Reactive-Greedy-Reactive in Unmanned Aeronautical Ad-hoc Networks: A Combinational Routing Mechanism

by

Rostam Shirani, **M.A.Sc.**

A thesis submitted to the
Faculty of Graduate and Postdoctoral Affairs
in partial fulfillment of the requirements for the degree of

Master of Applied Science in Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario
August 2011

The undersigned hereby recommends to the
Faculty of Graduate and Postdoctoral Affairs
acceptance of the thesis

Reactive-Greedy-Reactive in Unmanned Aeronautical Ad-hoc Networks: A Combinational Routing Mechanism

submitted by **Rostam Shirani**, **M.A.Sc.**

in partial fulfillment of the requirements for the degree of

Master of Applied Science in Electrical and Computer Engineering

Professor Thomas Kunz, Thesis Co-supervisor

Professor Marc St-Hilaire, Thesis Co-supervisor

Professor Howard Schwartz, Chair,
Department of Systems and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
Department of Systems and Computer Engineering
Carleton University
August, 2011

Abstract

Based on the rapid growth in the applications of Unmanned Aerial Vehicles (UAVs) in the last decade, the idea of in-flight communication among UAVs has been proposed in cooperative missions. Since medium and small sized UAVs are cheap and deployable, several of them can be used to form what we call an *Unmanned Aeronautical Ad-hoc Networks (UAANETs).*

Due to the availability of location information, geographic protocols can be an option for routing in UAANETs. Although we show that greedy geographic forwarding alone is not sufficient in UAANETs, our results illustrate that a combination of greedy geographic forwarding with a reactive mechanism, which forms a Reactive-Greedy-Reactive (RGR) routing, can be beneficial. Simulation results show that RGR outperforms existing protocols such as Ad-hoc On-demand Distance Vector (AODV), and greedy geographic forwarding in searching missions in terms of delay and packet delivery ratio while its overhead is comparable with traditional mechanisms.

In memory of Hajieh Khanom Jahan Shiran

To my parents, Manijeh and Mahmoud, and my sister Roya

Acknowledgments

I would like to express my deep and sincere gratitude to my supervisors Professors Thomas Kunz and Marc St-Hilaire for the end-less support, understanding, kindness, and great guidance. During my studies at Carleton University, they not only helped me in my research but also they taught me in my life.

I also would like to acknowledge researchers in *Mobile Ad-hoc and Sensor Network Systems* group in Communication Research Center (CRC) Canada, especially Dr. Yifeng Zhou, Dr. Jun Li, and Mrs. Louise Lamont not only for funding the project but for supporting me as a visiting researcher in CRC during the past year.

Last but not least, I would like to thank my family, my parents and my sister, who are my reasons in life.

Table of Contents

Abstract **iii**

Acknowledgments **v**

Table of Contents **vi**

List of Tables **x**

List of Figures **xi**

List of Acronyms **xv**

1 Introduction **1**

1.1 Research Objective 2

1.2 Thesis Contributions 2

1.3 Organization of the Thesis 3

2 Related Work on UAANET **5**

2.1 Introduction 5

2.2 Unmanned Aeronautical Ad-hoc Networks 6

2.2.1 UAV Operations 6

2.2.2 Applications of UAANETs 7

2.2.3 Differences between UAANETs and AANETs 8

2.3 Cooperation, Control and Path Planning 8

2.4 Networking Characteristics of UAANETs 11

2.4.1 Rapid Topology Change 11

2.4.2 Application-based Mobility 12

2.4.3 Medium Access Control Requirements 12

2.5 Reactive Routing . . . 13
2.6 Geographic Routing . . . 14
2.6.1 Location Service Module . . . 15
2.6.2 Location Prediction in Geographic Routing . . . 16
2.7 Towards a Routing Protocol for UAANETs . . . 17
2.7.1 Greedy Geographic Core in Data Forwarding . . . 18
2.7.2 Reactive-Greedy Combination . . . 18
2.8 Summary . . . 19

3 Greedy Geographic Forwarding in UAANETs 20

3.1 Introduction . . . 20
3.2 Background . . . 21
3.3 Simulation Environment . . . 22
3.4 Simulation Results . . . 24
3.5 Observations From Simulation Results . . . 27
3.6 Quadratic Estimation of Success Probability . . . 28
3.7 Summary . . . 32

