimate a similar value for UDP. Thus, we model the routing overhead for UDP taking Eq 18, Eq 20 and Eq 1 as reference, resulting in Eq 21 which represents the AODV routing overhead generic equation with UDP traffic.

**Eq 21.** $\Omega_{Ru}(M,N) = (155.4 + 5.1N)e^{(0.045+0.009N)M}\ (Kbytes)$

The associated equations with the OLSR routing overhead simulation results for the different number of nodes with TCP traffic flows and the lowest approximation error $r^2 = 0.24$ are Eq 22, Eq 23 and Eq 24.

**Eq 22.** $\Omega_{Pt}(M, N = 25) = 3027.7e^{0.012M}\ (Kbytes)$

**Eq 23.** $\Omega_{Pt}(M, N = 50) = 4014.7e^{0.013M}\ (Kbytes)$

**Eq 24.** $\Omega_{Pt}(M, N = 100) = 5297.4e^{0.024M}\ (Kbytes)$

In proactive routing protocols the routing overhead significantly increases with the number of nodes as stated in Proposition 6. From the associated equations, the routing overhead value roughly increases by 1000Kbytes when doubling the number of nodes. The slope factor doubles when the number of nodes increases from 25 to 100.

The generic equation associated with the OLSR routing overhead with TCP traffic flows is drawn up taking the equations Eq 22, Eq 23, Eq 24 and obtaining the associated equation with the lowest approximation error resulting in Eq 25.

**Eq 25.** $\Omega_{Pt}(M,N) = (1843 + 1134N)e^{(0.0037+0.0065N)M} \approx (1850 + 1130N)e^{(0.004+0.0065N)M}\ (Kbytes)$

When comparing Eq 23 and Eq 2. $\Omega_{Pu}(M) = 1521e^{0.047M}\ (Kbytes)$ obtained to model the routing overhead for 50 nodes using TCP and UDP respectively the results are different. Both the base and slope factors are 3 times lower in UDP than in TCP. Thus, we model the routing overhead using UDP taking Eq 23, Eq 25 and Eq 2 resulting in Eq 26 which represents the OLSR routing overhead generic equation with UDP traffic.

**Eq 26.** $\Omega_{Pu}(M,N) = (615 + 375N)e^{(0.001+0.002N)M}\ (Kbytes)$

Therefore, the routing overhead increases with the number of nodes as stated in Proposition 6 and the proactive routing protocols present higher overhead than reactive protocols as stated in P7.1. Increasing the number of nodes affects more on the proactive protocols routing overhead while increasing the node mobility affects more on the reactive protocols routing overhead. For this reason, proactive routing protocols are not scalable in large Ad hoc networks.

Figure 12 shows that the end to end packet delay is similar in case of reactive and proactive routing protocols when the increase in the number of nodes is small (i.e. ±0.02s end to end packet delay variation when 25≤N≤50). When increasing the number of nodes (i.e. N=100) the end to end packet delay is higher in proactive than in reactive routing protocols.

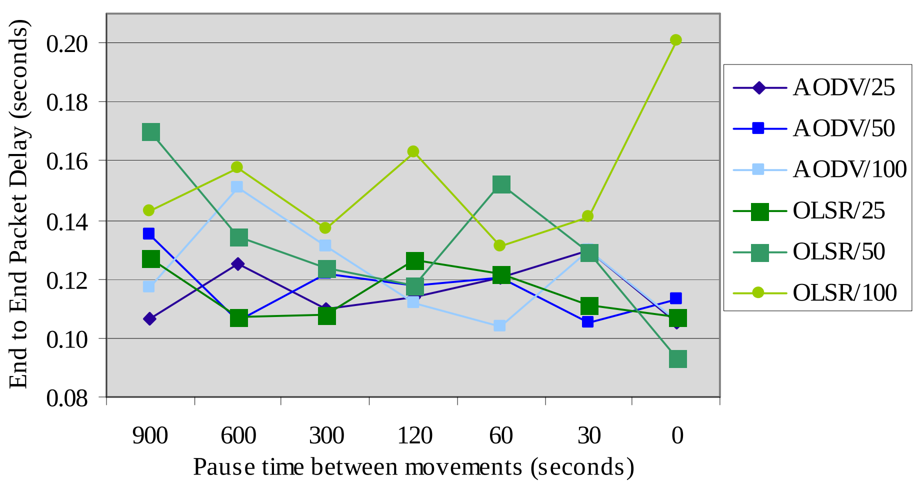


**Figure 12. End to end packet delay in reactive and proactive routing with 25, 50 and 100 nodes.**

The associated equations with the AODV end to end packet delay simulation results for the different number of nodes with TCP traffic flows and the lowest approximation error $r^2$ =0.41 are Eq 27, Eq 28 and Eq 29.

**Eq 27.** $D_{Rt}(M, N = 25) = 0.001M + 0.114(s)$

**Eq 28.** $D_{Rt}(M, N = 50) = 0.0025M + 0.127(s)$

**Eq 29.** $D_{Rt}(M, N = 100) = 0.0037M + 0.136(s)$

The end to end packet delay is almost constant (i.e. between 114-136ms for M=0) for reactive routing despite increasing of the number of nodes when mobility is zero. However, the end to end packet delay increases with the number of nodes as stated in Proposition 8.

The generic equation associated with the AODV end to end packet delay with TCP traffic flows is drawn up taking the equations Eq 27, Eq 28 and Eq 29 and obtaining the associated equation with the lowest approximation error resulting in Eq 30.

**Eq 30.** $D_{Rt}(M, N) = (0.0014N)M + 0.1 + 0.011N(s)$

When comparing Eq 28 and Eq 5. $D_{Ru}(M) = 0.008M + 0.021(s)$ obtained to model the end to end packet delay for 50 nodes using TCP and UDP respectively the results are different. The values obtained with UDP in Eq 5 are optimistic compared to Eq 28, giving an end to end packet delay value of 21ms when mobility is zero. The latest simulations using TCP provide more realistic values despite of the higher approximation error. Thus, we model the end to end packet delay using the same Eq 30 which represents the AODV end to end packet delay generic equation with UDP and TCP traffic:

$D_{Ru}(M, N) = D_{Rt}(M, N) = D_R(M, N) = (0.0014N)M + 0.1 + 0.011N(s)$

The associated equations with the OLSR end to end packet delay simulation results for the different number of nodes with TCP traffic flows and the lowest approximation error $r^2 = 0.43$ are Eq 31, Eq 32 and Eq 33.

**Eq 31.** $D_{Pt}(M, N = 25) = 0.001M + 0.121(s)$

**Eq 32.** $D_{Pt}(M, N = 50) = 0.0076M + 0.161(s)$

**Eq 33.** $D_{Pt}(M, N = 100) = 0.0048M + 0.134(s)$

From the equations Eq 27, Eq 28, Eq 29, Eq 31, Eq 32 and Eq 33 we observe that proactive and reactive protocols have similar end to end packet delay (i.e. between 114-136ms delay for mobility zero), which contradicts P2.3. However, when the number of nodes is high N=100, the end to end packet delay in proactive routing protocols show more dependency with the mobility (i.e. mobility incremental factor of 0.003) than in reactive routing protocols (i.e. mobility incremental factor of 0.001).

The generic equation associated with the OLSR end to end packet delay with TCP traffic flows is drawn up taking the equations Eq 31, Eq 32 and Eq 33 and obtaining the associated equation with the lowest approximation error ($r^2 = 0.43$) resulting in Eq 34.

**Eq 34.** $D_{Pt}(M,N) = (0.0025N)M + 0.113 + 0.07N(s)$

When comparing Eq 32 and Eq 6. $D_{Pu}(M) = 0.172M - 0.302(s)$ obtained to model the end to end packet delay for 50 nodes using TCP and UDP respectively the results are considerable different because UDP does not provide connection failure detection so the routing protocol does not trigger a route update early enough. The latest simulations provide more realistic values despite of the higher approximation error. Thus, we model the end to end packet delay using the same Eq 34 which represents the OLSR end to end packet delay generic equation with UDP and TCP traffic.

$$D_{Pu}(M,N) = D_{Pt}(M,N) = D_P(M,N) = (0.0025N)M + 0.113 + 0.07N(s)$$

Reactive and proactive routing protocols are not highly affected by the number of nodes from the end to end packet delay point of view. Proactive protocols present scalability issues when the number of nodes is high due to network congestion because of the additional routing overhead as stated in Proposition 7.

Figure 13 shows that the percentage of packet loss increases with the mobility and the number of nodes in both reactive and proactive routing protocols. Left corner of the Figure 13 shows that the percentage of packet loss in static conditions (i.e. the maximum mobility is represented in Figure 13 with 0 pause time between movements) and for a small number or density of nodes (i.e. N=25) is the same for reactive and proactive routing protocols. Moreover, when the number of nodes increases (i.e. 50≤N≤100), the percentage of packet loss is higher for reactive routing protocols than for proactive routing protocols. This contradicts P3.3 which only stands in punctual cases with high mobility and number or density of nodes (i.e. OLSR with N=100 and 30 pause time). This means that regarding the percentage of packet loss reactive routing protocols are less scalable than proactive routing protocols.

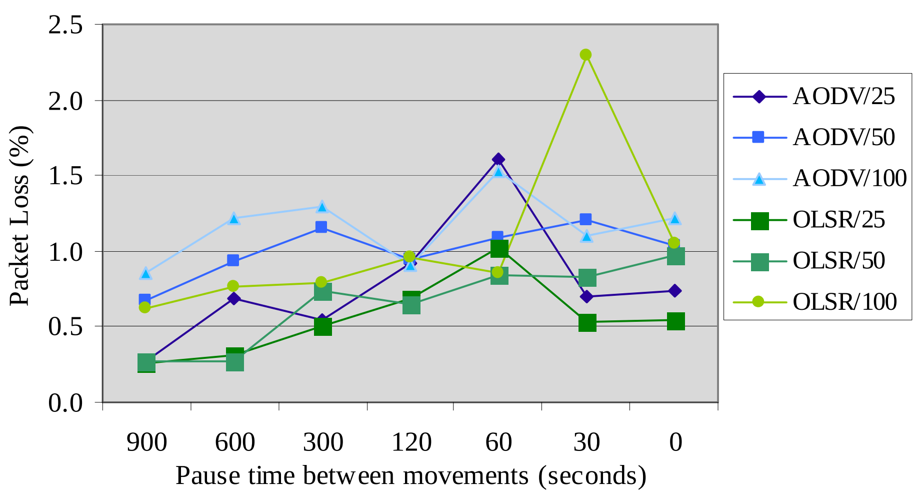


**Figure 13. Percentage of packet loss in reactive and proactive routing with 25, 50 and 100 nodes.**

The associated equations with the AODV percentage of packet loss simulation results for the different number of nodes with TCP flows and the lowest approximation error $r^2 = 0.48$ are Eq 35, Eq 36 and Eq 37.

**Eq 35.** $L_{Rt}(M, N = 25) = 0.38e^{0.146M}$ (%)

**Eq 36.** $L_{Rt}(M, N = 50) = 0.77e^{0.062M}$ (%)

**Eq 37.** $L_{Rt}(M, N = 100) = 0.98e^{0.036M}$ (%)

The equations Eq 35, Eq 36 and Eq 37 show that the percentage of packet loss is low in reactive protocols but it increases with the number of nodes as stated in Proposition 9.

The generic equation associated with the AODV percentage of packet loss with TCP traffic flows is drawn up taking the equations Eq 35, Eq 36, Eq 37 and obtaining the associated equation for the bases (i.e. 0.38, 0.77 and 0.98) and the slope factors (i.e. 0.146, 0.062 and 0.036) with the lowest approximation error ($r^2 = 0.48$) resulting in Eq 38.

**Eq 38.** $L_{Rt}(M,N) = (0.11 + 0.301N)e^{(0.192-0.05N)M}$ (%)

When comparing Eq 36 and Eq 9. $L_{Ru}(M) = 0.083e^{0.455M}$ (%) obtained to model the packet loss for 50 nodes using TCP and UDP respectively the results are roughly 10 times lower with UDP than with TCP traffic. However, the dependency with the mobility is higher in UDP than in TCP as represented by the slope factor 0.445 in UDP versus 0.062 in TCP which is 7 times lower. Thus, we model the AODV packet loss with UDP traffic using Eq 38 as reference resulting in Eq 39 which represents the AODV packet loss generic equation with UDP traffic.

**Eq 39.** $L_{Ru}(M,N) \approx (0.01 + 0.03N)e^{(1.34+0.35N)M}$ (%)

The associated equations with the OLSR percentage of packet loss simulation results for the different number of nodes with TCP traffic flows and the lowest approximation error $r^2 = 0.77$ are Eq 40, Eq 41 and Eq 42.

**Eq 40.** $L_{Pt}(M, N = 25) = 0.283e^{0.143M}$ (%)

**Eq 41.** $L_{Pt}(M, N = 50) = 0.241e^{0.221M}$ (%)

**Eq 42.** $L_{Pt}(M, N = 100) = 0.551e^{0.137M}$ (%)

The generic equation associated with the OLSR percentage of packet loss with TCP traffic flows is drawn up taking the equations Eq 40, Eq 41 and Eq 42 and obtaining the associated equation with the lowest approximation error resulting in Eq 43.

**Eq 43.** $L_{Pt}(M,N) = (0.091 + 0.134N)e^{(0.174-0.003N))M}$ (%)

When comparing Eq 41 and Eq 10. $L_{Pu}(M) = 0.225e^{0.89M}$ (%) obtained to model the packet loss for 50 nodes using TCP and UDP respectively, the results show a major difference in the slope factor. However, assuming the inaccuracy of the simulations and the associated approximation error $r^2 = 0.77$ we can still use those results as reference. Thus, we model the OLSR packet loss with UDP traffic flows using Eq 43 as reference resulting in Eq 44 which represents the generic equation associated to the OLSR packet loss.

**Eq 44.** $L_{Pu}(M,N) = (0.09 + 0.13N)e^{(0.69+0.012N)M}$ (%)

Figure 14 shows that the percentage of optimal routes obtained with reactive and proactive routing protocols with TCP traffic decreases with the number of nodes as stated in Proposition 10.

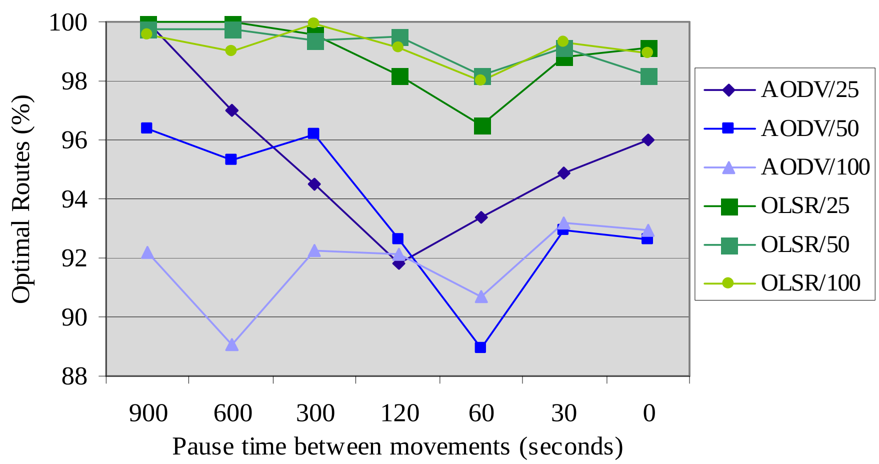


**Figure 14. Percentage of optimal routes in proactive and reactive routing with 25, 50 and 100 nodes.**

Proactive routing protocols exchange topology information periodically and can implement different algorithms to optimise the routes. The reactive routing protocols implement route optimisation during the route request based on the number of hops and sequence numbers to avoid loops.

The associated equations with the AODV percentage of optimal routes simulation results for the different number of nodes with TCP traffic flows and the lowest approximation error $r^2 = 0.504$ are Eq 45, Eq 46 and Eq 47.

**Eq 45.** $\Pi_{Rt}(M, N = 25) = 98.6 - 2.69\ln(M)(\%)$

**Eq 46.** $\Pi_{Rt}(M, N = 50) = 96.8 - 2.7\ln(M)(\%)$

**Eq 47.** $\Pi_{Rt}(M, N = 100) = 90.8 - 0.3\ln(M)(\%)$

The generic equation associated with the AODV percentage of optimal routes with TCP traffic flows is drawn up taking the equations Eq 45, Eq 46 and Eq 47 and obtaining the associated equation with the lowest approximation error resulting in Eq 48.

**Eq 48.** $\Pi_{Rt}(M, N) = (103 - 3.9N) - (4.2 - 1.2N)\ln(M)(\%)$

When comparing Eq 46 and Eq 13. $\Pi_{Ru}(M) = 94.028 - 2.864\ln(M)(\%)$ obtained to model the percentage of optimal routes for 50 nodes using TCP and UDP respectively, the results are similar due to the fact that the transport protocols used for the traffic flows do not affect the obtaining of optimal routes. Moreover, in both cases the approximation error is similar $r^2 = 0.729$ and $r^2 = 0.504$. Thus, we model the AODV percentage of optimal routes with UDP traffic flows using Eq 48 which represents the generic equation associated with the AODV percentage of optimal routes.

$$\Pi_{Ru}(M, N) = \Pi_{Rt}(M, N) = \Pi_{R}(M, N) = (103 - 3.9N) - (4.2 - 1.2N)\ln(M)(\%)$$

The associated equations with the OLSR percentage of optimal routes simulation results for the different number of nodes with TCP traffic flows and the lowest approximation error $r^2 = 0.61$ are Eq 49, Eq 50 and Eq 51.

**Eq 49.** $\Pi_{Pt}(M, N = 25) = 100 - 1.04\ln(M)(\%)$

**Eq 50.** $\Pi_{Pt}(M, N = 50) = 100 - 0.76\ln(M)(\%)$

**Eq 51.** $\Pi_{Pt}(M, N = 100) = 99.6 - 0.36\ln(M)(\%)$

The generic equation associated with the OLSR percentage of optimal routes with TCP traffic flows is drawn up taking the equations Eq 49, Eq 50 and Eq 51 and obtaining the associated equation with the lowest approximation error resulting in Eq 52.

**Eq 52.** $\Pi_{Pt}(M, N) = (98.6 - 0.13N)\ln(M)(\%)$

When comparing Eq 50 and Eq 14. $\Pi_{Pu}(M) = 100 - 2.381\ln(M)(\%)$ obtained to model the percentage of optimal routes for 50 nodes using TCP and UDP respectively the results show that the logarithmic factors have a difference of 3 times lower in TCP than UDP. However, we model the OLSR percentage of optimal routes taking the more optimistic equation with the lower logarithmic factor and using Eq 52 to represent the generic equation associated with the OLSR percentage of optimal routes.

$$\Pi_{Pu}(M, N) = \Pi_{Pt}(M, N) = \Pi_{P}(M, N) = (98.6 - 0.13N)\ln(M)(\%)$$

In reactive protocols the percentage of optimal routes decreases with the number of nodes while in proactive protocols the impact of the number of nodes is low. Therefore, when obtaining optimal routes, the reactive routing protocols are not scalable.

### 2.6.3 Complexity in Reactive and Proactive Routing Protocols

Table 3 compares reactive and proactive protocols in terms of complexity. The *storage complexity* indicates the size of the routing table required by each protocol. The *communication complexity* indicates the processing resources required to find routes or perform a route update operation. N denotes the number or density of nodes in the Ad hoc network, and complexity is represented with the big-O notation.

**Table 3. Comparative of reactive and proactive routing complexity.**

| | **Reactive Routing** | | **Proactive Routing** | | |
|---|---|---|---|---|---|
| | **AODV** | **DSR** | **OLSR** | **TORA** | **DSDV** |
| **Storage Complexity** | O(e)[1] | O(e) | O(N)[2] | O(N) | O(N) |
| **Communication Complexity** | O(2N)[3] | O(2N) | O(N)[4] | O(N) | O(N) |

1 Requires maintaining in the cache only the most recently used routes.
2 Requires maintaining tables with entries for all the nodes in the network.
3 Requires additional route discovery and maintenance that increases with high mobility.
4 Routing information is periodically maintained up to date in all the nodes.

## 2.7 Ad hoc Routing Protocols Simulation Conclusions

The reactive routing protocols under analysis have clear drawbacks such as the excessive flooding traffic in the route discovery and the route acquisition delay. When the network is congested, the routing information is lost and a consecutive set of control packets are issued to re-establish the links, increasing the routing latency (i.e. time the routing protocol requires for obtaining the route to the destination node) and percentage of packet loss. If the Hello messages are not received, then error requests are issued and new route requests are sent to re-establish the link. Thus, the reactive protocols do not scale when the load and node density increase. Moreover, the reactive routing protocols do not have knowledge about the QoS in the path before the route is established and the routes are not optimised.

The reactive routing protocols suffer from high routing latency and percentage of packet loss, which increase with mobility and large networks. The percentage of optimal routes calculated with reactive protocols is lower than in proactive protocols and it decreases in large networks. An advantage of reactive protocols like AODV is that they maintain only the active routes in the routing table, which minimizes the memory required in the node. Moreover, the protocol itself is simple so the computational requirements are minimal, extending the lifetime of the node in the Ad hoc network. The routing overhead is equivalent to additional packet processing, thus reactive protocols will have lower power consumption than proactive protocols. In simulations with a small number of nodes, AODV has lower percentage of packet loss than OLSR. Therefore in networks with light traffic and low mobility reactive protocols are scalable because of the small bandwidth and storage requirements.

The proactive routing protocols under analysis maintain topology information up to date with periodic update messages. The proactive routing protocols minimize the route discovery delay, which minimizes the percentage of packet loss since the routes are known in advance and no additional routing overhead and processing are required. However, under high mobility conditions more and more routes established based on the previous periodic update become stale leading to an increased percentage of packet loss.

The proactive routing protocols have low routing latency since all the routes are available immediately even in large networks. The proactive routing protocols calculate the most optimal routes since they apply hop count based routing algorithms. The proactive routing protocols have higher percentage of packet loss than reactive protocols in networks with reduced number of nodes and high mobility as depicted in Figure 7. However, if the transport protocol includes connection control mechanism (i.e. TCP) that detects link breaks and triggers route update or route recalculation, then proactive protocols present lower percentage of packet loss than reactive protocols as depicted in Figure 13.

A drawback of proactive routing protocols is that they require a constant bandwidth and cause a processing overhead to maintain the routing information up to date. This overhead increases with the number of nodes and mobility since the updates have to be more frequent to maintain accurate routing information. The proactive routing protocols have lower routing latency but they do not react quickly enough to topology changes. The proactive routing protocols have been enhanced towards hybrid and hierarchical solutions to deal with this scalability problem in Ad hoc networks. OLSR reduces the control and processing overhead by selecting some nodes (i.e. Multipoint Relay nodes) within the network to maintain the routing information. The link information updates are propagated between MPR nodes only, reliving the rest of the nodes from participating in the topology maintenance. Other optimizations consist of exchanging only the differential updates, implementing hybrid solutions such as ZRP [41] that combines reactive and proactive routing protocols or routing protocols that use the nodes location data such as LAR [42].

In order to analyse the performance of the hybrid protocols versus reactive and proactive, we run additional simulations in the ns-2 with similar parameters.

- Simulation area: 1500m x 300m.
- Transmitter range: 250m and 2Mbit bandwidth.
- Simulation time: 900 seconds.
- Constant Bit Rate (CBR) traffic with UDP transport: 15 IP unidirectional connections.
- Connection rate: 5 packets/second.
- Packet size: 65 bytes.
- Number of nodes: 50 nodes using random waypoint mobility pattern.
- Pause time between node movements: 0, 30, 60, 120, 300, 600 and 900 seconds.

Figure 15 and Figure 16 show the results of the additional simulations run including hybrid routing.

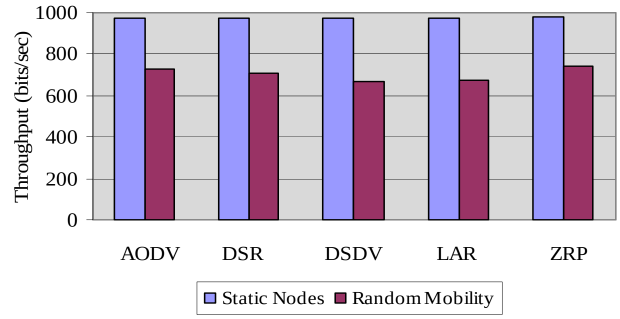


**Figure 15. Throughput versus mobility in reactive, proactive and hybrid routing.**

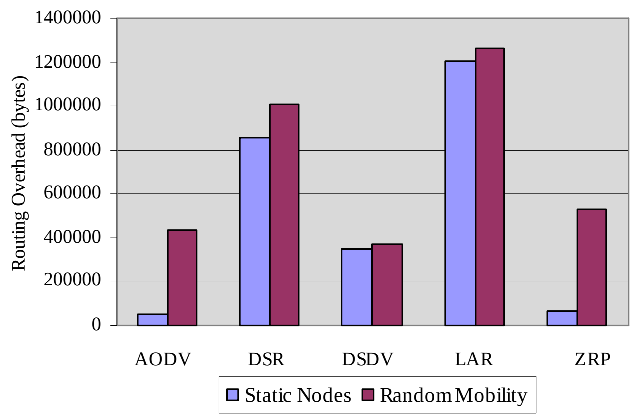


**Figure 16. Routing overhead versus mobility in reactive, proactive and hybrid routing.**

Figure 15 and Figure 16 show the throughput and routing overhead for AODV, DSR, DSDV, LAR and ZRP, comparing two scenarios; zero node mobility and random pause time (i.e. static nodes and random mobility). Mobility affects similarly the throughput of the different routing protocols while the routing overhead is different for both static and mobile nodes. The simulations have been executed for ZRP with the radius of 1 hop and they show the same throughput results as for AODV. If we extend the ZRP radius to several hops, where proactive routing is used, then it will have a similar behaviour to DSDV where the routing overhead is not affected by mobility. The routing overhead with static nodes is the same for AODV and ZRP but it is 15% higher for ZRP with random mobility. LAR introduces the highest routing overhead for the same mobility conditions.

In addition to the hybrid routing protocols such as ZRP and LAR, other alternatives have been proposed to improve the reaction time to link breaks of the proactive routing protocols. One of them is a cross-layer architecture to receive information directly from the link layer in order to react quickly to topology changes when route breaks happen [46]. Despite of this when the network size increases, the bandwidth and processing overhead can still reach limits that cannot be afforded by Ad hoc nodes. Another alternative consists of moving from flat to a more scalable hierarchical routing as proposed in the Fuzzy Sighted Link State (FSLS) routing [47]. FSLS defines a multilevel routing update hierarchy where each level has a different routing packet size and frequency of the routing updates. FSLS minimizes the flooding traffic but increases the complexity when defining levels with different updates frequency. In this thesis we will analyse a third alternative, which consists of a new hybrid routing approach based on AODV. AODV is extended with scalability optimizations in order to reduce the routing latency, the percentage of packet loss and increase the routing efficiency when mobility, the number of nodes or the network size increase.

## 2.8 Ad hoc Routing Protocols Test Bed

The goal of this section is to verify that simulations results are aligned with the values obtained from real Ad hoc networks. The simulations results highlight the overall performance results but they do not reflect the requirements of applications in real Ad hoc networks, or they may differ from results in real devices with limited resources. The simulations provide Ad hoc networks performance results considering a wide range in the variation of parameters such as node density and node mobility. A small-scale experimental Ad hoc network introduces new parameters such as number of hops and route discovery latency that affect the performance. Therefore, in order to verify the accuracy of the simulations and measure the effect of those new parameters, we run a set of tests with real Ad hoc nodes, different routing protocols and a real time VoIP application. The tests were carried out using different devices and in various locations to avoid any bias by environmental factors.

Figure 17 shows the Ad hoc routing framework implemented to build up the test bed. It is a software package with several modules implementing different routing protocols (e.g. AODV, OLSR). A common module allows the different protocols running simultaneously in the node to store and access the same routing information. The framework also includes a real time VoIP application.

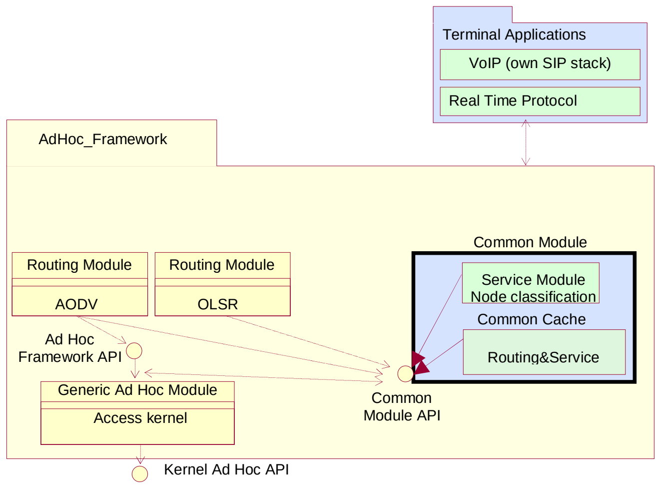


**Figure 17. Ad hoc routing framework.**

### 2.8.1 Testing a Real Time Voice over IP Application

This section analyses the Ad hoc test bed results for an application with real time requirements like Voice over IP (VoIP). The selected traffic with a Constant Bit Rate (CBR) of 15packets/second over UDP used in the simulations is similar to real time VoIP sessions transmitting 20ms voice packets encapsulated with GSM codec [48] and using Real Time Protocol (RTP) [49] protocol over UDP as represented in Figure 18.

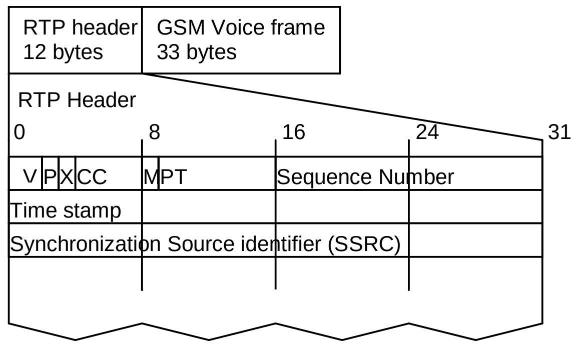


**Figure 18. VoIP packet structure.**

IP protocol offers a best-effort approach where the packets can be lost, delivered with different delay, out of order, corrupted or duplicated. RTP provides packet sequence order and timing information for reconstructing the audio stream in the receiver. VoIP applications have to implement in the receiver the appropriate techniques to buffer and re-order the packets to provide a voice service resilient to a percentage of packet loss and variable packet arrival delay.

The VoIP test bed consists of the underlying Ad hoc networking stack and the real time VoIP application including the Session Initiation Protocol (SIP) [50] signalling protocol, the transport protocol (i.e. RTP) and the components for capturing the voice in the sender and playing it back in the receiver.

The VoIP application in the transmitter starts by sampling the analogue audio signal, digitalizing it to audio bytes at a sampling frequency. The typical sampling frequency value (i.e. PCM format) for audio streams is 8000 Hz with 8 bits per sample, which results in a 64 kbps audio stream. Following, the VoIP application breaks down the sampled audio into small packets that are compressed using specific algorithms (e.g. GSM codec [48]) to generate audio frames that will be transmitted using RTP [49]. GSM codec takes an audio stream sampled at 8000 Hz and 13 bits per sample. GSM audio frames contain 20ms of audio recorded at 8 samples/ms with 13 bits/sample that result in 260 bytes of uncompressed audio data per frame. The GSM codec generates a 33 byte packet of compressed audio (i.e. compression ratio is then 33/260 = 12.69%).

Figure 19 represents the model of the sender and receiver including the different processes that take place between capturing the audio in the microphone, the sampling and GSM codec processing until the audio frame is sent over the network via RTP message. The receiver's VoIP application takes care of receiving the RTP messages from the network (i.e. socket module), unpacking them (i.e. decompresses the audio frames using the selected GSM codec), interpreting the sequence numbering and implementing the buffering of audio frames to ensure continuous playback. After buffering enough frames, the receiver reconstructs the audio samples and plays them back.

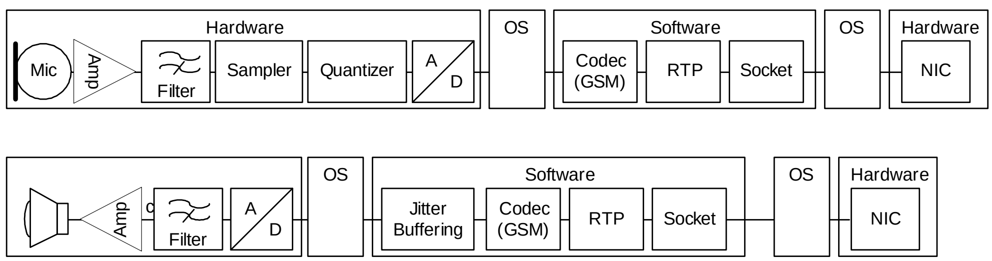


**Figure 19. Audio sender and receiver model.**

In this model there are different buffers that affect the end to end delay. The audio device system used in the test bed (i.e. Open Sound System; OSS [51]) implements different buffers for playback and recording. The OSS provides an interface for the applications to interact directly with the audio driver. Thus, the VoIP application can specify the number and length of the recording and playback buffers. In real time applications it is recommended to keep the audio buffers small in order to speed up the processing. The VoIP application requests the audio driver to allocate two buffers of 512bytes each that will allow recording 256 audio samples (i.e. 256/8 KHz = 32ms audio fragment) of 16bits on each buffer. The VoIP application also requests the audio driver several buffers in the receiver side to store the decoded audio before playback in order to compensate additional network delays.

An additional buffer to consider in the model is the jitter buffer that the VoIP application implements to correct the inter arrival delay difference between consecutive packets. The jitter delay has higher variability in Ad hoc networks because of the additional parameters involved such as processing delay in the intermediate nodes, and dynamic route changes. The jitter buffer length has an impact on the quality of the audio session. Increasing the jitter buffer length will reduce the perceived pauses in audio playback. This results in smooth playback but will increase the overall delay. ITU-T quality recommendation [52] is a maximum delay of 400 ms, and 250 ms for an audio session. On the other hand, if we reduce the jitter buffer length, the overall delay decreases. However, the overall result is a low quality session with a lot of pauses in the playback. Therefore, the length of the jitter buffer has to be balanced between these two extremes, to keep a reasonable quality of service.

The number and length of the audio system buffers have an impact on the end to end delay and needs to be optimised in order to provide good performance. On the sender side the audio device has to record full segments before delivering the audio samples back to the VoIP application. Thus, large buffers have to be filled before the application is able to encapsulate the audio. On the receiver side, if the amount of audio data in the buffer is not enough, the playback is stopped resulting in popping sound. The application has to guarantee enough audio data in the buffers to provide a continuous playback.

The RTP payload length is another parameter that affects the system performance. The RTP payload consists of the number of GSM audio frames included in each RTP message. If the RTP payload increases, the audio playback at the receiver will be enhanced since each packet holds enough audio data to play until the next packet arrives. However, increasing the payload means longer recording time resulting in a higher overall delay. In addition, if one packet is lost, a larger amount of audio data is lost resulting in longer pauses in the playback.

We performed several tests [53], [14] changing the jitter buffer length between 60 and 100ms, increasing the RTP payload from 1 to 10 GSM audio frames per RTP message and changing the number and length of the audio device buffers (i.e. from 1 buffer of 512 or 1024bytes to 8 buffers of 512 or 1024bytes). In the test results we experience that changing the RTP payload under different test conditions has a direct effect on the overall performance. Selecting a RTP payload length of only 1 or 2 GSM audio frames per RTP message provides the worst quality regardless of the network conditions. Instead, a RTP payload ranging from 3 to 5 GSM audio frames per RTP message provides the best quality under different test conditions. Thus, in network conditions with a higher percentage of packet loss using a RTP payload of 3 GSM audio frames per RTP message is the best approach. However, in situations with lower percentage of packet loss and higher bandwidth, a RTP payload length of 5 GSM audio frames provides an excellent quality.

The results from those tests show that independently of the routing protocol (i.e. AODV or OLSR) and under different test conditions the following settings provide the best performance of the system.

- GSM audio frames per RTP message: 3.
- Jitter buffer length: 60ms.
- Audio buffer in the recording side: 1 (1024bytes).
- Audio device buffers in the playback side: 4 (512bytes x 4).

The next objective of the test bed is to measure the overall performance of VoIP sessions in Ad hoc networks considering AODV and OLSR [16] as the routing protocols.

In Ad hoc networks, VoIP applications have to deal with new requirements because of node mobility and self-created nature of the network. The Ad hoc routing protocols do not affect the VoIP sessions once the route is established. However, the routing protocols have to ensure reliable routes and react quickly to route changes to guarantee a smooth audio packet delivery. Using the test bed we analysed the performance from signalling overhead, end to end packet delay and routing latency (i.e. route re-establishment).

VoIP sessions were set up using 2, 3 and 4 nodes to measure the effect of increasing the number of hops. We consider that when increasing the number of nodes we are analysing the increase in the number of hops but we cannot measure the effect of the node density like in the simulations. We cannot measure the effect of mobility since the tests were performed with zero mobility (i.e. all the nodes were static while the two endpoints established the VoIP session). However, during the tests the link was broken to measure the routing latency and the consequent effect on the QoS. Moreover, we run the test in the laboratory with continuous link breaks caused by metal doors, people passing by the testing scenario and other wireless networks running at the same time (i.e. TKK WLAN).

Figure 20 depicts the layout of the three test cases performed using PDAs (i.e. HP 3850 iPAQs [54], running Familiar Linux distribution [55] including standard implementation of IEEE 802.11 MAC protocol, 206 MHz Intel StrongARM processor and 64 MB memory) with wireless card 802.11b at 11Mbps, channel 10 (2.457MHz) and the following system parameters.

- Jitter buffer length: 60ms.
- Recording buffer length: 1 buffer x 1024 bytes.
- Playback buffer length: 4 buffers x 512 bytes.
- RTP payload: 3 GSM packets (GSM library v06.10 [48]).
- Traffic measurement tools: Ethereal and Tcpdump [56].
- Signalling protocol: SIP [50].
- Transport protocol: JRTP library v2.9 [57].
- Ad hoc routing protocols: OLSR v0.45 and AODV v0.91.

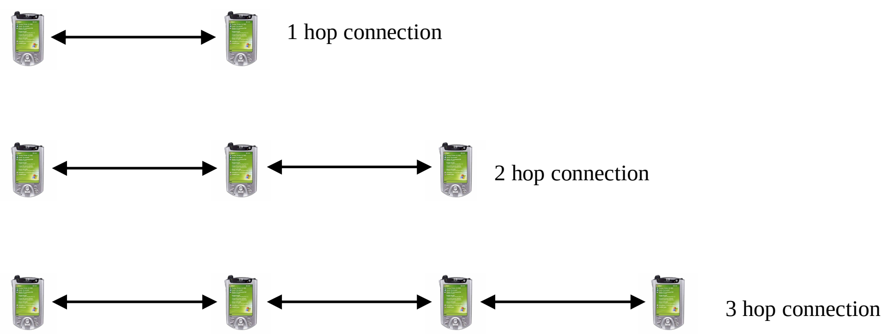


**Figure 20. VoIP test bed scenarios.**

The layout used for the test bed is linear but due to link breaks and fluctuations in the signal, the environment resembles a small Ad hoc network with dynamic topology. Thus, nodes that are 2 hops away can have a direct link when the signal is strong. However, in other conditions, even nodes located 1 hop away can be momentarily unreachable. Moreover, when we compare our measurement results with simulation results, we seek to take into account the differences in the measurement and simulation scenarios.

We studied the jitter delay, the end to end packet delay and their distribution for OLSR and AODV over 1 hop connection (i.e. direct connection between endpoints) with zero node mobility. This study is the basis for the analysis of the system performance measures after increasing the number of hops and node mobility. Node mobility is implemented manually by breaking the link between two nodes to measure the routing latency.

Figure 21 shows the OLSR jitter delay in the arrival of consecutive packets identified with their RTP sequence number over 1 hop connection. The delay between packets varies around the average of 60ms. The empty spaces in the figure are the effect of the manual link break to measure the routing latency.

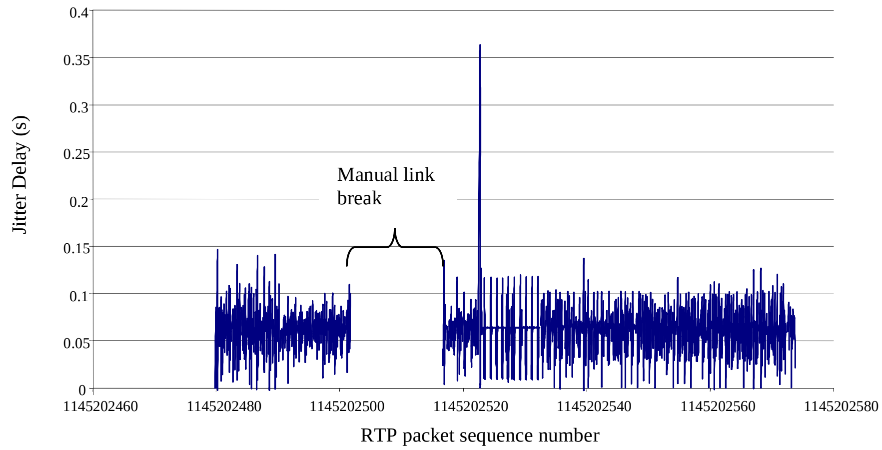


**Figure 21. OLSR jitter delay over 1 hop connection.**

Figure 22 shows the distribution of the OLSR jitter delay over the 1 hop connection, which is around the same value (i.e. 64ms) for most of the packets with a maximum deviation of 120ms.

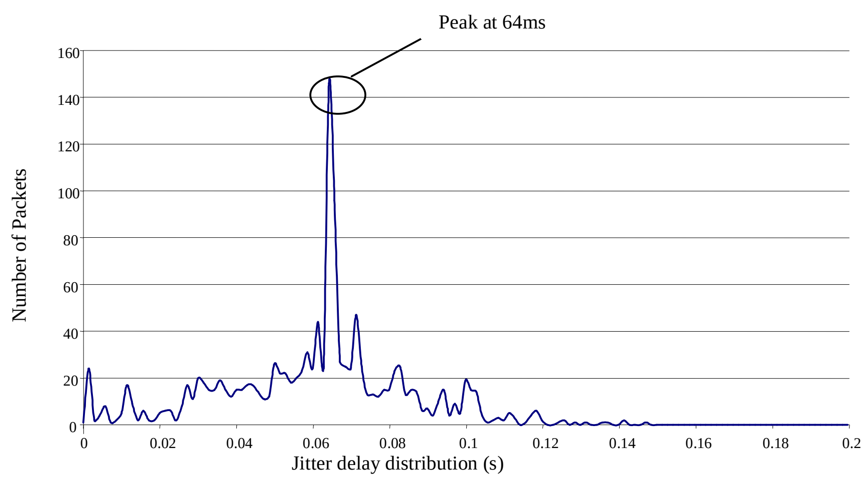


**Figure 22. Distribution of the OLSR jitter delay over 1 hop connection.**

Figure 23 shows that the OLSR end to end packet delay in the case of 1 hop connection is almost constant around 140ms with some exceptions due to interferences. The empty space is due to the manual link break to measure the routing latency.

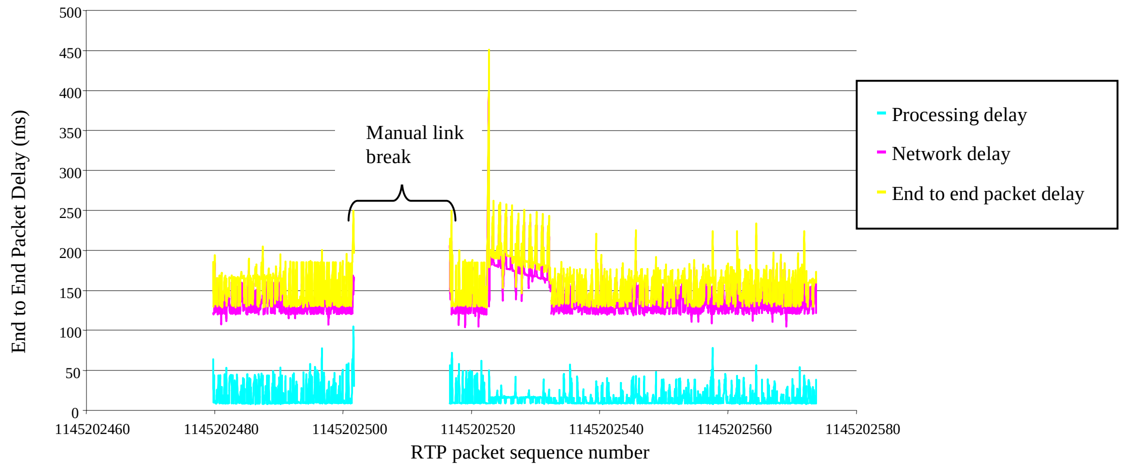


**Figure 23. OLSR end to end packet delay over 1 hop connection.**

Figure 24 shows the distribution of the OLSR end to end packet delay, which presents several peaks of different values at 133ms, 140ms, 155ms, 160ms, 170ms and 180ms. The variation is due to the additional processing delay and the length of the route re-establishment time after link breaks and the interferences as represented in Figure 23.

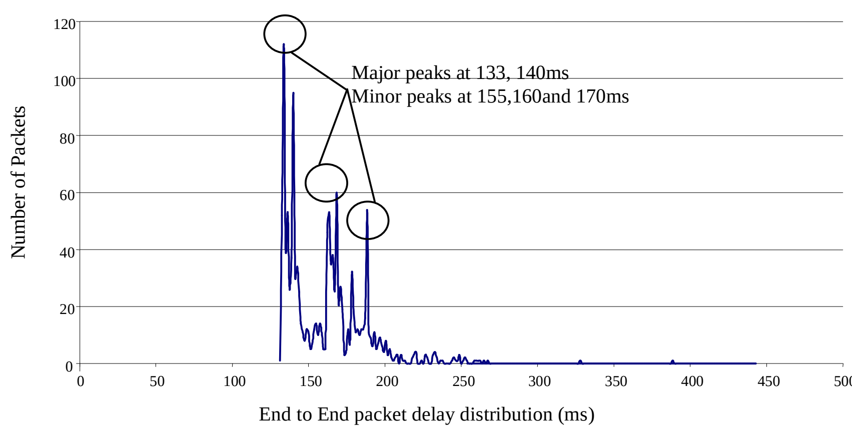


**Figure 24. OLSR end to end packet delay distribution over 1 hop connection.**

Figure 25 shows the AODV jitter delay in the arrival of consecutive packets identified with their RTP sequence number over a 1 hop connection. The delay between packets is around the same value (i.e. 60ms) than the ones obtained with OLSR. However, a more dynamic variation is observed in AODV.

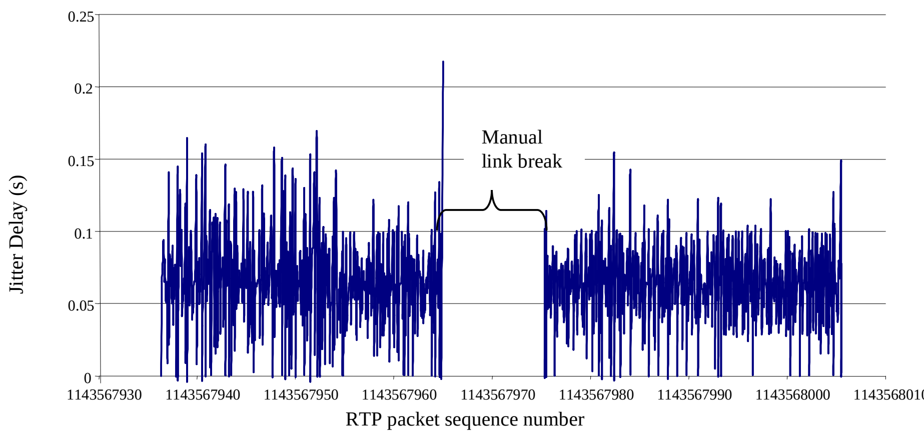


**Figure 25. AODV jitter delay over 1 hop connection.**

Figure 26 shows the distribution of the AODV jitter delay over a 1 hop connection, which in theory should be the same as with OLSR as the routing protocol does not affect the packet delivery after the route is found. In practice, the AODV jitter delay distribution presents a maximum deviation of 160ms, which is higher than the one obtained for OLSR (120ms) but the peak at 64ms is also visible in this case, and the number of packets in those peaks is lower in AODV (43packets) than OLSR (147packets).

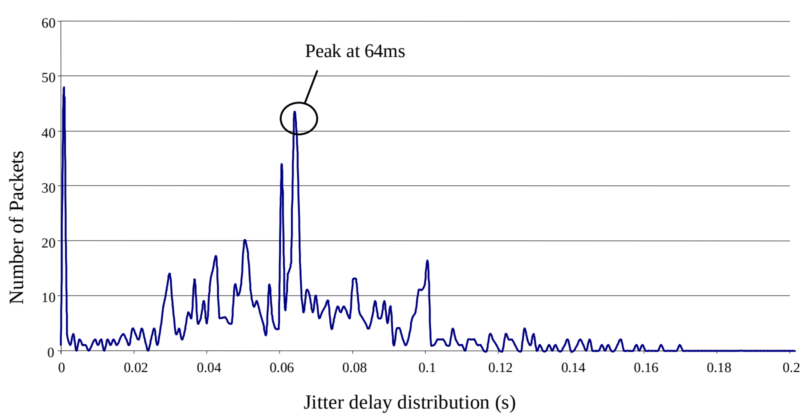


**Figure 26. Distribution of the AODV jitter delay over 1 hop connection.**

Figure 27 shows the AODV end to end packet delay over a 1 hop connection, which is similar to the values obtained for OLSR as depicted in Figure 23. However, a slightly higher and constant processing delay is visible for AODV.

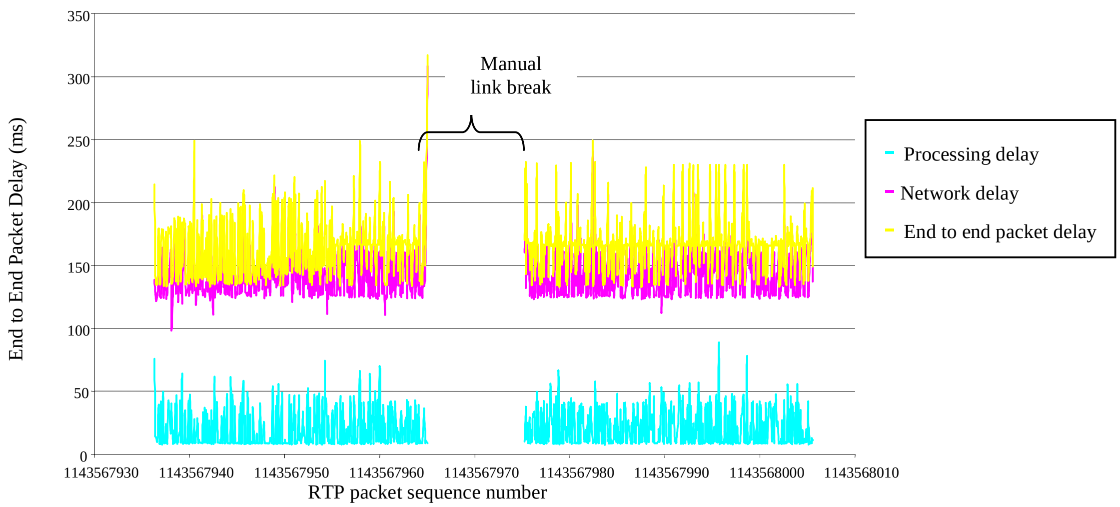


**Figure 27. AODV end to end packet delay over 1 hop connection.**

Figure 28 shows the distribution of the AODV end to end packet delay over the 1 hop connection, which presents fewer peaks than OLSR. The reason is that after the route is found, unless the link is broken, AODV spends fewer resources in additional routing processes than OLSR.

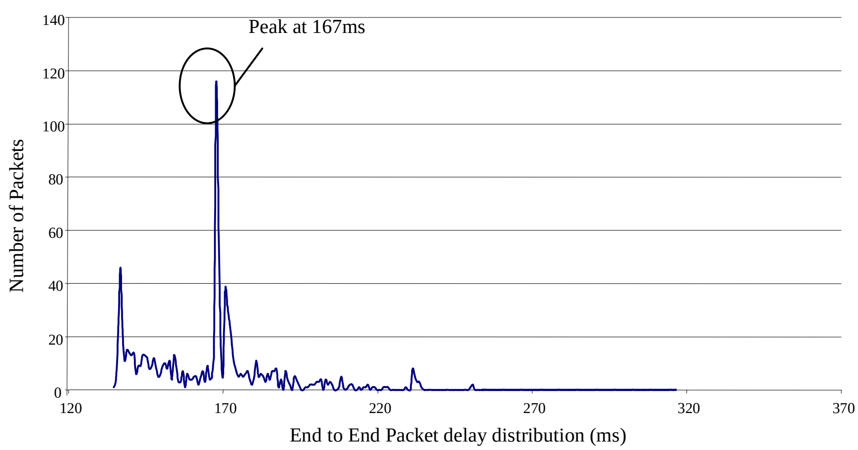


**Figure 28. AODV end to end packet delay distribution over 1 hop connection.**

The jitter delay and end to end packet delay test results for AODV and OLSR in the case of 1 hop connection have been presented in detail. Table 4 summarises the results from more than 100 tests, for AODV and OLSR in 1, 2 and 3 hops connections.

**Table 4. Summary of performance metrics for AODV and OLSR over 1, 2 and 3 hop connections.**

| Performance metrics | AODV/1hop | AODV/2hops | AODV/3hops | OLSR/1hop | OLSR/2hops | OLSR/3hops |
|---|---|---|---|---|---|---|
| **End to end packet delay** | | | | | | |
| Average (ms)<br>Std deviation (ms)<br>90% percentile | 163.595<br>21.915<br>188.419 | 168.468<br>25.299<br>202.487 | 195.739<br>20.854<br>228.611 | 158.414<br>27.250<br>187.411 | 166.485<br>43.669<br>227.539 | 187.093<br>37.638<br>244.214 |
| **Jitter delay** | | | | | | |
| Average (ms)<br>Std deviation (ms)<br>90% percentile | 61<br>31<br>98 | 62<br>34<br>99 | 61<br>32<br>92 | 60<br>26<br>92 | 60<br>33<br>97 | 61<br>43<br>99 |
| **Packet loss** | | | | | | |
| Number of packets lost<br>% of packet loss<br>(packets lost/RTP packets) | 1<br>0.04%<br>(1/2353) | 4<br>0.06%<br>(4/6858) | 15<br>0.4%<br>(15/3665) | 3<br>0.09%<br>(3/3215) | 4<br>0.08%<br>(4/4688) | 16<br>0.2%<br>(16/7969) |
| **Routing overhead** | | | | | | |
| % routing overhead<br>(Routing packets/RTP packets) | 7.22 %<br>(170/2353) | 7.38%<br>(506/6858) | 18.17 %<br>(666 / 3665) | 3.39 %<br>(109/3215) | 3.86%<br>(181/4688) | 3.58%<br>(286/7969) |
| **Re-Routing latency** (seconds) | 0.5 | 1 | 1.5 | 1 | 8 | 15 |

From this summary and considering the limitations of the results obtained from a small-scale real Ad hoc network we conclude that the jitter delay grows as a function of the number of hops. The percentage of packet loss is low in both AODV and OLSR. The percentage of packet loss increases with the number of hops for both protocols. The jitter delay in the receiving node will increase with the packet loss if it cannot be resolved with interleaving or additional buffering in reception. The end to end packet delay tends to increase equally in both AODV and OLSR and it increases quite linearly with the number of hops. The routing latency in AODV is lower than OLSR and in both cases it increases linearly with the number of hops. The routing overhead is higher in AODV than in OLSR. This is contradicting with the results from the simulations and Proposition 7. This is because we considered a small-scale Ad hoc network where OLSR maintains a small amount of routing information compared to AODV that has to flood the entire network for the routing discovery process. The routing overhead remains almost constant in OLSR regardless of the number of hops while in AODV increases exponentially when the number of hops grows (i.e. 3 hops). This behaviour was not observed in the simulations and supports the statement that AODV performs efficiently in small networks but its routing overhead increases significantly in large networks with long end to end paths. When comparing the results from the simulations and the test bed we have to consider that when increasing the number of nodes N in simulations we are increasing the node density, but when increasing the number of nodes N in the test bed we increase the network coverage by increasing the number of hops.

The end to end packet delay obtained from the simulations for reactive routing protocols with UDP traffic flows is modelled with Eq 30:

$$D_R(M, N) = (0.0014N)M + 0.1 + 0.011N(s)$$

The end to end packet delay from the simulations for proactive routing protocols with UDP traffic flows is modelled with Eq 34:

$$D_P(M, N) = (0.0025N)M + 0.113 + 0.07N(s)$$

Replacing the values for the number of nodes and the mobility used in the test bed (i.e. N=4 and M=0) the results are the following.

$$D_R(M = 0, N = 4)\big|_{Simulation} = 0.1 + 0.011 * 4 = 144(ms)$$

$$D_R(M = 0, N = 4)\big|_{TestBed} = 195(ms)$$

$$D_P(M = 0, N = 4)\big|_{Simulation} = 0.113 + 0.07 * 4 = 393(ms)$$

$$D_P(M = 0, N = 4)\big|_{TestBed} = 187(ms)$$

The end to end packet delay results from the simulation and the test bed for the reactive routing protocol are quite similar (i.e. around 150±50ms). This verifies the Eq 30 obtained from the simulations. The end to end packet delay should be similar in both reactive and proactive routing protocols when node mobility is zero. The results from the simulations for proactive routing protocol are two times higher than the results obtained in the test bed. The higher end to end packet delay in proactive routing than in reactive routing obtained in the simulations results is due to the effect of the link breaks where the routing latency increases the overall delay. The simulations provide an average end to end packet delay values that include the required effect of the routing latency in proactive protocols when the links break in high mobility conditions. The simulations consider a large area compared to the test bed and when applying the same number of nodes to the equations obtained from the simulations it is quite probable that nodes are quite disperse and they are not connected. Thus, the effect of link breaks will have a major impact when utilising the equations obtained from the simulations.

Therefore, we conclude that the equations obtained from the simulations to model the end to end packet delay are accurate enough. However, in low mobility and low density conditions the results are pessimistic for proactive routing protocols.