4 Reactive-Greedy-Reactive Routing for UAANETs 34

4.1 Introduction . . . 34
4.2 Motivation . . . 36
4.2.1 Blind Broadcasting . . . 36
4.2.2 Geographic Reverse-route Forwarding . . . 36
4.2.3 Reactive Reverse-route Forwarding . . . 37
4.2.4 Proactive Reverse-route Forwarding . . . 37
4.2.5 Discussion on the Proposals . . . 37
4.3 Implementation of RGR in a Network Simulator . . . 38
4.3.1 Overview of AODV in OPNET . . . 39
4.3.2 RGR Implementation Overview . . . 40
4.4 RGR Functionality . . . 41
4.4.1 Elements of RGR Mechanism . . . 41
4.4.2 Node Tables . . . 43
4.4.3 Reactive Route . . . 44
4.4.4 Packet Forwarding in an Intermediate Node . . . 45
4.4.5 Route Errors . . . 46

4.4.6 Switching to Greedy Geographic Forwarding 47
4.4.7 Handling a Received Packet via Greedy Geographic Forwarding 47
4.4.8 Dropping a Packet in the Greedy Geographic Function 48
4.4.9 Global Repair by the Source 49
4.4.10 Destination Operations 49
4.5 RGR versus Reactive and Geographic Routing 51
4.6 Summary 53

5 Simulation Results 55

5.1 Introduction 55
5.1.1 Packet Delivery Ratio 56
5.1.2 Delay 57
5.1.3 Overhead 57
5.2 Mobility Modelling 58
5.2.1 Mobility Parameters for Searching Mission 59
5.2.2 Mobility Parameter for Tracking Mission 60
5.3 MAC/PHY Specifications 62
5.4 Network Traffic 62
5.5 Routing Protocol Setting 64
5.6 Simulation Results for a Searching Mission 65
5.6.1 Small Network with 10 UAVs 66
5.6.2 Large Network with 20 UAVs 73
5.7 Simulation Results for a Tracking Mission 82
5.8 Number of Switches 88
5.9 Summary 89

6 Conclusions and Future Work 91

6.1 Conclusions 91
6.2 Future Work 92

List of References 95

Appendix A OPNET Implementation 102

A.1 Adding Location Information in Header Files 102
A.2 Switching to Greedy Geographic Forwarding 103
A.3 Lifetime 103

A.4 Create Functions and Arrival Functions in RGR 104
A.5 Statistics of the Number of Switches 105