The percentage of packet loss obtained from the simulations for reactive routing protocols with UDP traffic flows is modelled with Eq 39:

$$L_{Ru}(M,N) \approx (0.01 + 0.03N)e^{(1.34+0.35N)M} \ (\%)$$

The percentage of packet loss obtained from the simulations for proactive routing protocols with UDP traffic flows is modelled with Eq 44:

$$L_{Pu}(M,N) = (0.09 + 0.13N)e^{(0.69+0.012N)M} \ (\%)$$

Replacing the values for the number of nodes and the mobility used in the test bed (i.e. N=2, N=4 and M=0) the results are the following.

$$L_{Ru}(M=0, N=2)\big|_{Simulation} = 0.01 + 0.03 * 2 = 0.07(\%)$$

$$L_{Ru}(M=0, N=2)\big|_{TestBed} = 0.06(\%)$$

$$L_{Ru}(M=0, N=4)\big|_{Simulation} = 0.01 + 0.03 * 4 = 0.13(\%)$$

$$L_{Ru}(M=0, N=4)\big|_{TestBed} = 0.4(\%)$$

$$L_{Pu}(M=0, N=2)\big|_{Simulation} = 0.09 + 0.13 * 2 = 0.35(\%)$$

$$L_{Pu}(M=0, N=2)\big|_{TestBed} = 0.08(\%)$$

$$L_{Pu}(M=0, N=4)\big|_{Simulation} = 0.09 + 0.13 * 4 = 0.61(\%)$$

$$L_{Pu}(M=0, N=4)\big|_{TestBed} = 0.2(\%)$$

These results are quite accurate for reactive routing protocols with reduced number of hops but they are optimistic when the number of hops increases. On the other hand the simulations are over pessimistic for proactive routing protocols but the difference is lower when the number of hops increases. In general simulations reflect similar behaviour to the test bed. The percentage of packet loss increases with higher number of nodes or hops for reactive and proactive routing protocols. However, the simulations results for proactive routing protocols are 3 to 4 times higher than the values obtained in the test bed. The simulation results for reactive routing protocols with higher number of nodes are 3 times lower than the results obtained from the test bed. We believe, the reason is that the equations obtained from the simulations results are from a medium network, thus when applying the same equations to a small network the approximation error is higher. If we keep the network size and reduce the number of nodes to simulate a small network then we are reducing the node density, which increases the distance between nodes and the probability of link breaks. Moreover, the simulations consider multiple connections at the same time while in the test bed there is a single connection. In the simulations, several connections with different routes and number of hops are established. The packet loss is measured in the test bed considering the increase in the number of hops, which cannot be estimated in the simulations since the nodes move randomly (i.e. waypoint mobility model). The test bed provides a more controlled environment where we can measure the number of active connections, the routes and the number of hops on each route.

Another anomaly we observe in the test bed is that proactive routing protocols present a lower percentage of packet loss than reactive routing protocols for higher number of nodes. Thus, P3.3 holds for a reduced number of nodes but it does not apply in case of large networks. Therefore, we conclude that the equations obtained from the simulations to model the percentage of packet loss are accurate when considering a small network with a reduced number of hops. In small networks reactive protocols show better results. However, the simulations are more accurate when considering medium to large networks with a higher number of nodes. The simulation results are too optimistic for reactive routing protocols and pessimistic for proactive routing protocols.

The routing overhead obtained from the simulations for reactive routing protocols with UDP traffic flows is modelled with Eq 21:

$$\Omega_{Ru}(M,N) = (155.4 + 5.1N)e^{(0.045+0.009N)M}\ (Kbytes)$$

The routing overhead obtained from the simulations for proactive routing protocols with UDP traffic flows is modelled with Eq 26:

$$\Omega_{Pu}(M,N) = (615 + 375N)e^{(0.001+0.002N)M}\ (Kbytes)$$

Replacing the values for the number of nodes and the mobility used in the test bed (i.e. M=0, N=2 and N=4) the results are the following.

$$\Omega_{Ru}(M=0, N=2)\big|_{Simulation} = 155.4 + 5.1*2 = 165.6(Kbytes)$$
$$\Omega_{Ru}(M=0, N=4)\big|_{Simulation} = 155.4 + 5.1*4 = 175.8(Kbytes)$$
$$\Omega_{Pu}(M=0, N=2)\big|_{Simulation} = 615 + 375*2 = 1365(Kbytes)$$
$$\Omega_{Pu}(M=0, N=4)\big|_{Simulation} = 615 + 375*4 = 2115(Kbytes)$$

The simulations were executed during 900 seconds with 20 active connections, with a packet rate of 8packets/sec and 65bytes of packet size. This means that the total data transmitted during each simulation was 9360Kbytes as calculated in Eq 53.

**Eq 53.** $DataTransmitted = 20_{conn} * 8_{packet/\sec} * 65_{bytes/\,packet} * 900_{\sec} = 9360Kbytes$

We obtain that the percentage of packet loss for the number of nodes and the mobility used in the test bed (i.e. M=0, N=2 and N=4) are the following.

$$L_{Ru}(M = 0, N = 2)\big|_{Simulation} = 0.07(\%)$$
$$L_{Ru}(M = 0, N = 4)\big|_{Simulation} = 0.13(\%)$$
$$L_{Pu}(M = 0, N = 2)\big|_{Simulation} = 0.35(\%)$$
$$L_{Pu}(M = 0, N = 4)\big|_{Simulation} = 0.61(\%)$$

The total data received is the data transmitted minus the packet loss for each case.

$$L_{Ru}(M = 0, N = 2) \Rightarrow Data\_received = 9360 - 6.55 = 9353.45(Kbytes)$$
$$L_{Ru}(M = 0, N = 4) \Rightarrow Data\_received = 9360 - 113.25 = 9246.75(Kbytes)$$
$$L_{Pu}(M = 0, N = 2) \Rightarrow Data\_received = 9360 - 32.76 = 9327.24(Kbytes)$$
$$L_{Pu}(M = 0, N = 4) \Rightarrow Data\_received = 9360 - 57.09 = 9302.91(Kbytes)$$

Therefore, the percentage of routing overhead for each case is:

$$\Omega_{Ru}(M = 0, N = 2)\big|_{Simulation} = \frac{165.6}{9353.45} = 1.77\%$$

$$\Omega_{Ru}(M = 0, N = 2)\big|_{TestBed} = 7.38\%$$

$$\Omega_{Ru}(M = 0, N = 4)\big|_{Simulation} = \frac{175.8}{9246.75} = 1.9\%$$

$$\Omega_{Ru}(M = 0, N = 4)\big|_{TestBed} = 18.17\%$$

$$\Omega_{Pu}(M = 0, N = 2)\big|_{Simulation} = \frac{1365}{9327.24} = 14.63\%$$

$$\Omega_{Pu}(M = 0, N = 2)\big|_{TestBed} = 3.86\%$$

$$\Omega_{Pu}(M = 0, N = 4)\big|_{Simulation} = \frac{2115}{9302.91} = 22.73\%$$

$$\Omega_{Pu}(M = 0, N = 4)\big|_{TestBed} = 3.58\%$$

The values obtained from the simulations diverge from the test bed results.

Figure 29 shows that the AODV routing overhead is higher than in OLSR based on the results obtained from the test bed. OLSR keeps similar overhead regardless of the number of hops but AODV almost doubles the routing overhead when increasing the number of nodes.

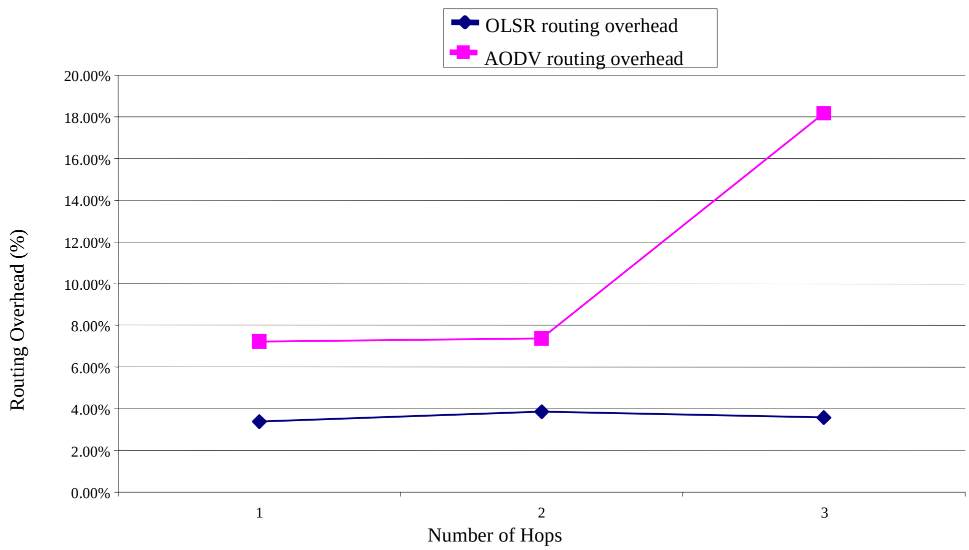


**Figure 29. AODV and OLSR routing overhead over 1, 2 and 3 hop connections.**

The equations obtained from the simulations show that both protocols are affected by the number of nodes. OLSR presents higher routing overhead than AODV for the same number of nodes. AODV maintains an almost constant routing overhead with a minor percentage increase with the number of nodes. The test bed shows the opposite results, OLSR has lower routing overhead than AODV and its value is almost constant regardless of the number of nodes. AODV presents a routing overhead three times higher than OLSR when the number of nodes increases. We have to consider that the results from the test bed are not considering the node density like in the simulations. However, the test bed shows considerable differences compared to the simulations so we can conclude that the estimated equations for modelling the routing overhead based on the simulation results are not accurate.

We have to highlight that when increasing the number of nodes in the test bed, we are also increasing the number of hops. This leads to the fact that in the test bed AODV generates higher routing overhead because there is a dependency with the number of hops, which cannot be reflected in the simulations. The simulations provide an overall value that represents the average results including different factors such as number of hops, multiple connections running in parallel with different paths and link breaks that may generate additional overhead. OLSR routing overhead results from the simulations increase with the number of nodes which is not visible in the results from the test bed. The simulations provide estimated values for OLSR in small scale networks where no considerable amount of routing overhead exists since the link information to be distributed among a few nodes is low.

Based on the results presented in this section we conclude that the number of hops is a relevant metric to consider when designing an efficient routing protocol. It has to be taken into account in the equations that model the routing overhead in order to accurately reflect the actual behaviour of the different protocols.

Figure 30 shows the test bed results of the routing latency (i.e. $\Phi$) versus the number of hops, a new metric that we did not measure in the simulations. This metric varies with mobility but mainly with the number of hops in the path. The routing latency affects the network QoS mainly when considering real time applications that suffer from jitter and end to end packet delay. Figure 30 shows that the routing latency in AODV and OLSR increases with the number of hops (i.e. $\gamma$). However, AODV reacts faster in order to obtain a new route and follows a linear increment with a smaller factor than OLSR.

The AODV and OLSR routing latency can be modelled with Eq 54 and Eq 55.

**Eq 54.** $\Phi_{AODV} = 0.5\gamma$

**Eq 55.** $\Phi_{OLSR} = 7\gamma - 6$

OLSR would seem to require a link layer alert mechanism (not implemented in the test bed) to detect broken routes and the node has to communicate the topology update to their neighbours so they can re-calculate the new route.

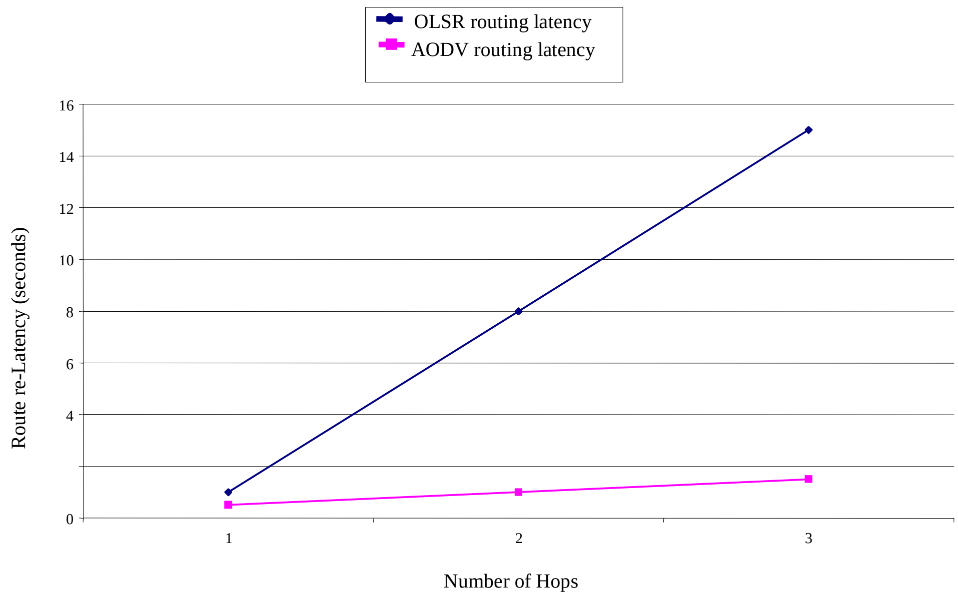


**Figure 30. AODV and OLSR routing latency over 1, 2 and 3 hop connections.**

### 2.8.2 Test Bed Results Conclusions

In general, the results obtained from the real time VoIP application and the simulations are comparable but there are some exceptions that we will review in this section [17].

The simulation results are quite accurate in the end to end packet delay for AODV but over pessimistic in the case of OLSR. The values for OLSR in real Ad hoc networks are lower than the ones obtained in the simulations. This is because the simulations include multiple connections with several hops while in the test bed we run a single connection with only a few hops. The difference is also due to the fact that the estimated equations from the simulations include the mobility effect where links can be broken, and for that reason OLSR presents a higher end to end packet delay to re-establish the route. However, in the test bed, with zero mobility, both AODV and OLSR introduce similar end to end packet delay.

The simulation results are quite accurate when measuring the percentage of packet loss for AODV in a small network but the results are optimistic when increasing the network size. On the other hand, the results are pessimistic for OLSR in small and large networks. The simulation results are in line with the test bed results when increasing the number of nodes since both indicate that the percentage of packet loss increases. The simulations indicate that AODV has lower packet loss than OLSR which is correct with the results from the test bed for a reduced number of nodes but the test bed shows the opposite when increasing the number of nodes. The test bed shows that OLSR has lower packet loss than AODV in medium to large networks. Nevertheless, we have to consider that the simulations provide overall results from several connections with a certain duration where the endpoints are selected randomly, while in the test bed a single bidirectional connection is maintained between the same nodes during the testing session.

In terms of routing overhead, OLSR shows higher values than AODV in the simulation results, while in the test bed it is just the opposite. The difference in the results is because the simulations obtain the overall value without considering the number of hops. In the test bed results AODV presents higher increase of the routing overhead with the number of hops while OLSR is not affected. Thus the equations from the simulations can be used to estimate the overall routing overhead in different protocols. However, they do not reflect the impact of certain metrics like the number of hops and they are not suitable for the protocol design.

The test bed provides measures about routing latency which cannot be obtained from the simulations. The test bed shows that routing latency is crucial for the real time communications in Ad hoc networks with multihop routes.

In general the simulations provide estimates about network performance with different routing protocols but we need the results from the test bed to correct and in some cases complement the simulation results.

Based on the results from the test bed, we conclude that proactive routing protocols in stable networks obtain a higher percentage of optimal routes, which minimises the end to end packet delay for real time applications. Obtaining the optimal routes is critical because of the impact of the number of hops in the end to end packet delay and jitter. Proactive routing protocols show lower packet loss than reactive routing protocols in large networks. Reactive protocols present a lower percentage of packet loss in small networks (i.e. reduced number of hops) with low mobility as well as prompt reaction under link breaks. These are all requirements necessary for real time applications. Moreover, to accommodate real time applications in Ad hoc networks a cross-layer architecture is required to establish a communication channel between end points. This will allow receiving routing information during an ongoing real time session to dynamically accommodate the RTP payload to the link conditions.

### 2.9 Ad hoc Routing Requirements

Routing protocols in Ad hoc networks need to rapidly adapt to network changes. They have to minimise the consumption of network processing, transmission and storage resources during the adaptation process to maximise the availability of the nodes. Ad hoc routing protocols have to cope with the topology dynamics, variable bandwidth, mobility and unreliable wireless connections. Simulation and test bed results demonstrate that protocols targeted for small and medium Ad hoc networks do not perform well in large networks.

Figure 31 shows that different routing protocols are required depending on the size of the Ad hoc network. The test bed results show that in small networks, packet loss and routing latency of reactive protocols is low while in large networks it is significantly higher. Moreover, the end-to-end path in small networks includes few hops while in large networks the number of hops is bigger with the consequent higher end to end packet delay. In Figure 31, A) we can see that a small network has a quick route discovery process and low end to end packet delay while in B) a large network suffers from long route discovery process and huge end to end packet delay.

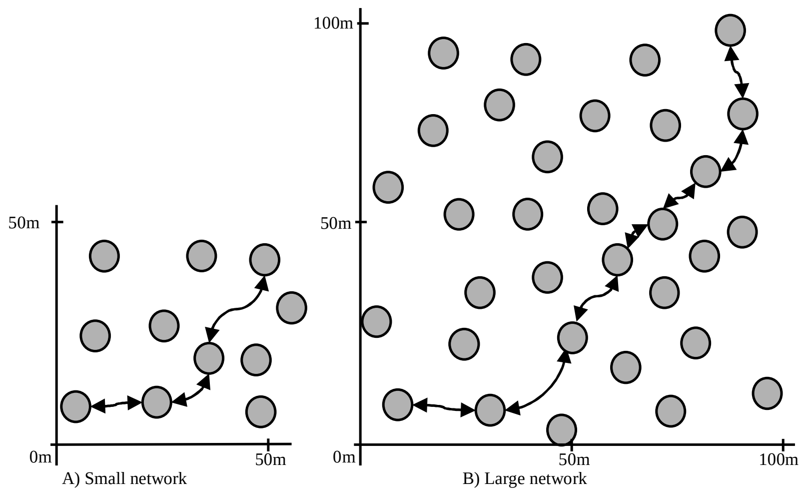


**Figure 31. Small versus large networks routing requirements.**

The simulations results show that proactive routing protocols obtain the most optimal routes regardless of the number of nodes and mobility. Proactive routing protocols maintain the network topology information up to date, reducing routing latency. The routes are optimised using algorithms based on different metrics such as number of hops and link cost. Different routes can be used depending on the application requirements (i.e. multipath routing optimisation [58]). An equivalent procedure in reactive routing protocols would take several iterations until the optimal route would be found, with the consequent routing latency. Proactive routing protocols are suitable for small networks with a limited number of nodes because the routing overhead, the routing table storage and the computational overhead are low. However, when the number of nodes increases, they are inefficient. Therefore, in Ad hoc networks a simple and low resource consuming protocol should be used for routing within a cluster while few selected nodes act as gateways providing network scalability [18].

## 2.10 Fully Distributed Virtual Backbone Concept

The existing Ad hoc routing protocols are reliable in small and stable networks, where each node can efficiently perform the routing functions based on the state information obtained from the entire network. However, in large networks the entire state information of the network is not available for the nodes, and routing is based only on partial topology knowledge.

### 2.10.1 Nodes Classification

We explore one solution to improve the scalability of Ad hoc networks based on a hybrid routing mechanism where the physical network is transformed into a virtual network [20]. In this virtual network we differentiate two types of nodes. The *ordinary nodes* perform the basic routing functionality such as packet forwarding and on demand route discovery, and the *smart nodes* maintain and acquire topology information to be distributed through the network via other smart nodes. Therefore, the diameter of the network is reduced by having a set of nodes that abstract the network state and reduce its variability. The smart nodes will facilitate the routing to the ordinary nodes in the network by reducing the number of hops, end to end packet delay[2] and increasing connectivity between distant nodes in large networks.

Based on the topology information, the smart nodes calculate the shortest path and optimal routes necessary to have a stable network. A stable network means that the topology changes have to be slow enough to allow the updates to reach all the nodes in the network. The Ad hoc nodes mobility may be high and the topology information is not steady during the necessary period of time required by the algorithm to calculate the optimal path based on known conditions. This sets a requirement for Ad hoc networks that is difficult to accomplish due to lack of nodes that maintain the network state when using reactive protocols. The heterogeneous conditions in Ad hoc networks make the routing unreliable and

[2] Each node in the path contributes to the total end to end delay with a fixed delay from the MAC layer to access the shared channel plus other delay components such as the transmission delay from the message processing and the radio delay when the node switches from reception to transmission mode.

difficult to optimize based on metrics like shortest path, minimum delay or energy cost.

The routing in Ad hoc networks will not converge into the shortest path unless there are smart nodes maintaining the topology information and calculating the optimal routes. Therefore, Ad hoc networks require a proactive routing protocol to maintain the network topology information despite that in some cases it will be stale due to high node mobility. The smart nodes implement a higher hierarchical routing level than the ordinary nodes as represented in Figure 32. The ordinary nodes do not participate in the shortest path calculation and use reactive routing. The smart nodes also use the reactive routing and participate in the lower hierarchical routing layer together with the ordinary nodes.

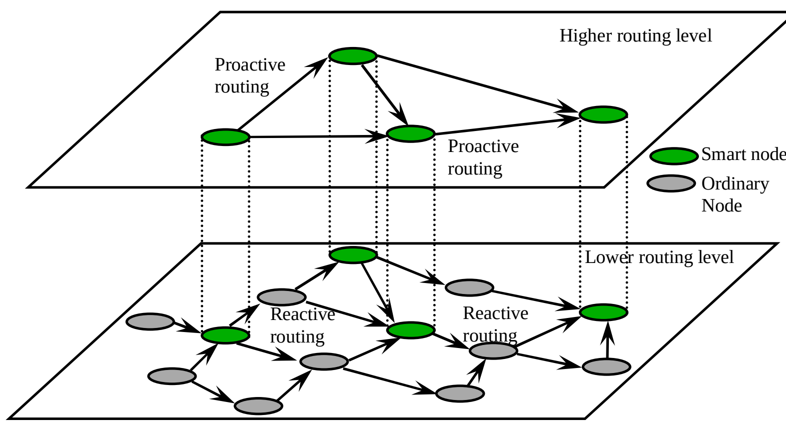


**Figure 32. Node classification based on contribution to network topology information.**

The main criterion for the proposed node classification is based on the connectivity and the capability for maintaining and distributing topology information in a reliable manner. In principle, any node can maintain the topology information if it has enough resources (i.e. memory, battery and processing power, etc). Nodes can share the topology information within the network if they have a reliable connectivity (i.e. low mobility) that allows them to follow continuous topology updates. The smart nodes will create a Fully Distributed Virtual Backbone (FDVB) to maintain and distribute the network topology information at the expense of consuming their own resources. The FDVB will provide a mechanism to allow quick network knowledge to converge with minimal messaging control and complexity.

### 2.10.2 Hybrid Routing Approach

We identified the need to introduce smart nodes performing extra routing functionality in Ad hoc networks. However, the preferred routing protocol to be implemented is the most critical part to improve scalability in Ad hoc networks, and it remains to be selected.

Based on the simulation results and the test bed analysis, the combination of a reactive protocol that responds quickly to link breaks and a proactive protocol that provides optimal routes seems to be the optimal solution. Therefore, we propose a novel hybrid approach named Scalable Ad hoc Routing Protocol (SARP) to overcome the drawbacks of existing routing protocols to scale up to large Ad hoc networks. In our hybrid approach the nodes are grouped into clusters and the cluster heads provide scalability by taking care of the heavy routing functionality between clusters. The drawbacks in cluster-based routing protocols are the additional complexity required in the nodes to implement the clustering algorithm. These protocols have additional overhead required for selecting the cluster head and the fact of having a single node acting as a bridge between clusters may become a bottleneck. SARP is based on the FDVB concept where the ordinary nodes run reactive routing protocols while the smart nodes abstract the network and run an hybrid routing protocol (i.e. reactive together with proactive routing).

Each node interested and capable of becoming cluster head (i.e. smart node) will create its own cluster and will try to become part of the FDVB. SARP does not define any cluster selection logic that forces the nodes to become cluster heads depending on their location (i.e. in the centre of the cluster) or other metrics. SARP algorithm allows the nodes to become cluster heads just based on their resources availability. A node can measure the environment (i.e. local traffic, channel utilisation) and based on its available resources decides to become a cluster head or not. Therefore, there is no network wide logic for selecting the cluster heads. Instead, any node can become a cluster head at any point in time. The nodes have the possibility to become cluster heads (i.e. smart) randomly and they can fall back and act as cluster nodes (i.e. ordinary) after exhausting some of their resources. Thus, smart nodes have enough resources and willingness to maintain route and service information. Ordinary nodes are devices with limited resources, running an Ad hoc MANET [33] protocol with low complexity and computational requirements (i.e. a reactive protocol such as AODV).

Only the nodes that become cluster heads (i.e. smart nodes) will engage in additional control transactions for exchanging cluster information. The FDVB is composed of the smart nodes that exchange link state information between them in order to share the network topology information using a proactive protocol such as OLSR, DSDV or a reactive protocol such AODV with new extension messages.

The cluster is set up by the TTL and all the nodes that are close to the cluster head (i.e. nodes within TTL=1 or 2) will be just ordinary nodes. SARP does not impose any additional requirements to the ordinary nodes and they perform reactive routing and packet forwarding functionality as usual. In the same area we can have several smart nodes each of them controlling its own cluster, thus the clusters can overlap and the ordinary nodes can be part of multiple clusters. This leads into a fully distributed cluster creation that will benefit the ordinary nodes. A cluster head will receive a route request from a cluster node, and if the cluster head has the route information available, it will return a route response to the cluster node. If the route information is not available in the cluster head, it will initiate a request to other cluster heads in the FDVB reaching all clusters.

Figure 33 shows the concept of a fully distributed virtual backbone, where several cluster heads are randomly distributed forming a FDVB.

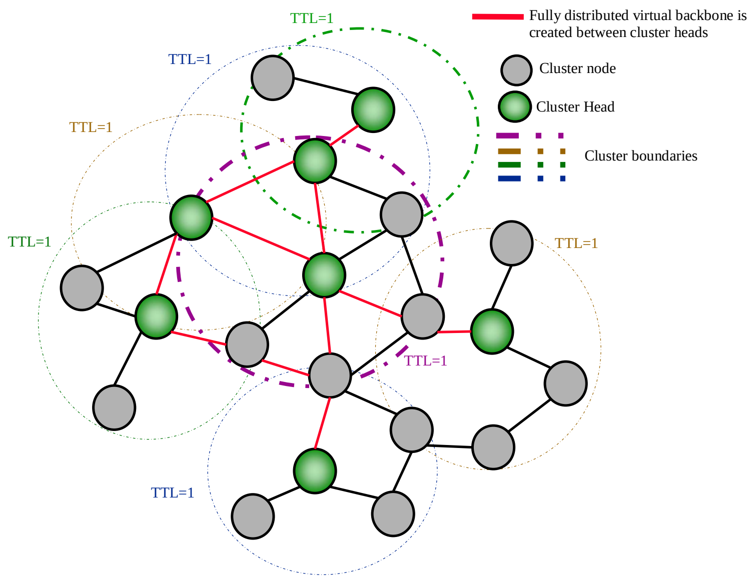


**Figure 33. Fully distributed virtual backbone created with multiple cluster heads.**

SARP is an alternative approach to existing hybrid routing protocols such as the Cluster head-Gateway Switching Routing (CGSR), the Hierarchical State Routing (HSR) or the Zone Routing Protocol (ZRP).

CGSR is a proactive routing protocol that uses the Least Cluster head Change (LCC) algorithm to partition the network into clusters. In addition to the proactive routing overhead, LCC introduces some additional overhead and complexity in the creation and maintenance of the clusters.

HSR is another proactive routing protocol that defines different layers where the cluster heads maintain two hierarchies each of them with two instances of the proactive routing protocol. With the first instance of the proactive protocol the cluster head maintains the topology of the cluster nodes in the neighbourhood. The cluster head uses the second instance to maintain topology information with other cluster heads from the neighbour clusters. HSR presents additional overhead of maintaining two instances of the proactive routing protocol.

ZRP is quite similar to our SARP proposal but still there are a few differences. ZRP specifies the logic for selecting which nodes act as cluster heads and which ones act as border nodes (i.e. gateways between clusters). SARP is based on the concept of the FDVB where the logic for the nodes to become cluster heads is based on their resources, and the nodes themselves decide whether they are capable of becoming cluster heads. SARP does not specify border nodes and instead all smart nodes act as border nodes. ZRP specifies the Intra-Zone Routing Protocol (IZRP) and the Inter-zone Routing Protocol (IERP). IZRP implements a proactive routing protocol used by all the nodes within the zone. IERP implements a reactive routing protocol used by the cluster head and the border nodes for routing purposes between clusters.  SARP use a reactive routing protocol within the cluster nodes and proactive routing protocol between cluster heads.

The question is why another hybrid routing protocol is needed. Based on the simulations we deduced that reactive routing protocols behave more efficiently within small networks. Therefore, reactive routing protocol would be enough for most of the cases, however when the network size increases reactive protocols are not scalable. Thus, we need to form some grouping or clusters to virtually simulate small networks but that means additional complexity requiring additional efforts from all the nodes. This decreases the efficiency of the reactive routing protocols and exhausts the node resources. Thus, the best approach is to keep most of the nodes running an efficient reactive protocol within a small area, and let smart nodes perform the clustering to support network scalability. The selection of the cluster heads does not affect the rest of the nodes, so the additional clustering complexity should be minimised and hidden from the ordinary nodes.

Based on the results from the simulations and the test bed SARP has been proposed to fix some of the drawbacks of reactive, proactive and hybrid routing protocols [15]. A mathematical model to evaluate the network performance with SARP is defined in Chapter 3.

# Chapter 3

## Performance Modelling of the Hybrid Routing Approach

In this Chapter we use the propositions formulated based on the simulations and the test bed, and define a mathematical model for evaluating the performance of the hybrid routing approach (i.e. SARP) in Ad hoc networks. We analyze a generic model for Ad hoc networks and top of that we apply the Fully Distributed Virtual Backbone (FDVB) concept to validate the hybrid routing approach and demonstrate the improvement of the overall network scalability. We use the model to study the network impact of having ordinary and smart nodes in an Ad hoc network and to identify the optimal number of smart nodes from performance point of view. Based on the results, we introduce the algorithm that the smart nodes have to implement in order to set up an optimal FDVB. Additional simulation results of AODV, OLSR and SARP are presented.

The performance in Ad hoc networks cannot be easily modelled due to the amount of variables and the uncertainty of their values. In the literature there are several attempts to provide a performance analysis of Ad hoc networks based on an imprecise network state model [59]. The existing models are not reliable due to the unpredictable behaviour of the nodes. An accurate attempt to model the Ad hoc networks performance should consider the nodes mobility and unpredictability of the network conditions. However, our objective is to validate the SARP protocol in terms of the impact on scalability rather than define an accurate Ad hoc network model. Therefore, we will define a generic Ad hoc network model and apply the Fully Distributed Virtual Backbone (FDVB) concept on top of it.

Variables such as the location of the smart nodes within the network are relevant but, in order to simplify the model, we will consider an area of the network where the smart nodes appear randomly and remain stable there for a certain period of time.

Our main objective is to prove that network scalability increases when we apply the FDVB concept, and to determine the density of smart nodes required to build an optimal FDVB independently of their location. For this purpose we define the *smart nodes access control algorithm*.

In Ad hoc networks the nodes exhaust their resources because they perform packet forwarding and routing functions that in fixed networks are normally implemented in static servers or routers. In order to define a generic Ad hoc network model, we will identify the metrics required to evaluate the performance.

### 3.1 Performance Metrics in Fixed Networks

Fixed networks are modelled as graphs $G(N, A)$ where *N* is the set of nodes and *A* is the set of arcs in the network [40]. The arcs are denoted as (*i*, *j*) representing the communication link between nodes $n_i$ and $n_j$. A scalar value $x_{ij}$ represents the flow between nodes *i* and *j* through the arc (*i*, *j*). In a graph $G(N, A)$ the set of flows $x_{ij}\big|_{(i,j)\in A}$ is referred to as the flow vector. A path *P* in a graph is a sequence of arcs $P_{1,k} \equiv (1,2,...k)$ where $k \geq 2$. A graph is connected if for each pair of nodes *i* and *j*, there is a path starting at *i* and ending at *j*.

The routing algorithms calculate the optimal routes obtaining paths where the flow vectors $x_{ij}$ are constrained between given lower and upper bounds (i.e. $b_{ij} \leq x_{ij} \leq c_{ij}$) in order to limit the available bandwidth for that flow.

In fixed networks, stability is good enough and the end to end packet delay and throughput capacity are the only metrics to be optimised. These metrics are known in fixed networks providing a NP-complete performance model that can be solved using some approximations.

The routing algorithms in fixed networks aim to find a path that connects source and destination nodes through a set of arcs that minimize a linear cost function $\sum_{(i,j)\in A} a_{ij} x_{ij}$ where $a_{ij}$ denotes for example the average packet delay to cross the arc $(i,j)$. The shortest path is the path with minimum average delay that can be used for packet forwarding. Therefore, we can model performance in fixed networks using Eq 56.

**Eq 56.** $f(a_{ij}) \equiv \min \sum_{(i,j)\in A} a_{ij} x_{ij}$

In order to enhance network performance, the generic shortest path algorithms used in fixed networks try to maintain and adjust a vector $(d_1, d_2, ..., d_N)$, where each $d_i$ is the node label and can be either a scalar or $\infty$.

Let $d_1, d_2, ..., d_N$ be scalars satisfying $d_j \le d_i + a_{ij}, \forall (i,j) \in A$ **(a)**

and let $P$ be a path starting at a node $n_i$ and ending at a node $n_j$,

if $d_j = d_i + a_{ij}$ for all arcs $(i, j)$ of $P$ then $P$ is the shortest path from $i$ to $j$. **(b)**

Where **(a)** and **(b)** are called the Complementary Slackness (CS) [40] conditions for the shortest path problem.

The routing algorithms use the CS conditions to calculate the shortest path. These algorithms select successively the arcs that violate the CS condition, meaning $d_j > d_i + a_{ij}$. If an arc that violates CS is found, the routing algorithms will set $d_j := d_i + a_{ij}$ and continue the processing through the available arcs until the CS condition $d_j \le d_i + a_{ij}$ is satisfied for all the arcs $(i, j)$ in the path. The routing algorithms reiterate the calculation over an existing graph and if they terminate then there is a node $j$ with $d_j < \infty$. This means that $d_j$ is the shortest distance with minimum delay (i.e. based on the cost $a_{ij}$ assigned to each arc) from $i$ to $j$. If the algorithm does not terminate, then a node *j* exists such that all sequences of paths that start at *i* and end at *j* will have lengths that diverge to -∞. The algorithm terminates if and only if there is no path that starts at *i* and contains a cycle with negative length.

In fixed networks, connectivity (i.e. probability of having active links) seldom changes. However, in Ad hoc networks connectivity and many other metrics impact network performance. Connectivity in the path between source and destination is often lost because links are broken due to node mobility. Ad hoc networks cannot rely on fixed routes and the frequent topology changes can make connectivity close to zero. Thus, network performance optimization cannot be solved within a limited processing time.

### 3.2 Performance Metrics in Ad hoc Networks

In fixed networks, performance is modelled with one equation that will be minimized by the routing algorithm. However, in Ad hoc networks several metrics will affect the performance independently and there is no single equation that considers all the metrics. A nontrivial problem like this can be resolved by approximation, heuristics or probabilistic methods. Thus, we need to identify the metrics and variables with a major impact on Ad hoc networks performance and define the relationship between them. To simplify the resolution, we will first find and compare the values of the variables that optimise the performance for each metric separately. After that we will select those values that give the best performance in all metrics.

Table 5 represents the basic variables in the Ad hoc network model.

**Table 5. Ad hoc network model basic variables.**

| Number of nodes | Node mobility | Number of hops |
|---|---|---|
| N | M | $\gamma$[1] |

[1] $\gamma$, represents the number of hops in the path as identified in the test bed.

Node mobility and the number of hops are variables that can be considered linear (e.g. nodes mobility can vary between 0-10m/s and the number of hops $\gamma$ depends on the selected path). The number of nodes is a critical variable for measuring the Ad hoc network scalability so we will analyze its impact.

We consider that the probability of nodes joining the Ad hoc network follows a Poisson arrival time distribution (Eq 57) where $\lambda$ is the average number of node arrivals in a given time interval *t* and *f*(*k*) is the probability of having *k* nodes in a given time.

**Eq 57.** $f(k, \lambda t) = \frac{e^{-\lambda t}(\lambda t)^k}{k!}$

In the FDVB concept we defined two types of nodes, ordinary and smart. The initial assumption is that the nodes do not earn incentives to become smart and implement SARP routing functionality. In this case we assume that the nodes select randomly with equal probability to be either ordinary or smart $p(t)\big|_{node=ordinary} = p(t)\big|_{node=smart} = 0.5$. Thus, $p_{S_a}(k, \lambda t)$ in Eq 58 represents the smart nodes arrival time distribution considering Eq 57 and the probability to be smart $p(t)\big|_{node=smart} = 0.5$.

**Eq 58.** $p_{S_a}(k, \lambda t) = 0.5\frac{e^{-\lambda t}(\lambda t)^k}{k!}$

The smart nodes may exhaust their battery after some time in the network and become ordinary or die. The battery consumed by a node is modelled using Peukert equation (Eq 59). The consumed battery capacity ($C_b$) increases with the time (*t*, *hour*) depending on the discharge current (*I*, Amperes) and the Peukert constant (*n*=1.1 or 1.2 typically).

**Eq 59.** $C_b = I^n t(Ampere * hour)$

The residual battery capacity in a node is $C_r = C_t - C_b$ where $C_t$ is the full capacity of the battery. Based on the residual battery, we can model the node death process with an exponential $p_d(t) = e^{-\partial t}$ where the slope, $\delta$, depends on the battery age and the processing consumption on each node among other variables. Nevertheless, we consider that all nodes have similar battery age but the processing consumption will be higher in smart nodes due to their participation in the SARP routing functionality.

Figure 34 represents the battery consumed by ordinary (n=1.1) and smart nodes (n=1.15) besides their residual battery capacity. The equations associated with the residual battery capacity for $C_r(n=1.1)$ and $C_r(n=1.15)$ with the lowest approximation error $r^2=0.95$ and $r^2=0.97$ result in exponentials with $\partial(1.15)=-0.0144$ and $\partial(1.1)=-0.0097$ slopes respectively. Thus we assume that the smart nodes slope is approximately $\partial_S \approx 0.015$ while it is $\partial_O \approx 0.01$ for ordinary nodes.

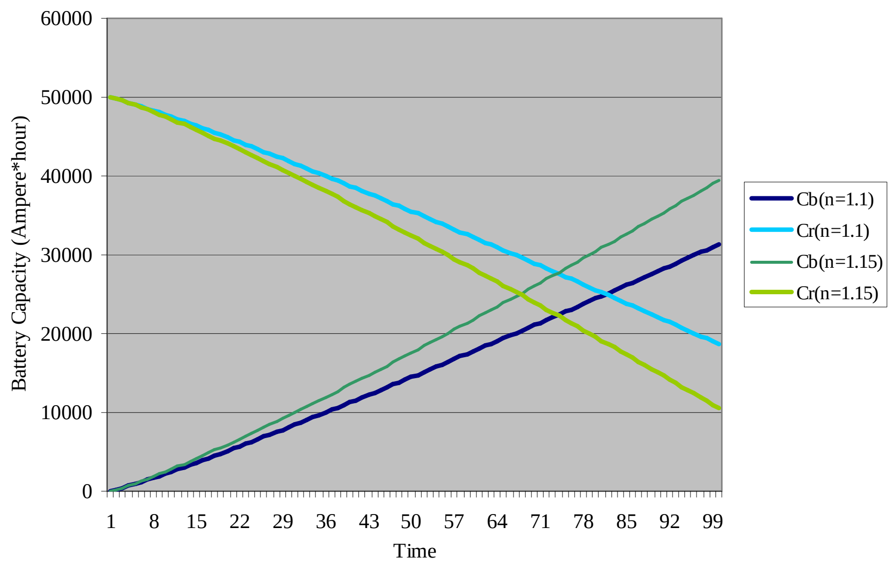


**Figure 34. Consumed and residual battery capacities in smart and ordinary nodes.**

We assume that the nodes arrival and death processes are independent. The $p_s(t)$ in Eq 60 represents the probability of having smart nodes in the network. In Eq 60 we consider that initially the number of smart nodes that are part of the FDVB is high but after a period of time the nodes exhaust their resources and the smart nodes death is not compensated with the new smart node arrivals. We also consider as new node arrivals those smart nodes that exhaust their batteries and become temporarily ordinary since the node may become smart again after re-charging the battery.

**Eq 60.** $p_s(t) = p_{S_a}(t)p_{S_d}(t) = e^{-\partial_S t}\sum_{t=0}^{t} 0.5\frac{e^{-\lambda t}(\lambda t)^k}{k!}$

Figure 35 shows $p_{S_a}(t)$ as the smart node arrival cumulative probability (i.e. considering an average node arrival of λ=5 nodes and equal probability to become smart or ordinary). $p_{S_d}(t)$ represents the smart node survival probability and $p_S(t)$ the probability of having smart nodes left in the network. Figure 35 shows that if we consider only the Poisson distribution of arrivals, it will result in the probability of having a constant share of smart nodes in the network as represented with $p_{S_a}(t)$. After adding the node survival probability due to battery consumption $p_{S_d}(t)$ the probability of having smart nodes in the network $p_S(t)$ after reacting an initial peak level decreases over time.

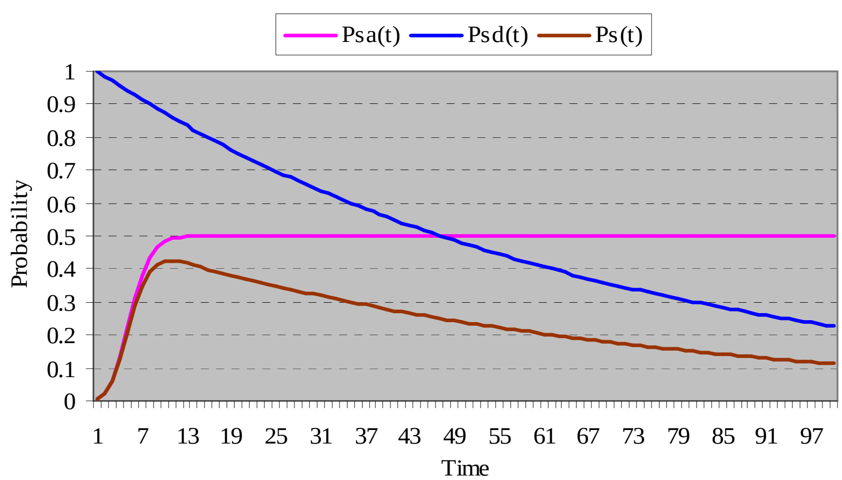


**Figure 35. Probability of arrival, death and smart nodes left in the Ad hoc network.**

The metrics under study to model the scalability of Ad hoc networks are represented in Table 6.

**Table 6. Ad hoc network model metrics.**

| Connectivity | Bandwidth | End to end packet delay | Percentage of packet loss | Jitter |
|---|---|---|---|---|
| C | B | D | L | J |

We focus on real time communications, which require an end to end packet delay below 200ms and a percentage of packet loss lower than 5%.

The metrics can be grouped based on how they affect the Ad hoc network performance. Performance can be modelled using multiplicative, $m(i, j)$, concave, $c(i, j)$, and additive, $a(i, j)$, groups of metrics. Connectivity and packet loss can be considered multiplicative, bandwidth is concave and end to end packet delay and Jitter are additive. We will obtain the equation that defines the relationship between each group of metrics and the Ad hoc network model basic variables presented in Table 5.

We start the analysis defining a theoretical function to model the performance based on the multiplicative metric of a path between source and destination nodes. In this first step we obtain a performance equation that depends on a single metric, which makes the routing analysis tractable. With this equation we obtain a list of optimal routes similarly to the routing algorithm used in fixed networks. Afterwards, a theoretical function is defined for the concave metric and from the list of optimal routes obtained for the multiplicative metric we select the ones that provide also the optimal values for the concave function. In the last step a new equation that models the additive metric is defined and the remaining routes are prioritised based on the values obtained from the additive function.

### 3.2.1 Multiplicative Metric of the Ad hoc Networks Model

Connectivity can be modelled as a multiplicative metric, $m(i, j)$, since it is defined as the probability of having active links leading to a successful packet delivery through all the links on the path. It has a critical impact on the Ad hoc network performance. If connectivity is null, the rest of metrics are irrelevant.

Connectivity strongly depends on mobility of all the nodes in the path. Node mobility can break a link in the path without time to update the network topology. Connectivity is inversely proportional to the percentage of packet loss, *L*. Thus, the percentage of the packet loss will be measured based on the results from the connectivity metric.

The connectivity $c_{kl}$ of a link (*k*, *l*) is the probability that the link is active in a communication network. Each link is active independently of the other links. Thus, the connectivity of nodes *i*, *j* $C(i, j)$ is the product of the connectivity of the links $(i, l_1), .....(k_n, j)$ on the path from *i* to *j*.

$$C(i, j) = m(i, j) = m(i, l_1) * m(k_1, l_2) * m(k_2, l_3) * ... * m(k_n, j) = c_{i,l_1} * c_{k_1,l_2} * c_{k_2,l_3} * ... * c_{k_n,j}$$

The routing algorithm has to find a path with the maximum value for the equation Eq 61.

**Eq 61.** $m(i, j) = \prod_{k=i}^{l=j} c_{kl} \forall k, l$

*n* is the number of links on the path such that links $(i, l_1), .....(k_n, j)$ form a path from *i* to *j* and $c_{kl}$ is the connectivity of the link (*k*,*l*) which depends on the mobility $M_{kl}$.

If $M_{kl} \rightarrow 0$ then $\lim_{M_{kl} \rightarrow 0} m(i, j) = 1$ and if $M_{kl} \rightarrow M_{\max}$ then $\lim_{M_{kl} \rightarrow M_{maz}} m(i, j) = 0$.

Based on Eq 61 and the limits, we can model the link connectivity as an exponential function Eq 62 that depends on the nodes relative mobility $M_{kl}$.

**Eq 62.** $c_{kl} = c_O e^{-\alpha M_{kl}}$

$c_O$ is the connectivity of the link (*k*, *l*) when the mobility is zero ($M_{kl} = 0$) and $\alpha$ is the slope factor representing the dependency from mobility of the connectivity function.

The maximum link connectivity between two nodes *k* and *l* is obtained when both are completely static ($M_{kl} = 0$) that rarely happens.

$$\lim_{M_{kl} \to 0} c_{kl} = 1 \Rightarrow c_0 e^{-\alpha M_{kl}} \Big|_{M_{kl}=0} = c_O \Rightarrow c_O \approx 1$$

The minimum link connectivity is reached when the nodes *k* and *l* are moving ($M_{kl} = M_{\max}$).

$$\lim_{M_{kl} \to M_{\max}} c_{kl} = 0 \Rightarrow c_0 e^{-\alpha M_{kl}} \Big|_{M_{kl}=M_{\max}} \approx 0$$

The connectivity will be null when the mobility is ∞ ($e^{-\alpha M_{\max}} = 0 \Rightarrow \alpha M_{\max} \to \infty$). This scenario is not feasible in practice but we consider that the probability of connectivity is almost null in high mobility conditions.

The aim of the FDVB architecture under study is to improve the connectivity by introducing nodes with enough resources and low mobility (i.e. smart nodes). These nodes will support the nodes with limited resources and higher mobility (i. e. ordinary nodes) in terms of routing functionality. The smart nodes will reduce the routing latency, find the optimal routes and also provide more stability where they are part of the routes.

The link connectivity between two smart nodes is higher than between two ordinary nodes ($c_{kl}\big|_{k,l=Smart} > c_{kl}\big|_{k=Ordinary,l=Smart} > c_{kl}\big|_{k,l=Ordinary}$). Thus, connectivity will increase with the introduction of smart nodes on the path. Eq 63 represents the link connectivity between two smart nodes.

**Eq 63.** $c_{kl}\big|_{k,l=Smart} = c_{O_S} e^{-\alpha_S M_{kl}}$

$c_{O_S}$ is the connectivity of a link (*k*, *l*) between smart nodes when mobility is zero ($M_{kl} = 0$), and $\alpha_S$ is the slope factor representing the dependency with mobility in the connectivity function of a link (*k*, *l*) between smart nodes.

Applying the FDVB concept on top of the generic Ad hoc network model the multiplicative metric is represented by Eq 64.

**Eq 64.** $$m_F(i,j) = \prod_{k,l \in O} c_O e^{-\alpha M_{kl}} * \prod_{k \in O, l \in S\_or\_k \in S, l \in O} c_O e^{-\alpha_{So} M_{kl}} * \prod_{k,l \in S} c_{O_S} e^{-\alpha_S M_{ij}} \forall k,l$$

$c_O$ is the connectivity of a link (*k*, *l*) when the nodes mobility is zero ($M_{kl} = 0$).

$c_{O_S}$ is the connectivity of a link (*k*, *l*) between smart nodes when the nodes mobility is zero ($M_{kl} = 0$).

$n$ is the number of links (*k*,*l*) on the path (*i*,*j*).

$\alpha$ is the slope factor representing the dependency with mobility in the connectivity function of the link (*k*, *l*) between ordinary nodes.

$\alpha_S$ is the slope factor representing the dependency with mobility in the connectivity function of a link (*k*, *l*) between smart nodes.

$\alpha_{So}$ is the slope factor representing the dependency with mobility in the connectivity function of a link (*k*, *l*) between a smart and an ordinary node.

$M_{kl}$ is the relative mobility of the nodes in the link (*k*, *l*).

Eq 60 shows that when the smart nodes energy decreases the probability of having smart nodes left in the network decreases. Therefore, the connectivity in Eq 64 will decrease. Increasing the number of hops in the path decreases the connectivity regardless the number of nodes in the network. Therefore, a small number of hops and smart nodes in the path will improve the connectivity in Ad hoc networks providing the highest value of the multiplicative metric.

### 3.2.2 Performance Simulation Based on the Multiplicative Metric

Once we have obtained the equations for modelling the connectivity as a multiplicative metric, we compare the results to evaluate the performance difference between the generic and the FDVB Ad hoc network models.

We set $c_0 \approx 0.7$ as the value for the connectivity in Ad hoc networks with ordinary nodes assuming static conditions ($M_{kl} = 0$). The connectivity decreases with mobility so taking as reference the equation that models the packet loss in reactive routing protocols, Eq 39. $L_{Ru}(M,N) \approx (0.01 + 0.03N)e^{(1.34+0.35N)M}(\%)$, we set $\alpha \approx 1.34$ as the slope factor for the ordinary nodes.

We set $c_{O_S} \approx 0.9$ as the value for the connectivity in Ad hoc networks with smart nodes assuming static conditions ($M_{kl} = 0$). The connectivity between smart nodes decreases with mobility so taking as reference the equation Eq 44. $L_{Pu}(M,N) = (0.09 + 0.13N)e^{(0.69+0.012N)M}(\%)$ that models packet loss in proactive routing protocols, we set $\alpha_S \approx 0.69$ as the slope factor for the smart nodes.

Figure 36 shows the results of the connectivity probability on paths with 2 hops in five scenarios. Each scenario considers a different percentage of smart nodes in the network (i.e. $p_s(t)$=1; 0.7; 0.5; 0.3; 0). In all these scenarios, we vary the mobility from 0m/s up to 4m/s with 0.5m/s increments (each of them represented with a different curve). The curve on the top represents the highest connectivity obtained when the mobility is 0m/s while the curve on the bottom represents the lowest connectivity obtained when the mobility is 4m/s. The results in Figure 36 show that the connectivity probability decreases when the percentage of smart nodes is low. However, when 50% of nodes are smart (S=50%) and 50% are ordinary (O=50%) the connectivity probability is similar to the scenario where all the nodes are ordinary (O=100% and S=0%). A low percentage of smart nodes (O=70% and S=30%) does not improve much the connectivity probability because it is mostly provided by the ordinary nodes.

Figure 36 shows that when the percentage of smart nodes is higher than the percentage of ordinary nodes, the connectivity probability is affected by mobility. Thus, in the scenario with 100% of smart nodes (S=100%) and non static conditions (i.e. second curve from the top in red represents mobility M=0.5m/s) the connectivity probability is lower than in the scenario with 100% of ordinary nodes (O=100%) and static conditions. The time units are not shown in the figure because time represents the total network lifetime. As we can observe in the value of the connectivity probability for each scenario in the simulation, the network lifetime is much shorter in the scenario with smart nodes only than with ordinary nodes regardless of mobility. At the end of the network lifetime (i.e. t=100) the scenario with ordinary nodes only (O=100%) is the one with the highest connectivity probability value (i.e. C=0.068%).

We can conclude that in Ad hoc networks with low mobility, a higher percentage of smart nodes than ordinary nodes in the path increases the connectivity but reduces the network lifetime. On the other hand, in high mobility conditions, a higher percentage of ordinary nodes than smart nodes increases the connectivity and extends the network lifetime. From the connectivity point of view, the benefit of a high percentage of smart nodes is considerable when the nodes mobility is low and they can guarantee stable routes. If the nodes mobility is high, the fact of being smart does not improve the connectivity because proactive routing might provide routes that are stale because of the nodes mobility.

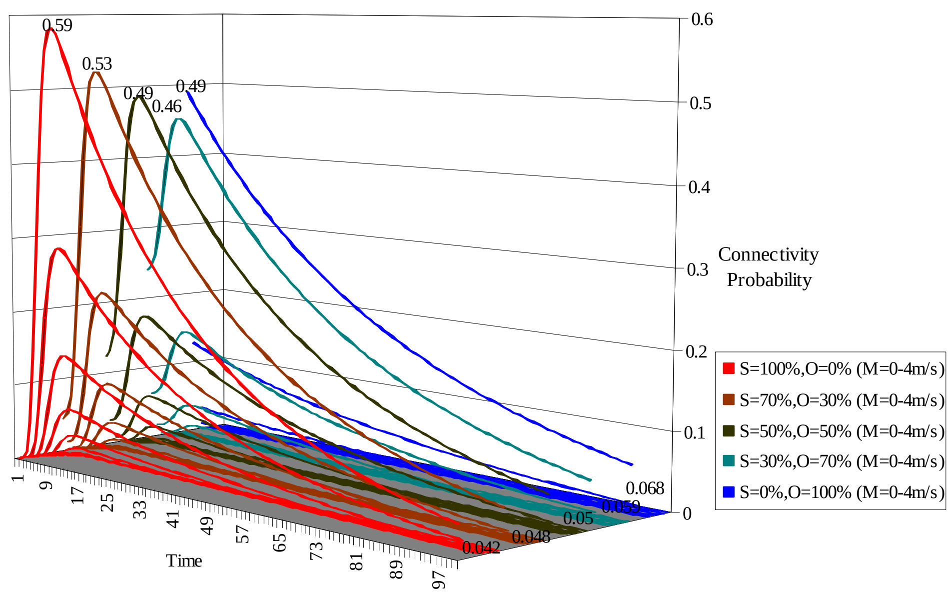


**Figure 36. Connectivity probability on routes with 2 hops in five different scenarios.**

### 3.2.3 Concave Metric of the Ad hoc Network Model

The path bandwidth is the minimum available bandwidth in any of the links $(i,l_1),.....(k_n,j)$ on the path from node *i* to node *j*.

$$b(i,j) = \min\{b(i,l_1), b(k_1,l_2), b(k_2,l_3),..., b(k_n,j)\}$$

The optimal bandwidth metric *B(i, j)* is the maximum available bandwidth on the paths from node *i* to node *j*. It is modelled as a concave metric because its maximum value is the minimum available bandwidth in any of the links on the path.

$$B(i,j) = \max\{b(i,j)\} = \max\{\min\{b(i,l_1),..., b(k_n,j)\}\}\forall k,l$$

Throughput is directly proportional to the available bandwidth and inversely proportional to the routing overhead $\Omega(i,j)$ which decreases the available bandwidth for data transmission. We consider an Ad hoc network scalable, if the performance metrics do not change when the number of nodes increases. Thus, the available bandwidth for data transmission is directly proportional to network scalability and inversely proportional to the routing overhead generated to keep the same connectivity while increasing the number of nodes.

The bandwidth metric decreases with the number of active connections that node *i* and *j* maintain with their neighbours because they share the same channel. For simplicity we assume that regardless of the number of active connections, the available bandwidth ($B_T$) on each node is $B_T \equiv \frac{B_N}{n_e}$ where $n_e$ is the number of neighbours and $B_N$ is the nominal bandwidth provided by the wireless technology. The available bandwidth in the Ad hoc network depends on the selected wireless technology (e.g. 802.11b: 11Mbs, 802.11a: 54 Mbs). Moreover, the available bandwidth on each link (*k,l*) in the path from node *i* to node *j* can be modelled with Eq 65 which is equal to the available bandwidth on the node, $B_T \equiv \frac{B_N}{n_e}$ minus the routing overhead on each link $\Omega(k,l)$.

**Eq 65.** $b(k,l) = \frac{B_N}{n_e} - \Omega(k,l)$

In order to maximize the available bandwidth on any of the links on the path we have to find the percentage of smart nodes that minimizes the routing overhead. The equations Eq 66 and Eq 67 represent the concave metric for the generic Ad hoc network model for reactive and proactive routing respectively. Where $\Omega_R(M,N)$ and $\Omega_P(M,N)$ represent the reactive and proactive routing overhead respectively.

**Eq 66.** $B_R(i,j) = \max\{b_R(i,j)\} = \max\left\{\min\left\{\frac{B_N}{n_e} - \Omega_R(M,N)_{kl}\right\}\right\} \forall kl \in Path$

**Eq 67.** $B_P(i,j) = \max\{b_P(i,j)\} = \max\left\{\min\left\{\frac{B_N}{n_e} - \Omega_P(M,N)_{kl}\right\}\right\} \forall kl \in Path$

Based on the simulation results we approximated the reactive routing overhead in Eq 20-Eq 21 and the proactive routing overhead in Eq 25-Eq 26. However, the test bed proved that those equations were not accurate and that the number of hops had an impact on the routing overhead. Therefore, we will define new equations to approximate the routing overhead using some of the results from the simulations but also from the test bed. We assume that reactive and proactive protocols increment the routing overhead exponentially with mobility because when the links break, the route recovery control messages are triggered mainly in reactive routing protocols. When mobility is zero, the routing overhead is minimum as in fixed networks, which leads to the following equation.

$$\Omega(M, N, \gamma) = f(N, \gamma) e^{\alpha M_{kl}}$$

N is the number of nodes in the Ad hoc network.

M is the node mobility.

$\gamma$ is the number of hops on the path.

$\alpha$ is the slope factor representing the dependency from mobility.

We consider the following limits for the routing overhead.

If $M_{kl} \to 0$ then $\lim_{M \to 0} \Omega(k,l) = A$ and if $M_{kl} \to M_{\max}$ then $\lim_{M \to \infty} \Omega(k,l) = \infty$

*A* is a constant value equivalent to the routing overhead with zero mobility.

In the test bed the mobility variable was zero. Thus, we assume that the slope factors representing the dependency with mobility are still valid, and we model the routing overhead with the values from Eq 21 and Eq 26.

Eq 21: $\Omega_{Ru}(M, N) = (155.4 + 5.1N) e^{(0.045+0.009N)M} \; (Kbytes)$; mobility is affecting with the slope factor of 0.045.

Eq 26: $\Omega_{Pu}(M, N) = (615 + 375N) e^{(0.001+0.002N)M} \; (Kbytes)$; mobility is affecting with the slope factor of 0.001.

Next we will identify the rest of the parameters in equations $f_R(N,\gamma)$ and $f_P(N,\gamma)$ to represent more accurately the routing overhead according to the test bed results. Let us first consider the impact of number of hops and nodes $f_P(N,\gamma)$ that represents the proactive routing overhead based on the number of nodes and hops. In proactive routing, the routing overhead is affected mainly by the number of nodes and not by the number of hops $f_P(N,\gamma) \equiv f_P(N)$, since each node has to exchange periodically topology information with the neighbours. We define $Q_P$ as the route updates per second that the nodes running proactive routing protocols have to send to their neighbours. The route update will contain the entire routing cache that includes the topology information from all the available nodes. We define a variable, $W_P$(N) that represents the bytes per route update. $W_P$(N) is represented as $W_P$(N)= K+4N including the fixed protocol information (i.e. K) plus a minimum of 4 bytes of link information (i.e. IP address: 4-byte, number of hops:1-byte, etc) associated with each node N in the network. Thus, the routing overhead per node in proactive protocols can be modelled with Eq 68.

**Eq 68.** $f_P(N) = (N-1)Q_P W_P(N)$

In order to evaluate the accuracy of Eq 68, Table 7 compares the values obtained from the test bed with the values obtained from the equation after replacing the variables with the values used in the test bed.

Table 7 shows that the values from the model equation are similar to the ones obtained from the test bed so we can conclude that the model equation Eq 69 accurately represents the OLSR routing overhead in real Ad hoc networks.

**Eq 69.** $\Omega_P(M,N,\gamma) = f_P(N,\gamma)e^{\alpha M_{kl}} = (N-1)Q_P W_P(N)e^{0.001M_{kl}}$

**Table 7. Proactive routing overhead comparison between the test bed and the model equation.**

| Performance metrics | OLSR/1hop | OLSR/2hops | OLSR/3hops |
|---|---|---|---|
| **N number of nodes** | 2 | 3 | 4 |
| **$Q_P$ (route updates/s)**[1] | 0.4 | 0.4 | 0.4 |
| **$W_P$ (bytes/route update)**[2] | 60+4*N=68 | 60+4*N=72 | 60+4*N=76 |
| **Routing overhead (model equation)** | | | |
| Routing overhead (bytes/s) | 27.2 | 57.6 | 91.2 |
| **Routing overhead (test bed)** | | | |
| % routing overhead (Routing packets/RTP packets) | 3.39 % (109/3215) | 3.86% (181/4688) | 3.58% (286/7969) |
| Routing overhead (bytes/s) | 32.3 | 54.2 | 88.4 |

[1] OLSR sends 0.2 TC updates/s and 0.5 Hello messages/s

[2] OLSR has 60bytes of fixed protocol info in the TC updates/s and the Hello messages/s

Let us now consider $f_R(N,\gamma)$ as the routing overhead for reactive routing based on the number of nodes that receive the route request and the number of hops. Each node in the network will send a route discovery broadcast when the route is not available in the routing cache. We define $Q_R$ as the number of requests per second that each ordinary node issues to find new routes. The route request message includes only the required information, $W_R$ bytes, to find the destination. If the node does not receive a response to the request within a certain time, it will increase the TTL and send again the same route request that will reach new nodes several hops away from the originating node.

In case of AODV the route request process starts with TTL=1 and if no response is received the source node will increment the TTL by 2 and will resend the route request with TTL=3. If no response is received, a new route request will be sent incrementing the TTL by 2 (i.e. TTL=5). The route request process is repeated until the maximum of TTL=7 is reached.

The number of route requests to be sent increases with the number of hops $\gamma$. If the nodes receiving each route request do not have the address of the destination node nor have they seen the route request before, they will issue a new route request increasing the routing overhead in the network. Thus, the routing overhead depends on the number of hops between the source and the destination. In order to measure this effect, we define $n_\gamma$ as the average number of neighbours in the network within each hop $\gamma$, from the originating node.

The originating node will launch several attempts to find the destination until either a node responds with the route to reach the destination node or no route is found and a node not reachable error occurs so, the overhead created is:

$$Round\_1: f_R(n_\gamma,1)\big|_{TTL=1} = Q_R W_R$$

$$Round\_2: f_R(n_\gamma,3)\big|_{TTL=3} = f_R(n_\gamma,1)\big|_{TTL=1} + Q_R W_R n_1 + Q_R W_R n_2$$

$$Round\_3: f_R(n_\gamma,5)\big|_{TTL=5} = f_R(n_\gamma,3)\big|_{TTL=3} + Q_R W_R n_4 + Q_R W_R n_5$$

$$Round\_4: f_R(n_\gamma,7)\big|_{TTL=7} = f_R(n_\gamma,5)\big|_{TTL=5} + Q_R W_R n_6 + Q_R W_R n_7$$

The total overhead generated from any node will depend whether the destination node is found close to the originating node or additional request with higher TTL is needed to reach the destination node. If the destination node is far away, the number of nodes that receive the route request on each hop, $n_\gamma$, will retransmit the route request causing a flooding explosion in the network as modelled in Eq 70. In Eq 70 we ignore the extra routing from the route reply (i.e. RREP) that all the neighbours that have a route to the destination node will send to the originating node. The routing overhead depends on the probability of having the destination node within a certain number of hops away from the originating node. The reactive routing overhead depends on $\gamma$ and the number of nodes on each hop from the originating node $n_\gamma$, so $f_R(N,\gamma) = f_R(n_\gamma,\gamma)$. If the destination node is found odd number of hops from the originating node, then TTL = γ and if it is an even number of hops, then the TTL = γ+1. We consider $\rho$ the number of rounds needed to reach a destination at the distance of γ hops, then the total overhead is modelled in Eq 70.

**Eq 70.** $f_R(n_\gamma,\gamma) = Q_R W_R\left[\rho + (\rho-1)(n_1+n_2) + (\rho-2)(n_3+n_4) + (\rho-3)(n_5+n_6)\right]$

where negative terms of the sum are capped to zero and $n_\gamma$ is the number of nodes being exactly $\gamma$ hops away from the originating node. With AODV the farthest we can reach are nodes that are at most exactly 7 hops away from the originating node. Eq 70 is pessimistic in the sense that we ignore the possibility that some intermediate node on a path to the destination may have a valid route to the destination when a route request reaches it. Nevertheless, despite the intermediate node has a valid route and sends the route reply, the rest of nodes that are not aware of a valid route will receive the route request and will forward it until TTL=0.

In order to evaluate the accuracy of Eq 70, Table 8 compares the values obtained from the test bed with the values obtained from the equation after replacing the variables with values equivalent to the ones used in the test bed.

**Table 8. Reactive routing overhead comparison between the test bed and the model equation.**

| Performance metrics | AODV/1hop | AODV/2hops | AODV/3hops |
|---|---|---|---|
| **$Q_R$ (route request/s)** | 0.7 | 0.7 | 0.7 |
| **$W_R$ (bytes/route request)** [1] | 68 | 68 | 68 |
| **$n_\gamma$** | 1 | 1 | 1 |
| **Routing overhead (model equation)** | | | |
| Routing overhead (bytes/s) | 47.6 | 142.8 | 142.8 |
| **Routing overhead (test bed)** | | | |
| % routing overhead<br>(Routing packets/RTP packets)<br>Routing overhead (bytes/s) | 7.22 %<br>(170/2353)<br>48.1 | 7.38%<br>(506/6858)<br>49.2 | 18.17 %<br>(666 / 3665)<br>121.8 |

[1] 68bytes message size for RREQ messages in AODV, 153bytes message size of RTP messages, 15messages/s.

Table 8 shows that the values from the model equation are similar to the ones obtained from the test bed for 1 and 3 hops. The difference for the case of 2 hops is due to the fact that the route request will have TTL=3 and in the test case there is a single node 2 hops away from the originating node that will provide the RREP so the route request will not be forwarded any further and no additional overhead is generated. On the other hand, the model measures the overall overhead generated with the route request that has TTL=3. Thus, despite the destination node is 2 hops away from the originating node, the route request will be forwarded by other nodes in the network that are not aware of the destination node and similar overhead to the case with 3 hops will be generated. Thus, the results from the test bed for 2 hops would be similar to the results obtained from the model for 1 hop. Therefore, despite the inaccuracy in some specific conditions we can conclude that the model equation represents accurately enough the AODV routing overhead in real Ad hoc networks is:

$$\Omega_R(M,N,\gamma) = f_R(n_\gamma,\gamma)e^{\alpha M_{kl}} =$$
$$\left[Q_R W_R\left[\rho+(\rho-1)(n_1+n_2)+(\rho-2)(n_3+n_4)+(\rho-3)(n_5+n_6)\right]\right]e^{0.045M_{kl}}$$

Using Eq 66, Eq 70 and Eq 21, Eq 71 represents the concave metric (i.e. the bandwidth) in the generic Ad hoc network model where all the nodes are ordinary.

**Eq 71.**
$$B_R(i,j)=\max\{b_R(i,j)\}=$$
$$\max\left\{\min\left\{\frac{B_N}{n_e}-\left(\left[Q_R W_R\left[\rho+(\rho-1)(n_\gamma+n_2)+(\rho-2)(n_3+n_4)+(\rho-3)(n_5+n_6)\right]\right]e^{0.045M_{ij}}\right)_{kl}\right\}\right\}\forall k,l\in Path$$

Using Eq 66-Eq 70, Eq 26 and Eq 60, Eq 72 represents the concave metric in the FDVB Ad hoc network model where there are ordinary and smart nodes in the network.

**Eq 72.**
$$B_F(i,j)=\max\{b_F(i,j)\}=$$
$$\max\left\{\min\left\{\begin{array}{l}\frac{B_N}{n_e}-\left(\left[Q_R W_R\left[\rho+(\rho-1)(n_{1\gamma}+n_{2\gamma})+(\rho-2)(n_{3\gamma}+n_{4\gamma})+(\rho-3)(n_{5\gamma}+n_{6\gamma})\right]\right](1-p_S(t))e^{0.045M_{ij}}\right)_{kl}\\-\left((N'-1)Q_P W_P(\widehat{N})p_S(t)e^{0.001M_{ij}}\right)_{kl}\end{array}\right\}\right\}$$
$$\forall k,l\in Path$$

$N'$ is the number of smart nodes that will exchange topology information. The smart nodes in the FDVB will not maintain the link information from all nodes in the network but only from the nodes they have received route requests, $\widehat{N}$ (i.e. $\hat{N}\subset N$). Thus, the size of the route updates will be proportional to the $\widehat{N}$ number of nodes (i.e. $W_P\left(\widehat{N}\right)=\widehat{N}*SizeofRouteEntry$).

### 3.2.4 Performance Simulations Based on the Concave Metric

Once we have obtained the equations for modelling the concave metric we compare the results to evaluate the performance difference between the generic and the FDVB Ad hoc network models. In order to simplify the equation, we consider an uniform distribution of nodes in all directions where $n$ is the average number of one hop neighbour of a node: $n_1 = n, n_2 = 2n, n_3 = 3n, n_4 = 4n, n_5 = 5n, n_6 = 6n$

Eq 70 becomes $f_R(n_\gamma, \gamma) = Q_R W_R[\rho + (\rho-1)3n + (\rho-2)7n + (\rho-3)11n]$

In order to evaluate the network performance in terms of the concave metric, Table 9 shows the values used for the variables in the equations.

**Table 9. Concave metric simulation values for the generic Ad hoc network model.**

| $B_N$ | $n$ | $B_T = \frac{B_N}{n}$ | $Q_R$ | $W_R$ |
|---|---|---|---|---|
| 11Mbs | 20 nodes | 11/20=0.55Mbs | 0.7 route request/s | 68bytes |

We will vary the percentage of ordinary and smart nodes in the network and their mobility to see the effect on the Ad hoc network performance.

The equation Eq 71 of the concave metric in the generic Ad hoc network model after replacing the proposed simulation values is the following:

$$B_R(i,j) = 0.55(Mb) - \left([Q_R 68\rho + Q_R 68(\rho-1)3n + Q_R 68(\rho-2)7n + Q_R 68(\rho-3)11n]e^{0.045M_{kl}}\right)_{k,l}$$
$$\forall k,l \in Path$$

The bandwidth in Ad hoc networks including the FDVB concept is modelled with Eq 72. In order to evaluate the network performance in terms of the concave metric, Table 10 shows the values used for the variables in the model.

OLSR defines a period of 2s (i.e. 0.5 route updates/s) between Hello messages and 5s (i.e. 0.2 route updates/s) between Topology messages. Considering that each node will have around 20 neighbours ($n_e$) and that the smart nodes will keep information only from those ordinary nodes that they received RREQ in the past, we assume that each smart node will maintain information from 80% of their neighbours ($\widehat{N}$=16).

The Hello messages in OLSR are similar to RREQ in AODV but the size of the Topology messages in OLSR depend on the number of neighbours for which the smart node keeps their link information ($60+4\hat{N}(bytes)$).

We vary the percentage of ordinary and smart modes in a range from 0% to 100%. Thus, since the total number of nodes within each hop is 20 we will have $N'=12$ for 2 hops routes and 30% of smart nodes.

**Table 10. Concave metric simulation values for the FDVB Ad hoc network model.**

| $Q_P$ | $\hat{N}$ | $W_P$ | $N'$ | $W_R$ |
|---|---|---|---|---|
| 0.4 route updates/s | 16 | $60+4\hat{N}(bytes)$ | 12 nodes | 68bytes |

The equation Eq 72 of the concave metric in the FDVB Ad hoc network model after replacing the proposed simulation values is the following:

$$B_R(i,j)=0.55(Mb)-\left(\left[Q_R 68\rho+Q_R 68(\rho-1)3n+Q_R 68(\rho-2)7n+Q_R 68(\rho-3)11n\right]e^{0.045M_{kl}}\right)_{k,l}$$
$$\forall k,l\in Path$$

$$B_F(i,j)=0.55(Mb)-\left(\left[\rho+(\rho-1)3n+(\rho-2)7n+(\rho-3)11n\right]Q_R 68(1-p_S(t))e^{0.045M_{kl}}\right)_{k,l}$$
$$-\left((12-1)0.4(60+4*16)p_S(t)e^{0.001M_{kl}}\right)_{k,l}\ \forall k,l\in Path$$

..

Figure 37 shows the available bandwidth in routes with 1, 2 and 3 hops with different percentage of smart and ordinary nodes in the network.

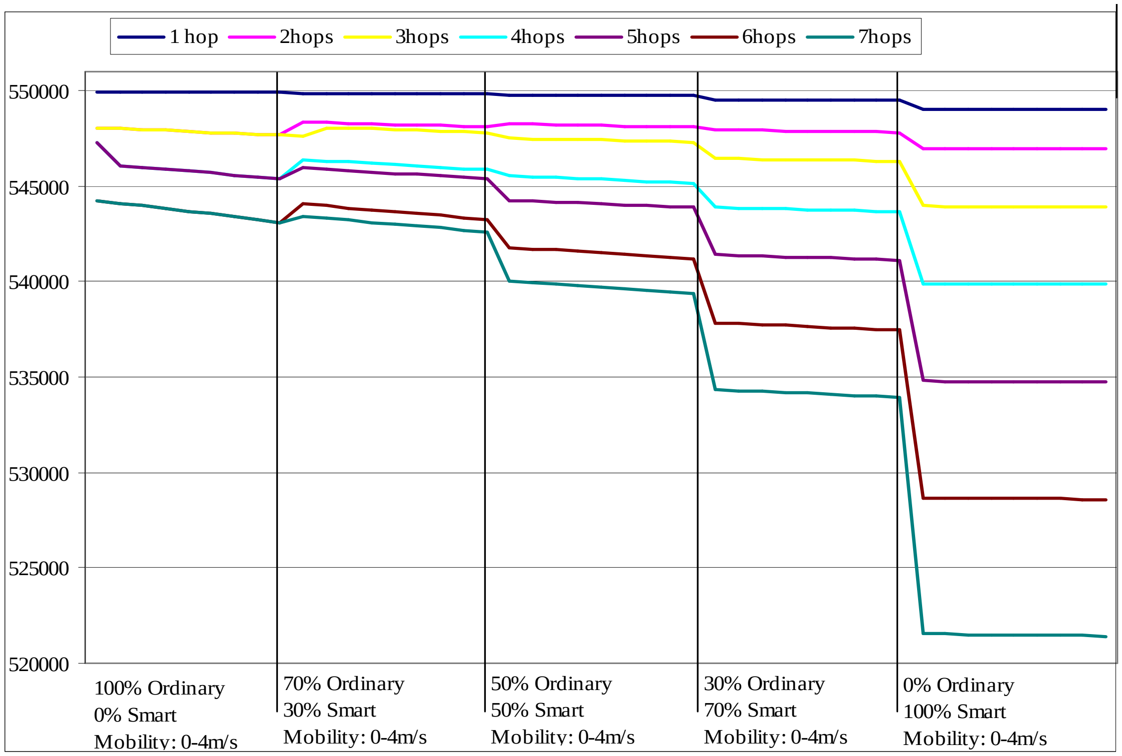


**Figure 37. Available bandwidth in routes with 1, 2 and 3 hops.**

Figure 37 shows that in routes with 1 hop size the percentage of ordinary or smart nodes does not have much impact on the available bandwidth. We can see that with ordinary nodes only (O=100% and S=0%) the overhead is the same for 2-3, 4-5 and 6-7 hops because the protocol uses the same TTL for the route request in those cases.

We can see that in all cases, except in 1 hop networks, introducing a low percentage of smart nodes (S=30% - 50%) increases the available bandwidth. However, when all the nodes in the route are smart (O=0% and S=100%) the bandwidth decreases. This effect has higher impact in large networks as we can see in Figure 37 where the bandwidth capacity is reduced 5% by the overhead when the destination is 7 hops from the originating node. We observe that introducing a low percentage of smart nodes (S=30%) gives the highest value of the concave metric when the size of the network increases (i.e. route with 4-5 or 6-7hops).

### 3.2.5 Additive Metric of the Ad hoc Network Model

The next step in the analysis is to define the model equation for the additive metric. The end to end packet delay *D(i, j)* is an additive metric because it is the sum of the packet delays on each link in the path from node *i* to node *j*. This metric depends on the number of hops in the path.

$$D(i,\ j) = a(i,\ j) = a(i,1) + a(1,2) + a(2,3) + \ldots + a(k,\ j) = a_{i,1} + a_{1,2} + a_{2,3} + a_{k,j}$$

This model is similar to the one used in fixed networks Eq 56. $f(a_{ij}) \equiv \min \sum_{(i,j)\in A} a_{ij} x_{ij}$ where *aij* is the average packet delay to cross the link (*i*,*j*).

However, in the Ad hoc network model we have to take mobility into account. The end to end packet delay in Ad hoc networks is higher than in fixed networks because there is an unstable network environment due to the nodes mobility and the topology information is constantly changing. For these reasons having optimized routes from the end to end packet delay standpoint, is difficult.

Therefore, when considering the FDVB Ad hoc network model we have analyzed the impact of the types of nodes in the network (i.e. ordinary and smart). We concluded that the end to end delay is not affected by the type of nodes since all of them will have similar processing capabilities. However, the end to end delay is affected by the number of hops in the route and the node mobility despite having smart nodes in the path. Thus, having smart nodes in the network will decrease the end to end packet delay because their mobility is lower and they find optimal routes to reach the destination with a minimum number of hops.

Therefore, we conclude that the routing optimization based on the minimum number of hops will provide the lowest end to end packet delay. However, an additional optimization based on the type of nodes in the path and their mobility should be considered. The routes with a higher number of nodes with low mobility might have lower end to end delay than routes with few nodes but high mobility and a higher number of hops.

### 3.2.7 Ad hoc Model Evaluation Conclusions

The simulation results for the multiplicative and additive metrics represented in Figure 36 show that adding smart nodes will improve the network performance in terms of connectivity and the end to end packet delay. However, increasing excessively the number of smart nodes will not be an optimal solution since smart nodes are severely affected by mobility that decreases the probability of connectivity and the network lifetime as shown in Figure 36. Moreover, the results for the concave metric depicted in Figure 37 show that a reasonable percentage of smart nodes (i.e. 30%) provides better performance than having either ordinary or smart nodes only in the network. In terms of the probability of connectivity the optimal value results when all the nodes are smart with mobility zero, which is equivalent to the fixed networks environment. However, after considering the rest of metrics we have seen that having a certain percentage of smart nodes joining the network will reduce the end to end packet delay and increase the available bandwidth keeping the connectivity at a certain level.

We also have to consider that the number of smart nodes joining the network decreases over time (i.e. Poisson arrival time), thus a control mechanism is necessary to keep the percentage of smart nodes in the network around 30%.

From the mathematical Ad hoc network models we now conclude that we obtained results that provide a first estimation of the optimal parameters to improve the network scalability.

## 3.3 Fully Distributed Virtual Backbone Creation Algorithm

SARP is based on the smart nodes that get incentives as reward for their contribution to the extra routing functionality in order to increase the network scalability. The ordinary nodes that do not implement SARP can still be part of the network and indirectly benefit from the SARP protocol. From the mathematical analysis, we concluded that when the number of smart nodes excessively increases, the optimal network performance is lost. Therefore, SARP has to implement a *smart nodes access control algorithm* to limit the number of nodes becoming smart and contributing to the Fully Distributed Virtual Backbone (FDVB).

When joining the Ad hoc network the nodes decide, following a policy, whether they become smart nodes and take part in the FDVB or they just remain as ordinary nodes, which is the default state. The FDVB creation policies are the following.

a) Fixed policy, where the nodes have a predefined logic assigned by an administration entity (e.g. professional radio networks) forcing them to become smart nodes.
b) Dynamic policy, where the nodes apply a dynamic heuristic algorithm based on their available resources to decide whether they become smart or remain as ordinary nodes.
c) Mixed policy, where both fixed and dynamic policies apply. There are some nodes that are forced to become smart nodes by an administration entity while other nodes voluntarily join the FDVB. As an example, in an emergency situation the rescue team implements the fixed FDVB creation policy and they are forced to become smart nodes to set up the Ad hoc network. Other nodes will apply the dynamic FDVB creation policy deciding by themselves to join the FDVB.

Figure 38 represents the steps to implement the FDVB creation algorithm following the dynamic policy. It includes the smart nodes access control algorithm based on heuristics.

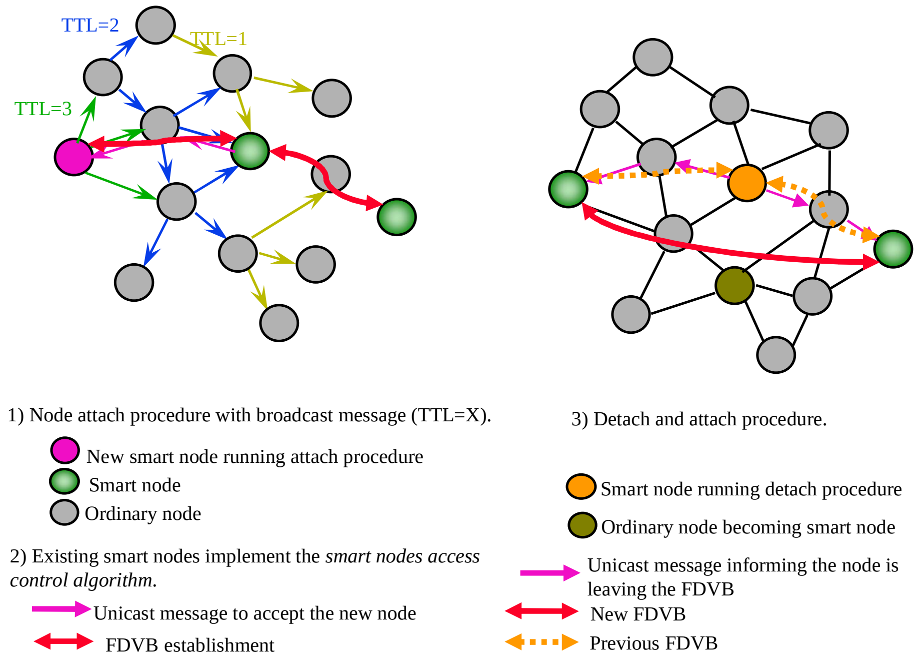


**Figure 38. Fully distributed virtual backbone creation algorithm.**

1) A node interested in joining the FDVB issues a broadcast message with TTL=X to discover other smart nodes present in the network and performs a node attach procedure as depicted in Figure 38 step 1). TTL=1 does not add any value because it restricts the discovery process to smart nodes located within a single hop. Thus either TTL=2 or 3 is required to find other smart nodes that are 2 or 3 hops away. Based on the test bed results, 2 or 3 hops set the limit of the path length between nodes for having good real time communications. The smart nodes communicate between them in the attach procedure so they are aware of the smart nodes available in the proximity (i.e. TTL=3). This approach does not require a complete knowledge of the network nor the total number of smart nodes existing in the network. It reduces the flooding but the drawback is that might lead to the creation of various disjoint virtual backbones (i.e. FDVB) in very large networks. Nevertheless, the overall network performance increases despite having separated FDVBs since the

routing is optimised when a route request reaches an area with a FDVB where the smart nodes maintain the topology information.

2) The smart nodes in the Ad hoc network (if any) will apply the smart nodes access control algorithm according to heuristics based on the current number of smart nodes in the network. If the number of smart nodes exceeds a limit, the new node will not join as a smart node. Otherwise, the smart nodes in the FDVB will send a unicast message to the new node to join the FDVB at the expense of its own resources as depicted in Figure 38 step 2). The threshold for accepting new smart nodes is set at the point when 30% of the nodes in the network are smart.

3) A smart node may become an ordinary node at any time e.g. when its resources are exhausted or it leaves the network. The node becoming ordinary performs the detach procedure by sending a unicast message to inform the rest of smart nodes that it is leaving the FDVB. The rest of the nodes notice that the number of smart nodes in the FDVB has decreased and they accept a new node trying to join the FDVB, establishing a new FDVB through a different path as depicted in Figure 38 step 3).

Figure 39 shows the SARP state machine that each smart node has run to implement the FDVB creation algorithm.

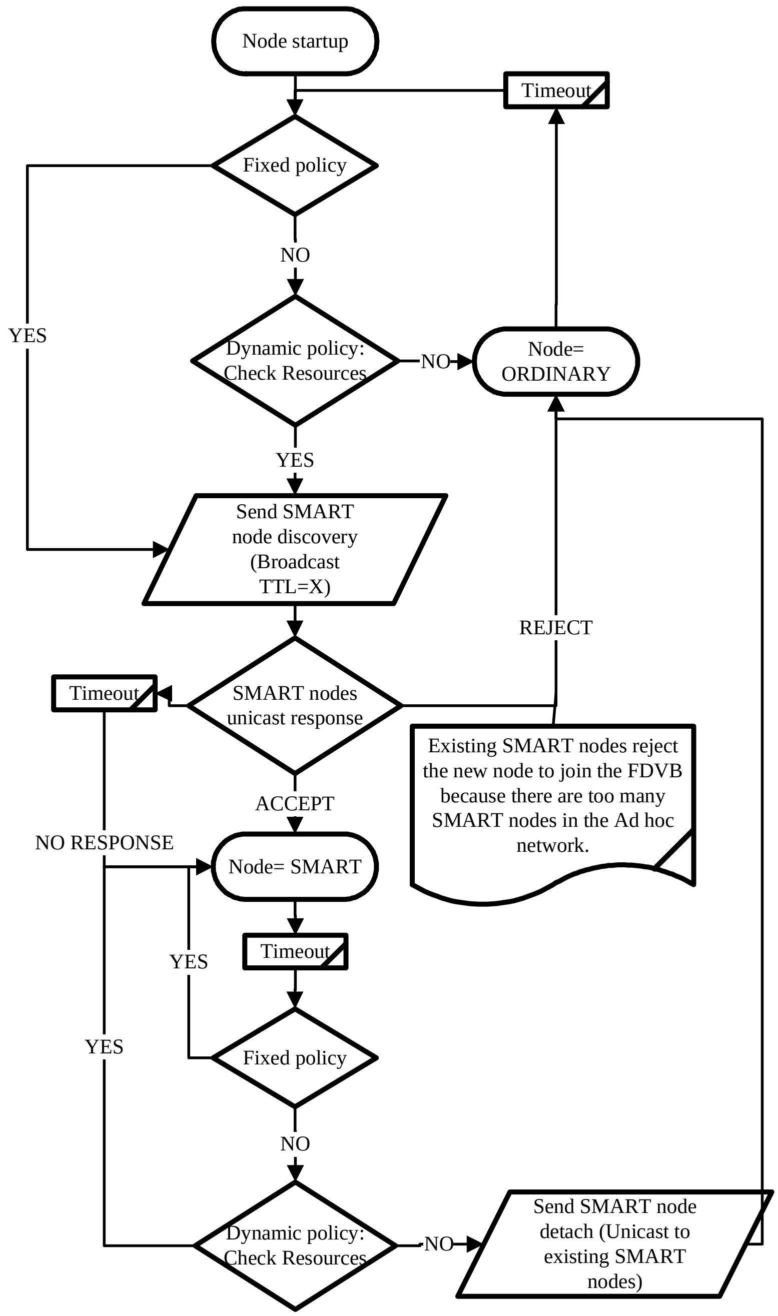


**Figure 39. SARP state machine for joining the FVDB.**

### 3.4 SARP Simulation Results

This section presents the SARP simulation results compared with the AODV and OLSR results. The SARP protocol is implemented in the simulator (i.e. ns-2) by integrating AODV and OLSR together. In order to simulate SARP using the existing reactive and proactive protocols, the routing information obtained from AODV has to be copied into the routing tables of OLSR and viceversa. The resulting protocol will behave like SARP where the smart nodes maintain the link state information while the ordinary nodes execute standard AODV. The smart nodes implemeting SARP execute OLSR to exchange route updates between them and AODV to receive route requests from ordinary nodes. The smart nodes deliver route responses based on the information obtained from OLSR. We consider standard IEEE 802.11 MAC protocol for both the simulations and test bed. In following chapters we analyse the need of a cross-layer architecutre with certain enhancements in the MAC protocol.

The results are obtained from the average of three simulations rounds considering the following parameters:

- Simulation area: 1500m x 300m.
- Simulation time: 900 seconds.
- Constant Bit Rate (CBR) traffic flows with UDP transport: 20 IP unidirectional flows.
- Connection rate: 8 packets/second.
- Packet size: 65 bytes.
- Number of nodes: 50 nodes using random waypoint mobility pattern.
- Pause time between node movements: 0, 30, 60, 120, 300, 600 and 900 seconds.
- Distribution of smart and ordinary nodes:
    1. SARP-5 = 5 smart nodes and 45 ordinary nodes.
    2. SARP-15 = 15 smart nodes and 35 ordinary nodes.
    3. SARP-30=30 smart and 20 ordinary nodes.

Figure 40 shows the routing overhead generated by AODV, OLSR and SARP versus node mobility. The routing overhead generated by SARP-5 and AODV are similar. The SARP routing overhead increases when the number of nodes increases. However, OLSR generates the highest routing overhead regardless the node mobility.

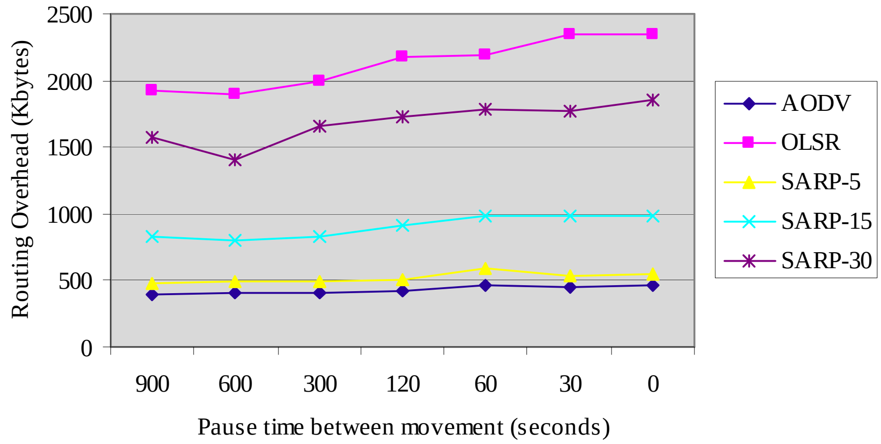


**Figure 40. AODV, OLSR and SARP routing overhead.**

Figure 41 shows the end to end packet delay versus the node mobility. In high mobility conditions SARP and AODV behave similarly but when the number of smart nodes increases the performance is slightly affected by the mobility. Nevertheless, SARP-30 still introduces lower delay than OLSR.

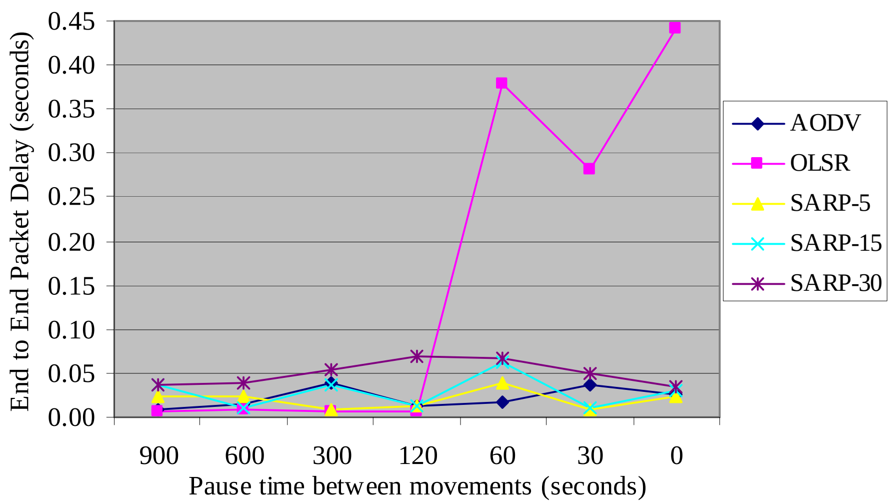


**Figure 41. AODV, OLSR and SARP end to end packet delay.**

Figure 42 shows that OLSR always provides the highest percentage of optimal routes despite incrementing the number of smart nodes using SARP.

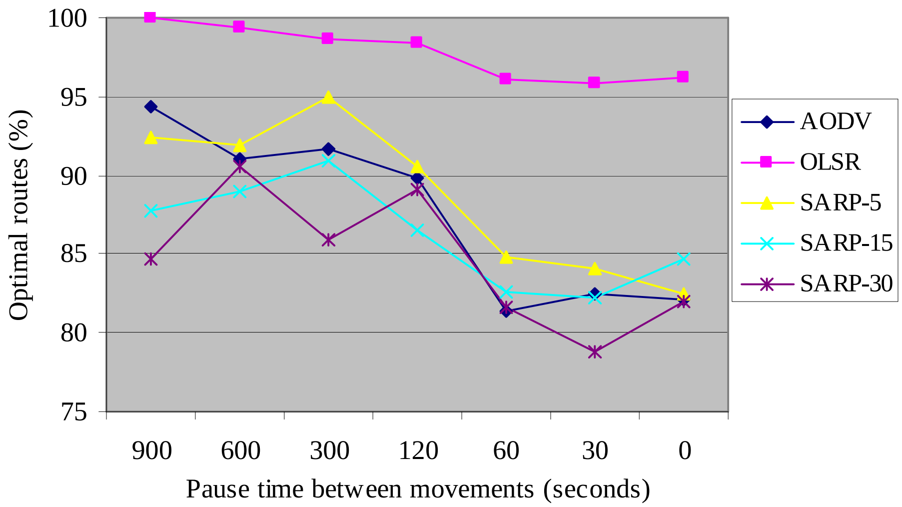


**Figure 42. AODV, OLSR and SARP percentage of optimal routes.**

In this section, we can conclude from the simulation results that SARP achieves similar performance in terms of data throughput and delay than reactive routing protocols. The side effect with SARP is an increase in the routing overhead but still far from the results obtained with proactive routing protocols. Therefore, despite of the inaccuracy of the simulations, a hybrid approach such as the SARP routing protocol improves Ad hoc networks scalability.

### 3.5 SARP Implementation Advantages and Drawbacks

The advantages of the proposed SARP implementation are backward compatibility with existing routing protocols and the minimum set of requirements in the nodes. The drawback is that the routes obtained from OLSR might not be valid due to the longer periods of topology updates in OLSR. The probability of stalled routes provided by OLSR is low since the basic criteria for becoming a smart node is the resources availability and low mobility. Another inconvenience of implementing SARP using reactive and proactive protocols is that the routing cache expiration timeouts are different. OLSR keeps on introducing the routes into the AODV routing cache since AODV deletes the routes that are not in use. This justifies our proposal of implementing a single protocol that behaves as reactive and takes a proactive behaviour when the node becomes smart. Therefore, in Chapter 5 we propose the implementation of SARP using AODV extended with clustering features.

Chapter 4 analyses the feasibility of implementing the SARP protocol in real networks in terms of node incentives.

# Chapter 4

## Incentives for Participating in Hybrid Routing

In real Ad hoc networks the users do not want to exhaust their resources for the benefit of other people. However, the hybrid routing approach requires that some nodes implement extra routing functionality. The basic packet forwarding required in all the nodes to support the Ad hoc networking consumes additional energy. Participation in the hybrid routing approach requires some incentives for the nodes to contribute to the extra routing functionality. The nodes can be forced to participate based on a fixed policy if they belong to a certain administration. However, if the nodes are not under a single administration policy their participation to the hybrid routing approach depends on their own willingness following a dynamic policy.

This Chapter considers only the dynamic policy to join the Ad hoc network, and studies the possibility to offer additional incentives to the nodes in order to encourage their participation to the routing functionality. We use game theory [3] to analyse the additional incentives for the Ad hoc nodes to contribute to the SARP protocol and be part of the FDVB [22]. SARP is a fully distributed cluster-based routing protocol where nodes become cluster heads simply based on available resources. The nodes are not forced to become cluster heads. However, the smart nodes have to gain some benefit from their cooperative behaviour as cluster heads. In this Chapter we propose a rewarding mechanism that ensures the payoff for the extra functionality performed by the cooperative nodes. For this reason, smart nodes have to implement a cross-layer architecture to enforce priority queues for packet forwarding. The incentives could be based on the QoS granted to the nodes that contribute to the routing and forwarding functions.

Game theory has been mainly used in economics to model business competition but recently it has been applied in other areas of science and engineering. Wireless networks in general and Ad hoc networks in particular can be modelled as a game where the nodes decide to transmit or not over a limited resource available such as the radio spectrum to the expense of their battery.

**4.1 Game Theory Introduction**

A game consists of an interaction between two or more players where each of them can make different moves or actions that result into a specific outcome for each player (i.e. payoff) depending on the moves. The moves taken by the players at any point in time are determined by the strategy followed during the game. Each player has its own strategy. The strategy is the guideline followed by the players to select the preferred move based on available information of the expected outcomes, the moves from the other players, previous moves, etc. The game is either *simultaneous*, if the players make their moves simultaneously and they are not aware of previous moves from other players, or *sequential* if the players have information about previous moves of other players. Chess is a good example of a sequential game where the players have exact information of previous moves from the other player.

A game could last infinitely depending on the possible combinations of moves. However, games are generally terminated in a finite number of moves where each player in a rational way tries to maximise the payoff of the game. The payoff is the outcome for the player in the game. In wireless networks maximising the payoff during each move would be equivalent to minimising the cost of the transmission, routing, etc. A game can be classified either as a *zero-sum* game if the game payoff (i.e. the sum of each of the player's payoff) always adds up to zero for any possible combination of strategies, or as a *nonzero-sum* game if the game payoff can be different from zero. Poker is a zero-sum game since the money that some players loose is collected by others and the game payoff is zero despite that the winner has a higher individual payoff than the other players.

A game can be also classified as *cooperative* if the players agree on some moves or *non-cooperative* when the players make the moves on their own without any previous agreement with other players.

In order to analyse a game we introduce Eq 73 as the normal expression used to represent a game [3].

**Eq 73.** $G = (P, S, U)$

$P$ is the set of players which most games consider only two $p_1, p_2 \in P$ despite that the game can be extended to any number of players without losing generality. We consider $p_i$ the player under analysis and $p_j$ the opponent. $S$ represents the strategies of the game where $S_i$ is the strategy of player $p_i$ and $S_j$ is the strategy of the opponent $p_j$. The players can select a specific strategy during the game set by the strategy profile $s = \{s_1, s_2\}$. $U$ represents the payoff of the game where $u_i(s)$ is the payoff of the player $p_i$ for the strategy profile $s$ taken in the game.

The normal form to represent a game uses a matrix to visualize the different strategies and the payoff for the move of each player acting simultaneously. The different strategies for each player define the number of rows or columns. Table 11 shows the matrix of the "Prisoner's Dilemma" game that is characterized by the scenario where the police is interrogating two thieves in separate prisons [3]. The police offers them separately to go free if they confess the crime implicating their partner. The prisoners have two options: to confess or not confess. If one prisoner confesses but his partner does not, the cooperative prisoner will go free and get all the money (i.e. payoff=1) while his partner will go to jail and loose his part of the money (i.e. payoff=-1). If both prisoners cooperate with the police and confess, they will go to jail and loose all the money (i.e. payoff= -0.5). If neither of them confesses, they will be free and will have to share the stolen money (i.e. equal payoff=0.5).

**Table 11. Matrix representation of the prisoner's dilemma game.**

| | | Thief B | |
|---|---|---|---|
| | | Confess | No Confess |
| Thief A | Confess | -0.5,-0.5 | 1, -1 |
| | No Confess | -1, 1 | 0.5,0.5 |

In case of sequential games where the players do not make their moves simultaneously, they are represented with a tree structure where each vertex is a point of choice where each player has to make the move and each branch represents a different strategy [3].

Once the game is identified, the next step is to solve it by predicting the strategy that each player will take. The game can be solved if there is a strict dominance, meaning a player who decides the moves based on a dominant strategy. According to the payoff matrix the dominant strategy provides the best results regardless of the opponents' moves. Eq 74 indicates that any strategy $s_i$ is dominated by the dominant strategy $s_i^d$ since the payoff will always be higher.

**Eq 74.** $u_i(s_i, s_j) < u_i(s_i^d, s_j)$

In some games there is no clear dominant strategy and instead the game can be solved considering a weak dominance. Eq 75 represents the inequality of the payoff when the player $p_i$ selects the weakly dominant strategy $s_i^{wd}$ versus any other strategy $s_i$.

**Eq 75.** $u_i(s_i, s_j) \leq u_i(s_i^{wd}, s_j)$

In most of the cases the games cannot be solved using dominance techniques because it is difficult to always find a dominant or a weakly dominant strategy. Therefore, some games need to be solved using the concept of Nash equilibrium [60] represented in Eq 76.

**Eq 76.** $u_i(\hat{s}_i, \hat{s}_j) \geq u_i(s_i, \hat{s}_j)$

Eq 76 indicates that the pure strategy profile $\hat{s}$ constitutes a Nash equilibrium if none of the users can unilaterally increase their payoff by changing their strategy.

**4.2 Formulation of the Ad hoc Routing Game**

In the remaining of this Chapter we use game theory to analyse a wireless Ad hoc network [61], [62] considering it as a game where the players are the wireless Ad hoc nodes. An Ad hoc network is a non-cooperative game since there are no previous agreements between the nodes. In Ad hoc networks the strategy of the game consists of either to participate in packet forwarding and basic routing function required to create the network, or not to participate but still benefit from the Ad hoc networking for the communications. Therefore, we consider that Ad hoc networks are fully distributed and can be modelled as a simultaneous game where the nodes are not aware of the strategy of other nodes. The nodes will try to maximize their payoff function by reducing their participation to minimise the associated cost but benefiting from the network. Following, we represent the Ad hoc network game according to Eq 73.

$G_{Ah} = (P, S, U)$

$P$ is the set of Ad hoc nodes. $S$ represents the strategies of the game, $s_1 = D$ to drop the packet and $s_2 = F$ to forward the packet. $U$ represents the payoff of the game and $u_i(s)$ is the node payoff.

According to Metcalfe’s law the value of the network is equivalent to the number of nodes one can communicate with. Thus, the value to one node is $u \approx n$ and the value of the whole network is $u \approx n^2$. Thus, each node have an inherent payoff, which is equivalent to the number of nodes in the network $u \approx n$ that they can communicate with minus the cost of the packet forwarding, $c_f$.

In Table 12 we can see that if node *j* forwards the packet but node *i* drops it, node *j* will have a negative payoff equivalent to the cost of forwarding the packet but node *i* will have the maximum payoff because it is saving energy and is able to communicate with the help of node *j*. If both nodes forward the packet they will get the payoff equivalent to the benefit of establishing the communication minus the cost associated with packet forwarding. If both nodes drop the packet, their payoff will be null because the rewarding is zero since they cannot communicate and the cost associated with the packet forwarding will be also zero.

**Table 12. Matrix representation of the basic Ad hoc network game.**

| | | Node j | |
|---|---|---|---|
| | | F | D |
| Node i | F | ($1-c_f$, $1- c_f$) | ($-c_f$,1) |
| | D | (1, $-c_f$) | (0,0) |

This game can be easily solved using the dominant strategy. From Table 12 we can see that the strategy $s_1 = D$ is the dominant because it provides the best option regardless the strategy of the opponent node *j*. Therefore, the result of the game is $s = s_i = s_j = D$ with $u_i(s) = u_j(s) = 0$.

We conclude that in the Ad hoc network game, as a non-cooperative game, each node tries to optimise its own payoff capturing the available bandwidth in the network. Unfortunately, this behaviour does not lead to the best network performance.

### 4.3 SARP Ad hoc Game Formulation

In this section we extend the previous analysis of the Ad hoc network game considering the hybrid Ad hoc routing approach (i.e. SARP). When using game theory for analyzing SARP we define a game based on the Ad hoc network game with a new additional strategy.

The performance results showed that if all the nodes only contribute with the basic packet forwarding and basic routing functions, the network will not scale properly. Therefore, there has to be nodes that in addition to the packet forwarding and basic routing functions implement the SARP protocol, which is the new strategy in the game.

In the SARP Ad hoc game analysis we will use the following notation and the distribution of nodes as represented in Figure 43.

$N$ number of nodes in the Ad hoc network.

$k$ number of cooperative nodes = number of (ordinary + smart) nodes.

$N-k$ number of non-cooperative or free rider nodes.

$l$ number of smart nodes is a subset of the number of cooperative nodes (i.e. $l=\eta k$ where $0 \le \eta \le 1$)

$k-l$ number of ordinary nodes which is equivalent to $k(1-\eta)$.

$u_N$ network payoff for $N$ number of nodes.

$u^O$ individual payoff for an ordinary node.

$u^S$ individual payoff for a smart node.

$u^{nc}$ individual payoff for a non-cooperative node.

$u_N^O$ network payoff for all the ordinary nodes.

$u_N^S$ network payoff for all the smart nodes.

$u_N^{nc}$ network payoff for all the non-cooperative nodes.

$c_f$ cost of packet forwarding and basic routing functions.

$c_S$ cost of SARP routing functionality.

$c$ cost of smart nodes routing functionality ($c = c_S + c_f$).

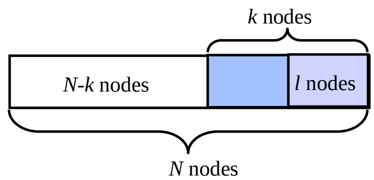


**Figure 43. Distribution of smart, ordinary and non-cooperative nodes.**

We analyse an Ad hoc network were there are $k$ nodes in the network that contribute to the basic routing functionality (i.e. cooperative nodes) and the rest of nodes (i.e. N-$k$ ) in the network are free rider (i.e. non-cooperative) nodes that will benefit from them. The non-cooperative nodes do not contribute to the packet forwarding and basic routing functionality while the ordinary nodes do. We consider that within the cooperative nodes there are $l$ nodes (i.e. smart) that are part of the FDVB (i.e. implement SARP) and increase the network payoff by contributing to the scalability. The rest of cooperative nodes (i.e. $k$-$l$ ) are ordinary nodes that are not part of the FDVB but contribute to the packet forwarding and basic routing functions.

The basic assumptions in the SARP Ad hoc game are the following:

- Improving scalability in Ad hoc networks requires that certain nodes take an active role and help others in the benefit of the Ad hoc network. These active nodes (i.e. smart nodes) will spend some extra resources in performing the SARP routing functionality besides the basic routing functions. Therefore, they will exhaust their resources more rapidly than the ordinary nodes.
- A smart node participating in the FDVB may give up at some point in time because its resources are exhausted. In this situation, the node should not be considered as a non-cooperative node but just a node that punctually was not able to continue the extra effort for helping others and it returned to its default mode (i.e. ordinary node).

These assumptions change the game because the nodes have the SARP routing $s_3 = F_S$ as an additional strategy besides the packet forwarding and basic routing functions. Moreover, this new strategy will have an additional cost meaning the cost from the packet forwarding and basic routing functions (i.e. $c_f$) plus the extra cost due to the SARP routing functionality (i.e. $c = c_f + c_S$).

Following, we represent the SARP Ad hoc game using Eq 73.

$G_{Ah}(SARP) = (P, S, U)$

$P$ is the set of the Ad hoc nodes.

$S$ represents the strategies of the game, $s_1 = D$ to drop the packet, $s_2 = F$ to forward the packet and implement the basic routing functions, and $s_3 = F_S$ to forward the packet, implement the basic routing functions and contribute to the SARP routing functionality.

In real implementations the dropping strategy is implemented by switching the node or routing functions off (i.e. equivalent to leaving the Ad hoc network) in order to save battery when the user does not want to communicate. Packet forwarding with the basic routing functions is implemented by switching the node or the routing functions on (i.e. equivalent to joining the Ad hoc network again) when the user wants to communicate. The SARP routing will be implemented when the node decides to engage into SARP routing functions if the node has enough resources. $U$ represents the payoff of the game and $u_i(s)$ is the node $i$ payoff.

According to Metcalfe's law the value of the network per node is equivalent to the number of nodes. Thus, the sum of the node payoffs in the network is the network value, which is equivalent to the square of the number of nodes in the network $u_N \approx N^2$ that the nodes can communicate with minus the cost of packet forwarding, basic routing functions and the cost of contributing to the SARP routing functionality.

The smart nodes participate implementing the SARP protocol creating a link between them to help others. Thus, if these nodes participate and contribute to the network scalability, they increase the inherent network payoff $u_N \approx N^2$ due to the higher number of nodes available in the network ($u_N \approx N^2 = \max = 1$). The rest of the nodes in the network will act either dropping the packets (i.e. non-cooperative or free rider node) making the overall network payoff null ($u_N \approx N^2 = 0$) or participating in the packet forwarding and basic routing functions as ordinary nodes with less payoff depending on the number of nodes reachable ($u_N \approx N^2 = medium = 0.5$). If the nodes participate in the packet forwarding and basic routing functions but not in the SARP routing functionality, they will benefit from the value of the network but their payoff will be low because the number of nodes reachable in the network will decrease due to scalability and performance limitations in the network.

The smart nodes payoff is lower (i.e. $u^S = 1 - [c_f + c_S]$) than the non-cooperative nodes payoff since they simply drop the packets at no cost (i.e. $u^{nc} = 1 - [0]$)). The smart nodes payoff is also lower than the ordinary nodes payoff because they only participate in the packet forwarding and basic routing functions with a lower cost (i.e. $u^O = 1 - [c_f]$). The non-cooperative nodes benefit from the ordinary and smart nodes, while the ordinary nodes benefit from the smart nodes that contribute to the network scalability. The result is that $u^S < u^O < u^{nc}$.

Similarly to the Ad hoc game, we can apply the idea of trying to identify a dominant strategy for the SARP Ad hoc game. The values of $c_f$ and $c_S$ determine the dominant strategy in Table 13. We assume that $c_f$ is much lower than the benefit of participating in the network $c_f << 0.5$. On the other hand $c_S$ can be either reasonably small $c_S < 0.5$ making the sum $c_f + c_S$ considerably lower than the benefit of participating in the network $c_f + c_S < 0.5$ or reasonable big making the sum $c_f + c_S$ higher than the benefit of participating in the network $c_f + c_S > 0.5$.

**Table 13. Matrix representation of the SARP Ad hoc game.**

| | | Node j | | |
|---|---|---|---|---|
| | | F | $F_S$ | D |
| Node i | F | (0.5-$c_f$, 0.5-$c_f$ ) | (1-$c_f$, 1-[$c_f$+$c_S$]) | (-$c_f$, 0.5) |
| | $F_S$ | (1-[$c_f$+$c_S$], 1-$c_f$) | (1-[$c_f$ +$c_S$], 1-[$c_f$+$c_S$]) | (-[$c_f$+$c_S$], 1) |
| | D | (0.5, -$c_f$) | (1, -[$c_f$ +$c_S$]) | (0, 0) |

Using Table 14, we assign some numeric values to $c_f$ and $c_S$ to see the difference in the matrix representation of the game in each case. Table 14 shows the values obtained with $c_f \approx 0.1$ and $c_S \approx 0.2$ so $c_f + c_S < 0.5$.

**Table 14.** **Matrix representation of the SARP Ad hoc game with $c_f$+$c_s$<0.5.**

| | | Node j | | |
|---|---|---|---|---|
| | | F | $F_S$ | D |
| Node i | F | (0.4, 0.4) | (0.9, 0.7) | (-0.1 0.5) |
| | $F_S$ | (0.7, 0.9) | (0.7, 0.7) | (-0.3., 1) |
| | D | (0.5, -0.1) | (1, -0.3) | (0, 0) |

Table 15 shows the values obtained with $c_f \approx 0.1$ and $c_S \approx 0.5$ so $c_f + c_S > 0.5$.

**Table 15.** **Matrix representation of the SARP Ad hoc game with $c_f$+$c_s$>0.5.**

| | | Node j | | |
|---|---|---|---|---|
| | | F | $F_S$ | D |
| Node i | F | (0.4, 0.4) | (0.9 0.4) | (-0.1 0.5) |
| | $F_S$ | (0.4, 0.9) | (0.4, 0.4) | (-0.6, 1) |
| | D | (0.5, -0.1) | (1, -0.6) | (0, 0) |

We can see that there is no dominant strategy despite it looks $s_1 = D$ provides the best option regardless the strategy of the other node. This is the case for $c_f + c_S > 0.5$ where $s_1 = D$ seems to be the dominant strategy and the result of the game is the same as in the Ad hoc game $s_i = s_j = D$ with $u_i(s) = u_j(s) = 0$. However, when $c_f + c_S < 0.5$ we have that $u_i(F_S, F) > u_i(D, F)$ meaning that if the node *j* plays strategy $s_2 = F$ the node *i* has higher payoff by playing strategy $s_3 = F_S$ instead of playing strategy $s_1 = D$. However, if the node *i* plays this strategy then the node *j* obtains higher payoff than the node *i*. The node *j* payoff increases, thus it is not Nash Equilibrium but the benefit is that the overall network payoff increases.

The result is that the value of $c_S$ can determine the node strategy leading to a combination of scalability increase versus $c_S$ cost. Thus, when $c_S$ is low if nodes play $s_2 = F$ then playing $s_3 = F_S$ is a reasonable strategy that will increase the node payoff. Therefore, under these conditions an optimum network performance can be obtained if the nodes play $(F_S, F)$ instead of $(D, D)$.

In our analysis we have considered a game that consists of an Ad hoc network where the nodes either implement SARP becoming smart, they remain ordinary nodes or they do not participate in the packet forwarding. In non-cooperative games the nodes always try to maximise their payoff. In order to show the feasibility of this network, we have to find the motivation for the nodes to become smart. In Table 13 we see that payoff leads to equilibrium where all the nodes tend to drop the packets and the network payoff is null. Therefore, we consider an alternative strategy where the motivation of nodes can be modelled with Equity Reciprocity and Competition (ERC) [63] preferences in a game with a non-negative payoff for each node. The ERC does not differ too much from standard games with monetary payoff. In this case ERC proposes an additional payoff known as relative share, which is a measure of how a player's monetary payoff compares to the one obtained by the other players of the game.

ERC proposes that players will get motivated not only by their own standard monetary payoff named $m_i$ but also if that payoff is big compared to the one that other players will get, named relative share $n_i$.

The relative share is defined as $n_i = \frac{m_i}{\sum m_j}$ where *i* is the node under analysis and *j* are all the players.

The payoff function in Eq 77 considers both the standard monetary payoff $m_i$ and the relative share $n_i$.

**Eq 77.** $u = \alpha_i m_i + \beta_i n_i$

$\alpha_i, \beta_i \geq 0$ are numeric values measuring the weight of $n_i$ and $m_i$ contribution to the payoff function.

In order to analyse the motivation of the nodes to become smart, we define the individual and network payoff for non-cooperative, ordinary and smart nodes. As individual payoff we refer to the standard monetary payoff as defined in ERC. The relative share defined in ERC is a measure of how the individual payoff compares to the network payoff.

The individual payoff for non-cooperative nodes is defined in Eq 78.

**Eq 78.** $u^{nc} = B(k) + B_S(l)$

where $B(k)$ is the benefit taken from the ordinary nodes and $B_S(l)$ is the benefit taken from the smart nodes.

The ordinary nodes will have the cost from the packet forwarding and basic routing functions represented in Eq 79.

**Eq 79.** $c_f = C(k)$

The smart nodes are ordinary nodes with SARP functionality so they will have the cost of packet forwarding and basic routing functions plus the additional cost due to the SARP routing functionality as represented in Eq 80.

**Eq 80.** $c = C(k) + C_S(l) = c_f + c_S$

The individual payoff for the smart nodes will be equivalent to the benefit obtained from the rest of smart nodes and the benefit from the ordinary nodes minus the cost associated with the smart nodes as indicated in Eq 81.

**Eq 81.** $u^S = B(k) + B_S(l) - [C(k) + C_S(l)]$

The individual payoff for the ordinary nodes will be the benefit of the smart and ordinary nodes minus the cost from the packet forwarding and basic routing functions as defined in Eq 82.

**Eq 82.** $u^O = B(k) + B_S(l) - C(k)$

As indicated in Eq 83 we assume that the individual payoff for a new node to participate in the game as ordinary node $u^O(k+1) = B(k+1) + B_S(l) - C(k+1)$ is lower than the individual payoff as non-cooperative node $u^{nc}(N+1) = B(k) + B_S(l)$.

**Eq 83.** $B(k+1) - C(k+1) < B(k) \Rightarrow B(k+1) - B(k) < C(k+1)$

We assume that the individual payoff for a new node to participate in the game as smart node $u^S(l+1) = B(k+1) + B_S(l+1) - C(k+1) - C_S(l+1)$ is lower than the individual payoff as ordinary node $u^O(k+1) = B(k+1) + B_S(l) - C(k+1)$ as indicated in Eq 84:

**Eq 84.** $B_S(l+1) - C_S(l+1) < B_S(l) \Rightarrow B_S(l+1) - B_S(l) < C_S(l+1)$

In the case of Ad hoc networks we also have to consider the network as part of the game and the incentives for the nodes to become part of the network. To analyze the network incentive, we consider the network payoff for all the nodes as represented in Eq 85.

**Eq 85.** 
$$\begin{aligned} u_N &= (N-k)u^{nc} + (k-l)u^{O} + lu^{S} = \\ &= (N-k)[B(k)+B_S(l)] + (k-l)[B(k)+B_S(l)-C(k)] + l[B(k)+B_S(l)-C(k)-C_S(l)] = \\ &= N[B(k)+B_S(l)] - kC(k) - lC_S(l) \end{aligned}$$

The network payoff if the node joins as ordinary is:

$$u^{O}_{N+1}(N+1,k+1,l) = (N+1)[B(k+1)+B_S(l)] - (k+1)C(k+1) - lC_S(l)$$

The network payoff if the node joins as non-cooperative is:

$$u^{nc}_{N+1}(N+1,k,l) = (N+1)[B(k)+B_S(l)] - kC(k) - lC_S(l)$$

We compare the network payoff of the ordinary and smart nodes versus the network payoff of the non-cooperative nodes in the network. A new node might have an incentive to join the network as ordinary if the network payoff is bigger than becoming just non-cooperative node as indicated in Eq 86.

**Eq 86.** $u^{O}_{N+1}(N+1,k+1,l) > u^{nc}_{N+1}(N+1,k,l)$

$$\begin{aligned} &u^{O}_{N+1}(N+1,k+1,l) > u^{nc}_{N+1}(N+1,k,l) \Rightarrow \\ &(N+1)[B(k+1)+B_S(l)] - (k+1)C(k+1) - lC_S(l) > (N+1)[B(k)+B_S(l)] - kC(k) - lC_S(l) \Rightarrow \\ &(N+1)B(k+1) - (k+1)C(k+1) > (N+1)B(k) - kC(k) \end{aligned}$$

Applying the assumption Eq 83 we can simplify the previous equation into:

$$NB(k+1) - kC(k+1) > NB(k) - kC(k)$$

We reformulate this equation to place the benefits in one side versus the costs on the other side as follows:

$$N[B(k+1) - B(k)] > k[C(k+1) - C(k)]$$

Claim 1: *Provided that N times the payoff of joining as ordinary is bigger than k times the cost of joining as an ordinary node, it will make sense for a new node to join as ordinary from overall network payoff point of view.*

The result reflects the fact that costs are incurred only by *k* cooperative nodes while even non-cooperative nodes increase the network value (i.e. Metcalfe's law) by being reachable although they do not contribute to the communication between the cooperative nodes. We can also see that every new node creates cost in terms of overhead meaning there is a limit where it does not make sense for the network to accept new nodes.

The node may decide to become smart if the network payoff as smart node is bigger than becoming just ordinary node as indicated in Eq 87.

**Eq 87.** $u^{S}_{N+1}(N+1, k+1, l+1) > u^{O}_{N+1}(N+1, k+1, l)$

The network payoff if the node joins as smart is:

$$u^{S}_{N+1}(N+1, k+1, l+1) = (N+1)\left[B(k+1) + B_S(l+1)\right] - (k+1)C(k+1) - (l+1)C_S(l+1)$$

Let us now assume that the network payoff of the node joining as smart is bigger than joining as ordinary node.

$$u^{S}_{N+1}(N+1, k+1, l+1) > u^{O}_{N+1}(N+1, k+1, l) \Rightarrow$$
$$(N+1)\left[B(k+1) + B_S(l+1)\right] - (k+1)C(k+1) - (l+1)C_S(l+1) > (N+1)\left[B(k+1) + B_S(l)\right] - (k+1)C(k+1) - lC_S(l) \Rightarrow$$
$$(N+1)B_S(l+1) - (l+1)C_S(l+1) > (N+1)B_S(l) - lC_S(l)$$

Applying the assumption of Eq 84 we can simplify the previous equation into:

$$NB_S(l+1) - lC_S(l+1) > NB_S(l) - lC_S(l)$$

We reformulate this equation to place the benefits in one side versus the costs on the other side as follows:

$$N\left[B_S(l+1) - B_S(l)\right] > l\left[C_S(l+1) - C_S(l)\right]$$

Claim 2: *Provided that N times the payoff of joining as smart is bigger than l times the cost of joining as a smart node, it will make sense for a new node to join as smart from overall network payoff point of view.*

The result reflects the fact that costs are incurred only by *l* smart nodes while all the connected nodes increase the network value (i.e. Metcalfe's law) by being reachable. We can also see that every new smart node creates a cost in terms of overhead meaning there is a limit where it does not make sense for the network to accept new nodes.

Eq 76 indicates that the pure strategy profile $\hat{s}$ constitutes a Nash equilibrium if none of the users can unilaterally increase their payoff by changing their strategy.

Next we analyze the Nash equilibrium considering a simultaneous game. We consider that when a node joins the network for the first time, it checks the available resources and decides to play as ordinary, smart or non-cooperative node without previous knowledge of how other nodes are playing. We assume that *k* nodes are playing as cooperative from which, *l* nodes play as smart. Now as defined in Eq 76 we study whether any of the cooperative nodes can unilaterally increase their individual payoff and relative share by changing their strategy.

Thus, if we consider that the standard monetary payoff $m_i$ is equal to the individual payoff (i.e. $m_i = u_i$) and the relative share $n_i$ is equal to the individual payoff versus the network payoff (i.e. $n_i = \frac{m_i}{\sum m_j} = \frac{u_i}{u_N}$).

In order to have a Nash equilibrium the following changes in the node strategy should not increase their payoff.

1. An ordinary node cannot increase its individual payoff and relative share by becoming non-cooperative.

   $\alpha_i m_i^O + \beta_i n_i^O \geq \alpha_i m_i^{nc} + \beta_i n_i^{nc}$

2. A smart node cannot increase its individual payoff and relative share by becoming ordinary.

   $\alpha_i m_i^S + \beta_i n_i^S \geq \alpha_i m_i^O + \beta_i n_i^O$

We analyse case 1 where the payoff for the node should not increase after the move. Thus, we replace the individual and relative share obtaining the following:

$$\alpha_i[B(k)+B_S(l)-C(k)]+\beta_i\left(\frac{B(k)+B_S(l)-C(k)}{N[B(k)+B_S(l)]-kC(k)-lC_S(l)}\right)\geq$$
$$\alpha_i[B(k-1)+B_S(l)]+\beta_i\left(\frac{B(k-1)+B_S(l-1)}{N[B(k-1)+B_S(l-1)]-(k-1)C(k-1)-lC_S(l)}\right)$$

The values of $\alpha_i, \beta_i$ cannot be determined but they remain the same throughout the whole process so we can simplify the equation as follows:

$$\beta_i\left[\left(\frac{B(k)+B_S(l)-C(k)}{N[B(k)+B_S(l)]-kC(k)-lC_S(l)}\right)-\left(\frac{B(k-1)+B_S(l-1)}{N[B(k-1)+B_S(l-1)]-(k-1)C(k-1)-lC_S(l)}\right)\right]\geq$$
$$\alpha_i[(B(k-1)+B_S(l))-(B(k)+B_S(l)-C(k))]$$

In order to simplify the previous equation we assume that the cost to be an ordinary node is proportional to the number of nodes participating as cooperative $C(k)=ak$ and that the cost of SARP routing functionality in smart nodes is proportional to the number of nodes participating as smart nodes $C_S(l)=bl$ where $a<b$. The benefit for the ordinary nodes is proportional to the number of nodes that can be reached $B(k)=cN$, and the benefit for the smart nodes is proportional to the number of nodes that are reachable $B_S(l)=dN$, where in order to reflect the fact that with smart nodes the proportion of nodes that can be reached and therefore the benefit (i.e. $B(k)<B_S(l)$) is higher than with ordinary nodes, we assume that $c<d$. The benefit when reducing the number of ordinary or smart nodes is difficult to estimate because they can be part of a critical link. We assume that the benefit and cost is reduced by a factor of $\sigma$ where $0\leq\sigma\leq1$ so we obtain that $B(k-1)=c\sigma N$, $B_S(l-1)=d\sigma N$ and $C(k-1)=a\sigma k$. In order to simplify, we consider the same factor $\sigma$ for both ordinary and smart nodes, despite that the benefit loss can be higher when reducing the number of smart nodes.

$$\beta_i\left[\left(\frac{cN+dN-ak}{N[cN+dN]-ak^2-b\eta^2k^2}\right)-\left(\frac{\sigma cN+\sigma dN}{N\sigma[cN+dN]-a\sigma k^2-b\eta^2k^2}\right)\right]\geq$$
$$\alpha_i[(c\sigma N+dN)-(cN+dN-ak)]$$

$$\beta_i\left[\left(\frac{N(c+d)-ak}{N^2[c+d]-k^2(a+b\eta^2)}\right)-\left(\frac{\sigma N(c+d)}{\sigma N^2[c+d]-k^2(\sigma a+b\eta^2)}\right)\right]\geq\alpha_i[cN(\sigma-1)+ak]$$

Considering $\delta(k)=\frac{\left(\frac{N(c+d)-ak}{N^2[c+d]-k^2(a+b\eta^2)}\right)-\left(\frac{\sigma N(c+d)}{\sigma N^2[c+d]-k^2(a+b\eta^2)}\right)}{[cN(\sigma-1)+ak]}$

then node *i* will remain as ordinary if $\delta(k)\geq\frac{\alpha_i}{\beta_i}$

The main challenge in Ad hoc networks is the participation of nodes as just ordinary nodes to create the network. Afterwards, the nodes need to have additional incentives to participate as smart nodes and contribute in the FDVB, which is required for implementing SARP to improve the performance and network scalability. The nodes have to create the basic Ad hoc networking capability first, and after that support the FDVB.

If $\delta(k)<0$, the incentives for the nodes are to become non-cooperative and we cannot have a group with *k* cooperative nodes. The Nash equilibrium conditions imply that if $\delta(k)>0$ then we can have *k* cooperative nodes and *N*-*k* non-cooperative nodes since their strategy is to remain as ordinary instead of becoming non-cooperative. This is necessary but not sufficient condition to obtain first a group with *k* ordinary nodes in the network.

Then $\delta(k)=\frac{\left(\frac{N(c+d)-ak}{N^2[c+d]-k^2(a+b\eta^2)}\right)-\left(\frac{\sigma N(c+d)}{\sigma N^2[c+d]-k^2(\sigma a+b\eta^2)}\right)}{[cN(\sigma-1)+ak]}>0$

Then $\delta(k)=\left(\frac{N(c+d)-ak}{N^2[c+d]-k^2(a+b\eta^2)}\right)-\left(\frac{\sigma N(c+d)}{\sigma N^2[c+d]-k^2(\sigma a+b\eta^2)}\right)>0$

If we consider that $\sigma\approx 1$ then $N(c+d)-ak>N(c+d)\rightarrow ak<0$

If we consider that $\sigma\approx 0$ then $N(c+d)>ak\rightarrow k<\frac{(c+d)}{a}N$

We obtain that $ak<0$ proving that we cannot reach the equilibrium since this means that there are no cooperative nodes and the network is formed by non-cooperative nodes. In the case that $\sigma\approx 0$ we obtain that the number of cooperative nodes is lower than the total number of nodes. This means that a certain percentage (i.e. $\frac{(c+d)}{a}$) of the total number of nodes N are non-cooperative, which leads to some unfairness in the network payoff.

Next we analyse case 2 where the payoff for the node should not increase after the move to become ordinary. Thus, we replace the individual and relative share obtaining the following:

$$\alpha_i[B(k)+B_S(l)-C(k)-C_S(l)]+\beta_i\left(\frac{B(k)+B_S(l)-C(k)-C_S(l)}{N[B(k)+B_S(l)]-kC(k)-lC_S(l)}\right)\geq$$
$$\alpha_i[B(k)+B_S(l-1)-C(k)]+\beta_i\left(\frac{B(k)+B_S(l-1)-C(k)}{N[B(k)+B_S(l-1)]-kC(k)-lC_S(l-1)}\right)$$

$$\beta_i\left(\frac{B(k)+B_S(l)-C(k)-C_S(l)}{N[B(k)+B_S(l)]-kC(k)-lC_S(l)}-\frac{B(k)+B_S(l-1)-C(k)}{N[B(k)+B_S(l-1)]-kC(k)-lC_S(l-1)}\right)\geq\alpha_i[B_S(l-1)-B_S(l)+C_S(l)]$$

Considering that $C(k)=ak$, $C_S(l)=bl$, $B(k)=cN$, $B_S(l)=dN$, $B(k-1)=c\sigma N$, $B_S(l-1)=d\sigma N$ and $C(k-1)=a\sigma k$ then we obtain the following.

$$\beta_i\left(\frac{N(c+d)-ak-bl}{N^2[c+d]-ak^2-bl^2}-\frac{N(c+d\sigma)-ak}{N^2[c+d\sigma]-ak^2-\sigma l^2}\right)\geq\alpha_i[dN(\sigma-1)+bl]$$

Thus the node *i* will continue as smart if $\gamma(l)\geq\frac{\alpha_i}{\beta_i}$ where

$$\gamma(l)=\frac{\left(\frac{N(c+d)-ak-bl}{N^2[c+d]-ak^2-bl^2}-\frac{N(c+d\sigma)-ak}{N^2[c+d\sigma]-ak^2-dl^2}\right)}{[dN(\sigma-1)+b1]}$$

If $\gamma(l)<0$, the incentives for the nodes is to become ordinary and we cannot have a group with *l* smart nodes since the nodes do not get a positive payoff. However, if $\gamma(l)>0$ then we have *l* smart nodes within the *k* cooperative nodes since their strategy is to remain smart instead of becoming ordinary.

Therefore we will have that $\gamma(l)>0$ if

$$\frac{\left(\frac{N(c+d)-ak-bl}{N^2[c+d]-ak^2-bl^2}-\frac{N(c+d\sigma)-ak}{N^2[c+d\sigma]-ak^2-dl^2}\right)}{[dN(\sigma-1)+b1]}>0$$

$$\left(\frac{N(c+d)-ak-bl}{N^2[c+d]-ak^2-bl^2}-\frac{N(c+d\sigma)-ak}{N^2[c+d\sigma]-ak^2-dl^2}\right)>0$$

If we consider that $\sigma \approx 1$ then $N(c+d) - ak - bl > N(c+d) - ak \rightarrow bl < 0$

If we consider that $\sigma \approx 0$ then we obtain that $N(c+d) - ak - bl > Nc - ak \rightarrow Nd > bl$ which makes sense and indicates the number of smart nodes should be lower than the total number of nodes in the network. In order to have $\sigma \approx 0$ the benefit obtained from the smart nodes when the node moves from smart to ordinary becomes zero.

This situation could be possible if the smart node has a critical role in maintaining the topology of the network so if the node becomes ordinary the network is fragmented and the benefit becomes close to zero.

We can conclude that:

1) For any payoff the game with ERC preferences cannot reach an equilibrium where some nodes act as ordinary while other nodes which are also part of the network behave as non-cooperative. We need additional incentives provided by punishment mechanisms to motivate the nodes not to become non-cooperative.
2) If we reach the basic equilibrium based on ordinary nodes then we cannot reach an equilibrium with ERC preferences including smart and ordinary nodes. We need additional incentives provided by rewarding mechanisms to motivate the nodes to become smart nodes.

From these results we see that if we do not consider ERC all nodes act as non-cooperative, since they are in equilibrium as already indicated in Table 13. ERC does not introduce a new equilibrium and still the network cannot reach an equilibrium that optimises the individual and network payoffs.

Therefore, in order to reach an equilibrium we need to increase the network incentives for cooperative nodes to participate and increase the network payoff. We also need a punishment mechanism to enforce the participation of the non-cooperative nodes. Once we have the participation of the cooperative nodes we can introduce smart nodes if the payoff obtained from ordinary and smart nodes is higher than the cost associated with ordinary nodes, and the number of smart nodes is lower than the number of ordinary nodes.

Thus, in order to ensure the nodes participation in the FDVB we should consider a rewarding mechanism to increase their payoff. The participation of smart nodes will be rewarded in terms of traffic prioritisation when their resources are lower and they behave as ordinary nodes. While the node is smart it collects some benefits that it will utilize as ordinary node when its communications require traffic prioritization. When the nodes join the Ad hoc network, they know their available resources (e.g. energy, computational power, memory, etc) and will automatically decide to cooperate if the resources are above a certain threshold that guarantees the normal functionality of the device.

The use of a rewarding factor modifies the position of the maximum of the payoff function. This factor provides the incentives to a non-cooperative or ordinary node to become smart. The players want to keep their energy at the maximum level while being able to communicate, and in some conditions have higher priority for their flows. The player preferences are to communicate with other nodes with a minimum cost (e.g. considering the costs as the energy consumption and the price in case the access to external networks require some payment).

In the literature there are already proposals for implementing the proposed reward mechanism [64] using tokens or counters that are used as payment to forward every packet on each hop through the network. To ensure the reliability of the payoff function, the proposed rewarding mechanism has to be implemented in a secure manner and a monitoring mechanism is also required for punishment in case some malicious nodes decide to cheat.

Implementing the payoff function as part of the SARP routing protocol requires a cross-layer architecture ensuring that the rewarding mechanism is enforced within the network. The rewarding mechanism will consist of prioritising the traffic of the nodes with major contribution to the routing functionality while assigning a lower priority to the traffic of the other participating nodes. This proposal requires implementing in all the nodes a fairness mechanism in the MAC layer for packet forwarding. This mechanism could be implemented on top of the IEEE 802.11 MAC protocol without changing the standard behaviour other than adding some queues with different priorities for the packet forwarding. However, this packet prioritisation needs to be secure to avoid missusage by the malicious nodes. Therefore, this mechanism should be implemented with the routing function, which is something inbuilt in the device. The users cannot easily tamper this functionality that will implement the proposed rewarding mechanism required in the *FDVB algorithm*.

# Chapter 5

## Hybrid Ad hoc Routing Approach Implementation

This chapter analyses the implementation of SARP protocol including the required AODV extensions and the cross-layer architecture. The performance results of SARP implementation are presented in the last section.

This thesis defines SARP, a cluster-based routing protocol, to solve the scalability problems in Ad hoc networks. SARP proposes a node classification that differentiates ordinary versus smart nodes. The ordinary nodes implement a reactive protocol such as AODV, while the smart nodes implement both reactive and proactive routing protocols like AODV and OLSR. This allows the smart nodes to communicate between them using OLSR. The smart nodes can also communicate with ordinary nodes using AODV. However, implementing two different protocols in the same node is not efficient and instead we propose using AODV with new extensions to include a clustering or proactive behaviour. SARP also requires implementing a rewarding mechanism for the smart and ordinary nodes. Therefore, a new architecture based on a cross-layer interaction binding the MAC with the routing layer in order to provide a fairness algorithm based on a rewarding system is required.

Figure 44 shows the logical architecture of the SARP implementation based on AODV, where we present the required modules that will be analyzed in the following sections.

- Node classification module.
- Rewarding QoS/MAC module.
- AODV extensions for cluster routing and clustering information cache.

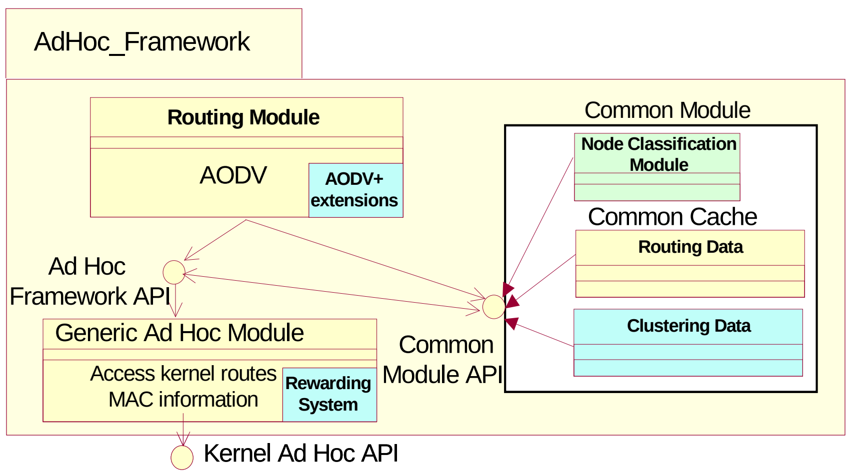


**Figure 44. SARP logical architecture.**

### 5.1 Node Classification Module

SARP is based on the concept of node classification where the nodes are classified as smart or ordinary. The ordinary nodes implement the basic reactive routing functionality while smart nodes in addition to the reactive routing implement proactive functionality. A smart node can become ordinary at any point in time when either its resources (e.g. battery life, memory, etc) decrease below a certain threshold or its mobility increases above a threshold.

Figure 45 shows using the Unified Modelling Language (UML) notation, the logic that has to be implemented in ordinary nodes to either become smart or remain as ordinary. This logic requires several modules to take care of several tasks such as node classification, training of the links between smart nodes active, and the implementation of the rewarding system using priority queues for packet forwarding.

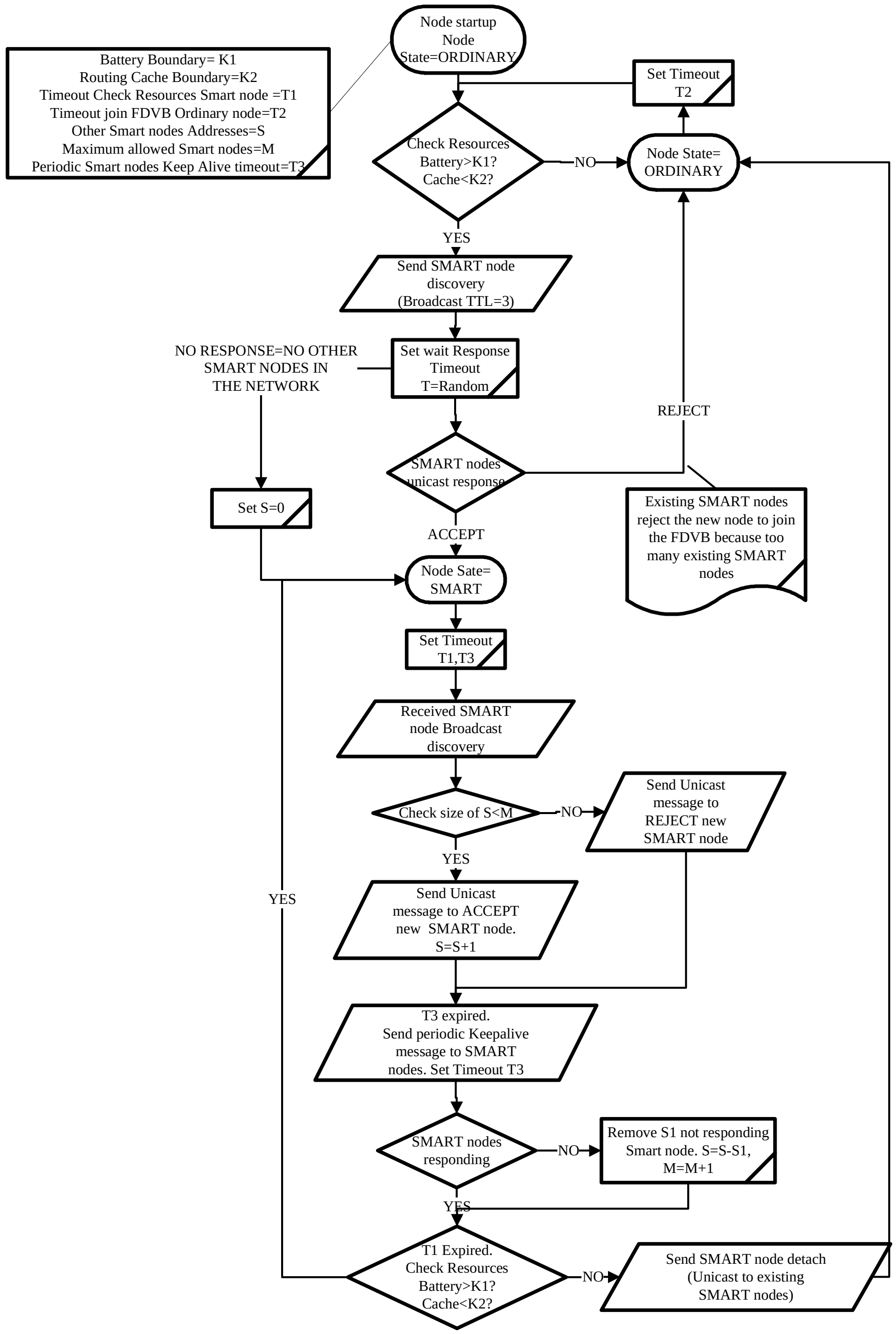


**Figure 45. Smart node selection state machine.**

In Figure 45 we can see that implementing the node classification requires a module that checks the node mobility, connectivity and other resources available to automatically decide whether the node is smart or ordinary. If the node is smart it will be communicated to the routing modules in order to implement the smart node attach procedure. The node classification module keeps checking the variables listed below and if they change over time and exceed a certain threshold it will communicate to the routing modules that the node has to become ordinary. Upon receiving this notification the routing module will implement the smart node detach procedure. The node classification module checks those variables periodically and if they change back to a certain level (i.e. the node starts charging) the node can become smart again at any point in time.

The variables considered in the node classification are the following:

- Mobility. Nodes with low mobility are more stable and capable of providing reasonable QoS.
- Connectivity. Quality of the connections with the neighbour nodes.
- Battery status. To guarantee a reasonable node lifetime.
- Memory Consumption. Memory consumed to maintain large routing information (e.g. routing entries in the cache).
- Local resources. The processing resources (e.g. CPU load) in the node will indicate whether the node is capable of maintaining extra routing functionalities.

The smart nodes also have to implement the proactive routing logic so they maintain a keep alive process to ensure that the links are active and they maintain up to date topology information. Thus, in case any of the smart nodes goes down or a link is broken the rest of smart nodes will notice the change and will allow new smart nodes to join the FDVB.

## 5.2 QoS Integrated with MAC Rewarding Module

SARP requires a cross-layer architecture for implementing a fairness algorithm and a module for selecting the smart or ordinary nodes behaviour. The cross-layer architecture consists of a direct binding between the link layer (i.e. MAC layer) and the network layer (i.e. routing protocol) to exchange information. In the SARP implementation, the smart nodes gain some benefits in terms of QoS. The routing layer informs the rewarding module in the MAC layer when the node is acting as a smart node and the rewarding module start collecting rewards. When the node changes back to ordinary mode the routing layer informs the rewarding module to stop collecting rewards. The rewards collected while the node was smart are stored in the MAC layer and they can be used for the real time applications that require higher QoS.

When an application requires higher priority for its packets (e.g. real time applications), it will inform the rewarding module that will start using the accumulated rewards and will indicate the IP stack to tag those packets using the ToS field in the IP header and will indicate the MAC layer to buffer the packets with higher priority in the transmission queues. The network and MAC layer prioritizes those packets and decrements the accumulated rewards unless the application indicates otherwise. The rewarding module in the MAC layer will inform the application when there are no more rewards. The intermediate nodes will receive those packets tagged with higher ToS and the IP stack will inform the rewarding module that will indicate the MAC layer to buffer them with higher priority in the transmission queues.

Figure 44 shows in blue the rewarding and packet priority assignment module implemented in the Ad hoc module to interact with the MAC layer. The rewarding and packet tagging is implemented in the MAC layer to avoid malicious usage. Moreover, an interface is provided to the routing and application layer to indicate when it has to start collecting rewards or using them and tagging the packets with higher priority. The proposed rewarding mechanism has to be implemented in all the nodes to ensure that the priorities are respected across the network. The MAC layer will prioritize the packets based on the assigned QoS (i.e. value of ToS field in IP header) and put each packet to be sent in queues with different priorities. The rewarding mechanism should be part of the driver that handles the wireless communications to guarantee that all the nodes interpret the priorities assigned to the packets in the same way.

This solution is backward compatible because the nodes that do not support the rewarding mechanism will implement packet forwarding as usual. Therefore, the routing algorithm in smart nodes can optimize the routes by assigning higher priority to the routes with a higher number of smart nodes capable of interpreting the packet priorities.

### 5.3 Cluster Routing Extensions Module for SARP Implementation in AODV

AODV already supports connectivity with the public Internet by using a gateway address stored in the routing cache. However, in large Ad hoc networks, AODV suffers from big delays and route discovery latency. AODV is suitable for small networks where the delay for finding new routes is low. Nevertheless, the extensibility of AODV makes it a good basis for the Scalable Ad hoc Routing Protocol. The benefit of implementing SARP using only a reactive protocol such as AODV instead of utilising both reactive and proactive protocols (e.g. AODV and OLSR as used in the test bed) is the simplicity and avoiding fragmented solutions. Thus, extending AODV with a new message type for sharing or updating information between smart nodes means that no additional proactive protocol is needed. The proposed extension will implement a broadcast algorithm that will use border nodes similarly to the bordercast routing protocol in ZRP [41]. The border nodes are the smart nodes in the node taxonomy proposal and they are identified during the initial neighbour discovery process in AODV.

SARP can be deployed by extending AODV with a broadcast algorithm for working with medium to large Ad hoc networks. This implementation allows the interaction of nodes running the extended version of AODV and nodes with the standard AODV protocol. The nodes with the standard AODV will discard the new broadcast messages.

After a node has decided to become smart, it will initiate the network attachment, which consists of finding other smart nodes in the network. Figure 46 shows the smart node attachment procedure (i.e. new node depicted in orange colour) initiated by sending a broadcast message with TTL=3. We limit the TTL=3 to reduce the flooding and the delay required to form the FDVB. However, if the smart nodes are grouped in areas of the network where the closest smart nodes are separated by more than 3 hops then multiple FDVBs will be created in different parts of the same network. The new smart node will receive information about other smart nodes in the response from the neighbour smart nodes located maximum 3 hops away.

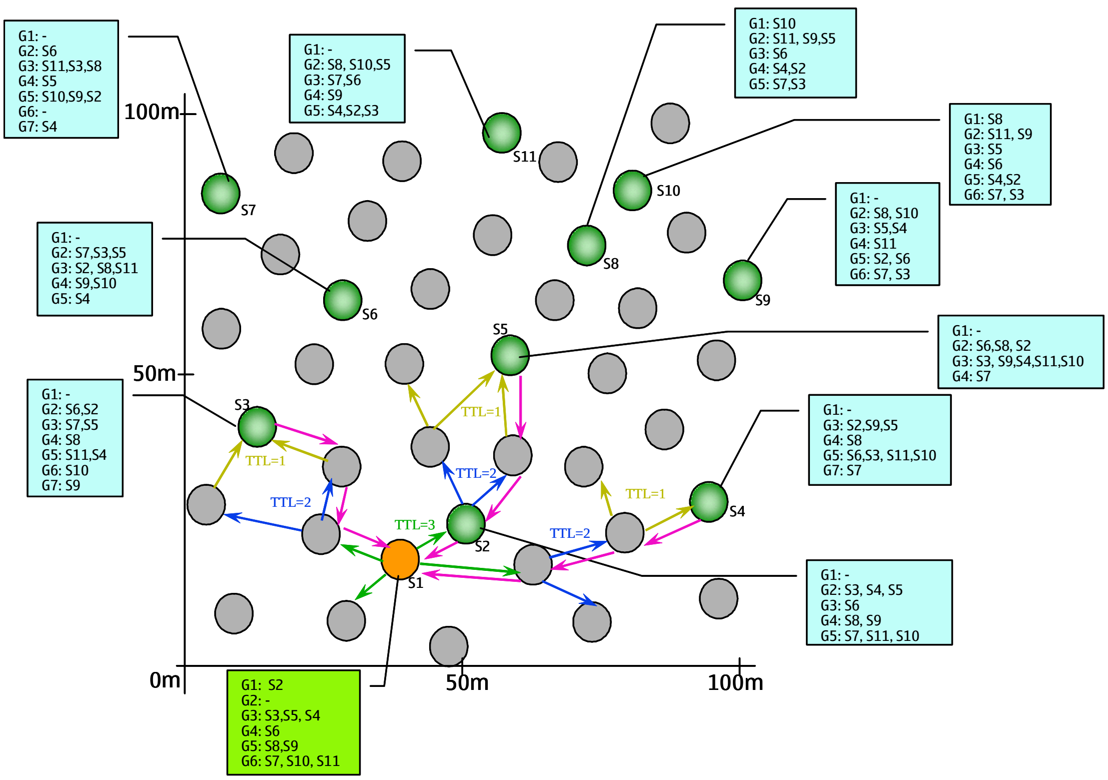


**Figure 46. Finding the AODV border nodes with the smart node attachment process.**

The network attachment includes a control mechanism to limit the number of smart nodes within a certain area. Therefore, the neighbour smart nodes will respond indicating whether the new node can join the FDVB or not. If any of them denies the attachment, the new node will not join as smart. In case no response is received or all the responses received from the existing smart nodes are positive the node can join as smart. The new smart node will utilise the received information to create different hierarchical groups categorizing the smart nodes (i.e. $G_1...G_{10}$), based on the number of hops distance. $G_\gamma$ is the group of smart nodes that are $\gamma$ hops away from the new smart node. Once the groups are created, the smart node will send periodically routing information updates to the smart nodes on each group. The hierarchical groups will implement fuzzy topology information sharing. The routing information exchange is periodical instead of event-triggered to avoid frequent link state updates caused by link breaks (i.e. unreliable wireless links and mobility) or expiration of routes in the AODV cache. The frequency of the periodical updates will vary for different groups depending on the number of hops distance. Therefore, the smart nodes nearby the current node will propagate the routing information more often than the smart nodes in groups far away. According to this, the frequency of updates to the group 1 ($G_1$ in Figure 47) is higher than to the group 6 ($G_6$ in Figure 47).

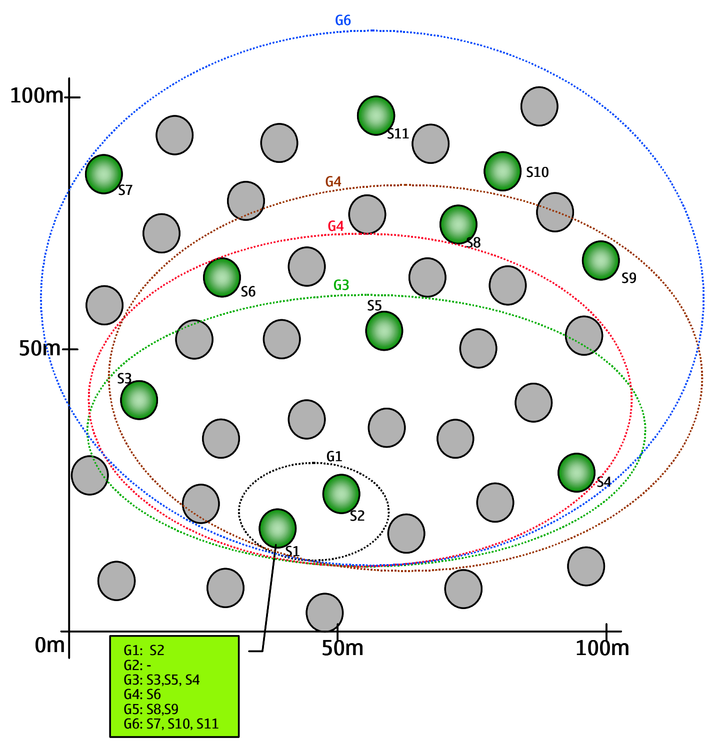


**Figure 47. AODV border nodes groups defined by the new smart node.**

The proposed algorithm is named as Hazy Sighted Link State (HSLS), which is introduced in the Fuzzy Sighted Link State (FSLS) routing [47]. The HSLS algorithm consists of sending Link State Updates (LSU) every $2^k T$ to a scope of $2^k$, where $k$ is the number of hops and $T$ is the minimum LSU transmission period. This approach reduces the overhead by limiting the scope of link state update dissemination in space and over time. The nodes in the same group will share the routing information that smart nodes maintain in their routing cache. This will disseminate more accurate distance and path information about the area around the current node. The smart node will have imprecise knowledge of the best path to a distant destination. However, this imprecision decreases progressively when the packet approaches the destination. The delay in the routing process decreases by having the fuzzy topology information. Therefore, when the smart node receives a route request it will check the routing information obtained from each group. In case the destination is not found in the cache, the smart node will initiate a standard broadcast route request but in addition the smart node will send a unicast route request to the border nodes listed on each group to speed up the route discovery for nodes located in large distances.

Figure 48 shows the FDVB obtained from grouping the smart nodes. In this topology there are nodes that implement standard AODV with the reactive behaviour necessary for supporting applications with real time requirements. In the same network there are nodes that include the proposed proactive AODV extensions implementing SARP protocol to provide the hierarchical benefits for large networks.

The additional routing overhead required for creating the FDVB and maintaining the routing information between smart nodes is similar to the proactive routing overhead, which was modelled in previous chapters. Moreover, the grouping of smart nodes within FDVB following the FSLS routing algorithm and the fact that not all nodes in an Ad hoc network belong to the FDVB reduces the overhead compared to standard proactive routing protocols.

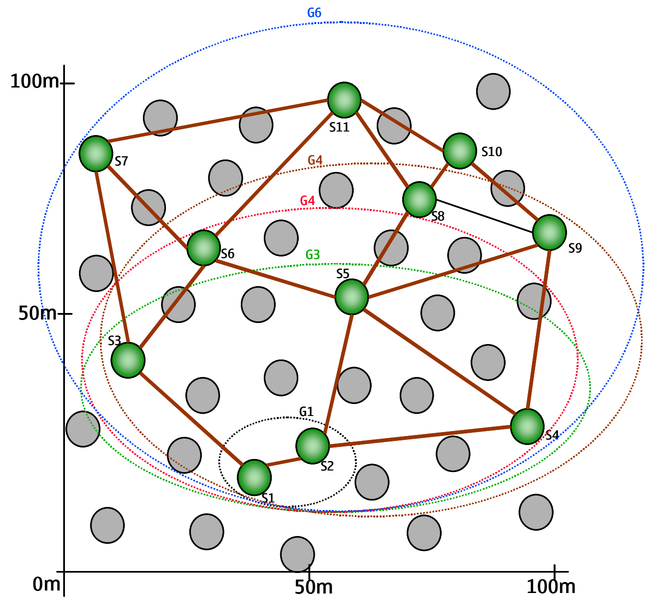


**Figure 48. SARP fully distributed virtual backbone.**

## 5.4 Performance Results of SARP Implementation

This section presents the performance results after implementing and testing SARP with AODV and OLSR in a small scale network. Figure 49 shows the test bed with the Ad hoc network with only 1 smart node implementing SARP (i.e. Node 1 in Figure 49) and 4 ordinary nodes implementing AODV. In this scenario we cannot test the benefits of SARP to its full extent in large networks but it provides results about the behaviour of SARP in a moderate size network giving a basis for performance comparisons. In the test bed we have performed manual breaks of the links between nodes to force topology changes and route recoveries. Moreover, the fluctuations in the signal and other obstacles provide a dynamic topology.

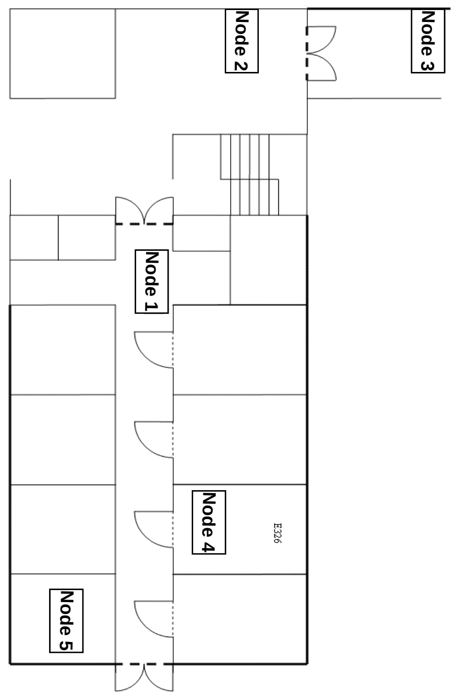


**Figure 49. Test bed environment for SARP protocol.**

Figure 50 shows the percentage of packet loss results in three different scenarios.

1) OLSR, where all the nodes are running OLSR.
2) AODV, where all the nodes are running AODV.
3) SARP, where one smart node is running SARP and four ordinary nodes are running AODV.

The set up is the one used in the VoIP test bed and the selected traffic is a Constant Bit Rate (CBR) of 15packets/second over UDP used previously in the voice sessions transmitting 20ms voice packets encapsulated with GSM codec [48] and using Real Time Protocol (RTP) [49] protocol over UDP as represented previously.

Figure 50 shows that the percentage of packet loss is the highest in the OLSR network while it is the lowest in the AODV scenario. SARP improves the percentage of packet loss compared to the OLSR scenario.

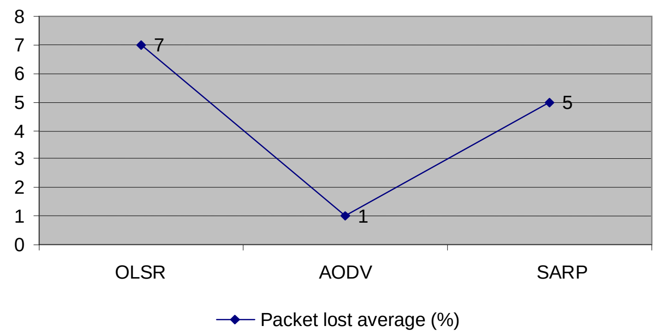


**Figure 50. Packet loss in test bed with Ad hoc framework.**

Figure 51 shows the routing latency results for the same scenarios. The OLSR scenario gives lower average delay compared to the AODV scenario due to the route availability in the routing table. In the SARP scenario, the routing latency gives results similar to the OLSR scenario since the route is available in the routing table.

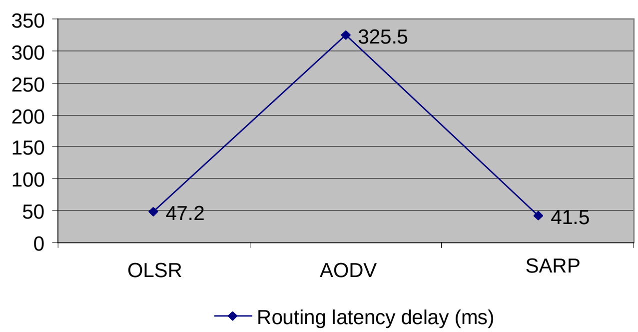


**Figure 51. Routing latency in test bed with Ad hoc framework.**

The percentage of packet loss and routing latency results show that SARP improves the performance compared to having either AODV or OLSR protocols running in the Ad hoc network. However, the real benefit of SARP is visible in large networks where the FDVB increases the network performance and scalability.

# Chapter 6

## Conclusions

The goal of this thesis has been to enhance scalability in Ad hoc networks. We have studied different routing protocols and evaluated their performance. We have demonstrated that currently there is no single protocol that accommodates the different requirements in Ad hoc networks. Therefore, we have designed and implemented a new hybrid routing protocol named Scalable Ad hoc Routing Protocol (SARP) to enable scalability and meet the different demands of the nodes in Ad hoc networks.

### 6.1 Results

The results obtained from the simulations and the test bed differ and even contradict in some cases. For that reason, we have concluded that the results from the simulations are not reliable enough to determine the performance of Ad hoc routing protocols. The simulations results can be used to estimate the network performance with different routing protocols but the results from the test bed are needed to confirm or interpret and in some cases correct the simulation results.

From the simulations and the test bed results we have concluded that a hybrid solution is the optimal routing protocol to enable scalability in Ad hoc networks. However, none of the existing hybrid routing protocols fulfil the Ad hoc networks requirements and a new protocol had to be designed.

SARP applies the advantages from AODV in small scale networks and the advantages of OLSR for distributing the optimal routes to reach larger distances. This protocol design avoids the excessive traffic generated by reactive routing when discovering new routes over a large network. SARP has been designed based on the results from the simulations and a small scale test bed, and a mathematical model has been defined to theoretically evaluate its performance.

The mathematical model shows that in networks with low mobility and a higher number of smart nodes than ordinary nodes, the connectivity increases but the network lifetime decreases. On the other hand, in networks with high mobility and a higher number of ordinary nodes than smart nodes, both the connectivity and the network lifetime increase.

In all the cases under study the available bandwidth increases after introducing a small number of smart nodes. However, a high number of smart nodes have a negative effect on bandwidth. Therefore, maintaining the number of smart nodes in the network under a certain limit improves considerably the available bandwidth when the size of the network increases.

The results from the mathematical model demonstrate that a balance between the number of smart and ordinary nodes is required to have reliable connectivity and longer network lifetime with enough bandwidth. This conclusion supports the SARP design with few nodes that implement proactive routing besides reactive routing. We can reach longer distances within the network through optimal routes and with a reliable connectivity with those few nodes. Thus, when the routing protocol calculates the optimal routes it has to minimize the number of hops but also select the routes with a higher number of smart nodes.

SARP enables Ad hoc network scalability but requires that some nodes spend additional resources to participate in the packet forwarding and extra routing functionality, which may lead into unfairness.

We applied game theory to evaluate the incentives for implementing SARP. The evaluation shows that the equilibrium on individual payoffs is obtained when all the nodes avoid participating in the Ad hoc network behaving as non-cooperative nodes. However, if we consider not only the individual payoff but also the network payoff then the nodes have extra incentives but still no equilibrium can be reached. The analysis shows that we need to have incentives for nodes to become ordinary and punishment mechanism to motivate the nodes to remain as ordinary instead of becoming non-cooperative. If we have the basic network running based on ordinary nodes then we can have a certain number of ordinary nodes running a reactive protocol and some of them as smart running a proactive protocol since the network provides enough incentives to reach an equilibrium. We have obtained results indicating that having cooperative nodes is critical to reach Nash equilibrium. Another finding consists on the fact that the number of smart nodes should be lower than the total number of cooperative nodes in order to reach the equilibrium.

In order to guarantee that the nodes will have additional network incentives, a rewarding mechanism has been studied. This ensures the participation of the Ad hoc nodes as ordinary and smart contributing to the network scalability. A cross-layer architecture has been designed to implement the rewarding mechanism. With this approach the Ad hoc nodes obtain a fair added value in return for their contribution to the routing functionality.

A small scale test gives some results from the benefits of using SARP instead of OLSR or the standard AODV. However, SARP performance has not been tested in large scale networks (i.e. N>1000) and the scalability enhancements remain to be measured. Also, the rewarding mechanism has not been implemented and tested to verify the nodes get the required incentives to participate and increase the performance.

## 6.2 Summary

This thesis is structured in three main sections. Firstly, we evaluate the performance of the existing routing protocols using simulations. We formulate some propositions to generalize the behaviour of the different routing protocols, and we verify those propositions with the results obtained from a test bed. Based on the results we propose the Scalable Ad hoc Routing Protocol (SARP).

Secondly, in order to evaluate theoretically the performance of SARP we propose a mathematical model for Ad hoc networks using different metrics. The results show that the connectivity and the bandwidth improve with a certain percentage of smart nodes.

Thirdly, after proving that the network scalability improves with SARP, we analyze the node incentives required for its implementation. We apply game theory to verify that there are incentives to implement SARP and reach an equilibrium. As a result of this analysis we concluded that an extra rewarding mechanism is needed to increase the incentives and ensure that there will be a minimum percentage of smart nodes in the network guaranteeing the optimal performance. A cross layer architecture is required for implementing these additional incentives. The SARP implementation based on AODV with additional broadcast messages has been presented, and some results from a small scale test bed are included in the last part of the thesis.

## 6.3 Future Research

This thesis has addressed the problem of scalability in Ad hoc networks. We have defined a new protocol that can to some extent overcome the limitations of large scale Ad hoc networks. We have demonstrated that nodes will get incentives for implementing SARP but additional rewarding might be required for increasing the network payoff. The game analysis and the mathematical model show that there is a threshold in the number of smart nodes required to reach an optimum equilibrium. Therefore, SARP includes an access control mechanism to limit the number of smart nodes in the FDVB. However, future study is required to determine the means that will allow the nodes to reach and maintain the equilibrium.

SARP relies on a cross-layer architecture to reward the nodes that participate in the routing functionality. However, the rewarding mechanism only benefits the smart nodes but also the participation of ordinary nodes is necessary in the Ad hoc network. Thus, a more complex rewarding system is required to ensure not only the participation of the smart but also the ordinary nodes. This rewarding mechanism can be associated with a QoS system that will benefit all the nodes participating in the Ad hoc network routing functionality. This solution will not only provide additional incentives for the nodes but also will increase the QoS in the network.

As part of the future development, the rewarding mechanism and the associated QoS system should be implemented and tested. We need to prove that the rewarding system with enhanced QoS for the nodes participating in the FDVB will increase the network performance compared to the best effort service equally used by all the nodes. We have to consider that the available resources (i.e. communication channels) in Ad hoc networks are limited. Thus, a rewarding proposal will increase the incentives for ordinary nodes to become smart nodes and participate in the FDVB since their traffic will be prioritised. Nevertheless, the overall network performance may decrease since the rest of ordinary nodes in the network will receive lower priority for their traffic.

Therefore, the implementation of the rewarding system may increase the motivation for the ordinary nodes to participate in the FDVB but it may also increase the incentives for ordinary nodes to become non-cooperative instead of remaining as ordinary nodes. The reason is that the QoS will increase for smart nodes but it will decrease for ordinary nodes and it will decrease also for non-cooperative nodes.

Thus, in addition to the rewarding mechanism we should improve the incentive for participating as ordinary nodes by including punishment mechanisms [62] for the non-cooperative nodes. This mechanism would motivate the non-cooperative nodes to participate either as ordinary or smart nodes.

SARP has been implemented and tested only in a small scale network. However, to fully analyze the SARP behavior, a medium to large network (N>1000) could be created in order to prove the SARP scalability benefits.

SARP implementation is based on AODV that includes inbuilt mechanism to connect the Ad hoc network with fixed networks through a gateway. The smart nodes could host this gateway to the fixed network and at the same time behave towards the rest of nodes as any other smart node that provide reliable connectivity. Thus, it would be beneficial to consider additional incentives for having the connectivity to fixed networks through those smart nodes acting as gateways. This approach would improve the network payoff since it increases the number of nodes connected to the Ad hoc network, thus promoting the usage of the Ad hoc network technology.

# References

[1] S. R. Das, R. Castaneda and J. Yan, "Simulation Based Performance Evaluation of Mobile, Ad Hoc Network Routing Protocols," *ACM/MONET Journal*, pp. 179-189, July 2000.

[2] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," ACM/IEEE MOBICOM'98, pp. 85-97, October 1998.

[3] H. Bierman and L. Fernandez, "Game Theory with Economic Applications," Addison-Wesley, 1993.

[4] C. Perkins and E. Royer and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561, July 2003.

[5] MobileMAN IST-2001-38113 Oct 02-05, http://cnd.iit.cnr.it/mobileMAN/

[6] T. Vadar, "Scalable Routing Mechanism in Ad Hoc Networks," Master Thesis, March 2006.

[7] L. Xiao, "Design and Implementation of an Ad Hoc Routing Framework," Master Thesis, May 2003.

[8] J. Gutierrez Plaza, "Design and Implementation of the OLSR Protocol in an Ad Hoc Framework," Master Thesis, October 2003.

[9] O. Leon Cadahia, "Design and Implementation of an Optimized Architecture for Service Discovery in Ad hoc networks," Master Thesis, September 2004.

[10] B. Jiang, "Clustering Extension to the Adhoc On-Demand Distance Vector Routing Protocol," Master Thesis, June 2006.

[11] J. Hijas Carretero, "Peer to Peer Applications in Ad Hoc Networks," Master Thesis, July 2006.

[12] J. Hughet, "Ad Hoc End to End Security Development," Master Thesis, September 2006.

[13] J. García Sánchez, "A Study of VoIP over Wireless Local Area Network," Master Thesis, May 2004.

[14] M. Ayyash, "Quality of Service Analysis of Real Time Applications in Ad Hoc Networks," Master Thesis, October 2006.

[15] J. Costa-Requena, T. Vadar, R. Kantola and N. Beijar, "AODV-OLSR Scalable Ad hoc Routing Proposal," ISWPC, January 2006.

[16] J. Costa-Requena, R. Kantola, M. Ayyash, J. Creado and J. Hakkinen, "VoIP and Service Discovery Testbed in Ad Hoc Networks," LANMAN, September 2005.

[17] J. Costa-Requena, M. Ayyash, T. Vardar and R. Kantola, "Scalable Hybrid Adhoc Routing Approach," *International Journal of Computer Science and Network Security (IJCSCS)*, Vol 6, N8, pp 82-98, August 2006.

[18] J. Costa-Requena, N. Beijar and R. Kantola, "Replication of Routing Tables for Mobility Management in Ad Hoc Networks," *ACM Wireless Networks (WINET) Journal*, 2003.

[19] J. Costa-Requena, R. Kantola and J. Nuevo Fonseca, "Strategies for the Formation of a Service Distribution Backbone in Ad Hoc Networks," CIIT 2004, November 2004.

[20] J. Costa-Requena, J. Gutiérrez, R. Kantola, J. Creado and N. Beijar, "Network Architecture for Scalable Ad Hoc Networks," ICT, August 2004.

[21] J. Costa Requena, R. Kantola, N. Beijar, "Mobility and Network Management in Ad Hoc Networks," CSN 2002, September 2002.

[22] J. Costa-Requena, R. Kantola, N. Beijar, "Incentive Problem for Ad Hoc Networks Scalability," ICNS, October 2005.

[23] R. Kantola, J. Costa-Requena, N. Beijar, "Interoperable routing for IN and IP Telephony," *Computer Networks*, vol. 35, no. 5, pp. 597-609, Elsevier Journal, 2001.

[24] J. Costa-Requena, I. O. Gonzalez, R. Kantola and N. Beijar, "Autoconfiguration Mechanism for IP Telephony Location Servers," IFIP TC6, October 2002.

[25] R. Kantola, J. Costa-Requena and N. Beijar, "A Common Numbering Infrastructure for IN and IP Telephony," IN Workshop, May 2000.

[26] IETF MANET working group, http://www.ietf.org/html.charters/manet-charter.html

[27] ITU-T Recommendation E.164, The international public telecommunication numbering plan, Revision 3, February 2005.

[28] J. Reynolds and J. Postel, "Assigned Numbers", RFC 1340, July 1992.

[29] V. Fuller, T. Li, J. Yu and K. Varadhan, "Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy," RFC 1519, September 1993.

[30] P.V. Mockapetris, "Domain Names-concepts and Facilities," RFC 1034, November 1987.

[31] R. Droms, "Dynamic Host Configuration Protocol," RFC 2131, March 1997.

[32] C. Perkins, "IP Mobility Support," RFC 2002, October 1996.

[33] J. Corson and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," RFC 2501, January 1999.

[34] J. Moy, "OSPF Version 2," STD0054, RFC2328, April 1998.

[35] P. Traina, "BGP-4 Protocol Analysis," RFC1774, March 1995.

[36] D. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, Chapter 5, pp. 153-181, Kluwer Academic Publishers Journal, 1996.

[37] A. Iwata, C. Chiang, G. Pei, M. Gerla and T.-W. Chen, "Scalable Routing Strategies for Ad Hoc Wireless Networks," *IEEE Journal on Selected Areas in Communications*, Special Issue on Wireless Ad Hoc Networks, vol. 17, No 8, pp. 1369-1379, August 1999.

[38] C. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," SIGCOMM, August 1994.

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

[40] D. P. Bertsekas, "Network Optimization. Continuous and Discrete Models," pgs. 73-75, Athena Scientific, 1998.

[41] M.R. Pearlman and Z.J. Haas, "Determining the Optimal Configuration for the Zone Routing Protocol," *IEEE Journal on Selected Areas in Communications*, Special Issue on Wireless Ad Hoc Networks, vol 17, No 8, pp. 1395-1414, August 1999.

[42] Y.-B. Ko and N. H. Vaidya, "Location-aided Routing (LAR) in Mobile Adhoc Networks," Mobicom98, pp. 66-75, October 1998.

[43] The Network Simulator NS-2, http://www.isi.edu/nsnam/ns/

[44] X. Hong, T. J. Kwon, M. Gerla, D. Lihui Gu, and G. Pei, "A Mobility Framework for Ad Hoc Wireless Networks," MDM, pp. 185-196, January 2001.

[45] E. Hyytia, P. Lassila, and J. Virtamo, “Spatial Node Distribution of the Random Waypoint Mobility Model with Applications,” IEEE Transactions on Mobile Computing, June 2006.

[46] M. Goto, S. Yoshida, K. Mase, and T. Clausen, “A Study of Link Buffering for OLSR,” OLSR Interoperability & Workshop, August 2004.

[47] C. Santivanez, R. Ramanathan and I. Stavrakakis, “Making Link-State Routing Scale for Ad Hoc Networks,” Mobihoc2001, October 2001.

[48] GSM library version 06.10, http://kbs.cs.tu-berlin.de/~jutta/toast.html

[49] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” RFC 1889, January 1996.

[50] M. Handley, H. Schulzrinne, E. Schooler and J. Rosenberg, “SIP: Session Initiation Protocol,” RFC 3261, June 2002.

[51] Programmers Guide to OSS, http://www.opensound.com/pguide/index.html

[52] ITU-T Recommendation E.800 (08/94), “Terms and definitions related to quality of service and network performance including dependability,” Revision 1, August 1994.

[53] Helsinki University of Technology, Networking Laboratory. “Test bed results,” June 2005, http://www.netlab.hut.fi/~mayyash/pisa/index.htm

[54] PDA HP iPAQ, http://h18000.www1.hp.com/products/quickspecs/10977_na/10977_na.HTML

[55] Linux Familiar project, http://familiar.handhelds.org/

[56] Ethereal, network protocol analyzer tool, http://www.ethereal.com/

[57] JRTPLib version 2.9, http://research.edm.luc.ac.be/jori/jrtplib/jrtplib_old.html

[58] K. Wu and J. Harms, “Performance Study of a Multipath Routing Method for Wireless Mobile Ad Hoc Networks,” MASCOTS, pp 99–107, August 2001.

[59] S. Chen, “Routing Support for Providing Guaranteed End to End Quality of Service,” http//www.cs.uiuc.edu/Dienst/UI/Describe/2.0/nc-strl.uiuc_cs/UIUCDCS-R-99-2090, UIUCDCS-R-99-2090, University of Illinois at Urbana/Champaign Urbana, Illinois, July 1999.

[60] J. F. Nash, “Non-cooperative Games,” Annals of Mathematics, 54(2):286–295, September 1951.

[61] E. Altman, A. Kherani, P. Michiardi and R. Molva, “Some Game-theoretic Problems in Wireless Ad hoc Networks,” EURO-NGI Workshop, October 2004.

[62] P. Michiardi, “Cooperation Enforcement and Network Security Mechanisms for Mobile Ad hoc Networks,” PhD Thesis Dissertation, Paris Institute of Technology, June 2005, http://pastel.paristech.org/archive/00001114/

[63] G. E. Bolton and A. Ockenfels, “ERC: A theory of equity, reciprocity and competition,” The American Economic Review, 90:166–193, March 2000.

[64] L. Butty´an and J-P. Hubaux, “Enforcing Service Availability in Mobile Ad-Hoc WANs,” Mobihoc 2000, August 2000.