Appendix B 10 Searching UAVs in a 25 km^2 Area 107

Appendix C Searching Mission with 30 Nodes 113

Appendix D Tracking Mission with 20 Nodes 118

List of Tables

4.1 Comparison of RGR with reactive/geographic routing mechanisms . . 54

5.1 Mobility parameters of a searching scenario 60

5.2 Mobility parameters of a tracking scenario 61

5.3 MAC layer specifications 62

5.4 Traffic parameters 64

5.5 AODV/RGR configurations 65

5.6 Greedy geographic forwarding configurations 65

5.7 Average number of switches per source-destination flow in search missions 88

5.8 Average number of switches per source-destination flow in tracking missions 89

List of Figures

2.1 Block diagram of ISR mission 7

3.1 The algorithm for gathering greedy geographic forwarding statistics . 23

3.2 Effects of increasing the number of nodes on successful source-destination communications 25

3.3 Effects of increasing the average number of neighbors on successful source-destination communications 26

3.4 Effects of transmission range on successful source-destination communications 27

3.5 Greedy geographic forwarding success probability (for number of UAVs=10, 15, 20, and 25) 29

3.6 Average success probability of greedy geographic forwarding 32

4.1 The scenario to establish a reactive route from S to D 41

4.2 Handling a packet arrival in RGR 43

5.1 Average delay in a searching scenario of size 4 km^2 including 10 low speed UAVs 66

5.2 Packet delivery ratio in a searching scenario of size 4 km^2 including 10 low speed UAVs 67

5.3 Average routing traffic in a searching scenario of size 4 km^2 including 10 low speed UAVs 68

5.4 Average delay in a searching scenario of size 4 km^2 including 10 medium speed UAVs 69

5.5 Packet delivery ratio in a searching scenario of size 4 km^2 including 10 medium speed UAVs 69

5.6 Average routing traffic in a searching scenario of size 4 km^2 including 10 medium speed UAVs 70

5.7 Average delay in a searching scenario of size 4 km^2 including 10 high speed UAVs 71

5.8 Average delivery ratio in a searching scenario of size 4 km^2 including 10 high speed UAVs 71
5.9 Average routing traffic in a searching scenario of size 4 km^2 including 10 high speed UAVs 72
5.10 Average delay in a searching scenario of size 25 km^2 including 20 low speed UAVs 74
5.11 Average delivery ratio in a searching scenario of size 25 km^2 including 20 low speed UAVs 74
5.12 Average routing traffic in a searching scenario of size 25 km^2 including 20 low speed UAVs 75
5.13 Average delay in a searching scenario of size 25 km^2 including 20 medium speed UAVs 76
5.14 Average delivery ratio in a searching scenario of size 25 km^2 including 20 medium speed UAVs 76
5.15 Average routing traffic in a searching scenario of size 25 km^2 including 20 medium speed UAVs 77
5.16 Average delay in a searching scenario of size 25 km^2 including 20 high speed UAVs 78
5.17 Average delivery ratio in a searching scenario of size 25 km^2 including 20 high speed UAVs 78
5.18 Average routing traffic in a searching scenario of size 25 km^2 including 20 high speed UAVs 79
5.19 Packet delivery ratio improvement of RGR compared to AODV in different scenarios 80
5.20 Delay improvement of RGR compared to AODV in different scenarios 81
5.21 Increment in RGR overhead compared to AODV in different scenarios 81
5.22 Average delay in a tracking scenario including 10 low speed UAVs . . 82
5.23 Average delay in a tracking scenario including 10 medium speed UAVs 83
5.24 Average delay in a tracking scenario including 10 high speed UAVs . 83
5.25 Average packet delivery ratio in a tracking scenario including 10 low speed UAVs 84
5.26 Average packet delivery ratio in a tracking scenario including 10 medium speed UAVs 85

5.27 Average packet delivery ratio in a tracking scenario including 10 high speed UAVs . . . 85
5.28 Average routing traffic in a tracking scenario including 10 low speed UAVs . . . 86
5.29 Average routing traffic in a tracking scenario including 10 medium speed UAVs . . . 87
5.30 Average routing traffic in a tracking scenario including 10 high speed UAVs . . . 87
B.1 Average delay in a searching scenario of size 25 km^2 including 10 low speed UAVs . . . 108
B.2 Average delivery ratio in a searching scenario of size 25 km^2 including 10 low speed UAVs . . . 108
B.3 Average routing traffic in a searching scenario of size 25 km^2 including 10 low speed UAVs . . . 109
B.4 Average delay in a searching scenario of size 25 km^2 including 10 medium speed UAVs . . . 109
B.5 Average delivery ratio in a searching scenario of size 25 km^2 including 10 medium speed UAVs . . . 110
B.6 Average routing traffic in a searching scenario of size 25 km^2 including 10 medium speed UAVs . . . 110
B.7 Average delay in a searching scenario of size 25 km^2 including 10 high speed UAVs . . . 111
B.8 Average delivery ratio in a searching scenario of size 25 km^2 including 10 high speed UAVs . . . 111
B.9 Average routing traffic in a searching scenario of size 25 km^2 including 10 high speed UAVs . . . 112
C.1 Average delay in a searching scenario of size 25 km^2 including 30 low speed UAVs . . . 113
C.2 Average delivery ratio in a searching scenario of size 25 km^2 including 30 low speed UAVs . . . 114
C.3 Average routing traffic in a searching scenario of size 25 km^2 including 30 low speed UAVs . . . 114
C.4 Average delay in a searching scenario of size 25 km^2 including 30 medium speed UAVs . . . 115

C.5 Average delivery ratio in a searching scenario of size 25 km^2 including 30 medium speed UAVs 115
C.6 Average routing traffic in a searching scenario of size 25 km^2 including 30 medium speed UAVs 116
C.7 Average delay in a searching scenario of size 25 km^2 including 30 high speed UAVs 116
C.8 Average delivery ratio in a searching scenario of size 25 km^2 including 30 high speed UAVs 117
C.9 Average routing traffic in a searching scenario of size 25 km^2 including 30 high speed UAVs 117
D.1 Average delay in a tracking scenario including 20 low speed UAVs . . 118
D.2 Average packet delivery ratio in a tracking scenario including 20 low speed UAVs 119
D.3 Average routing traffic in a tracking scenario including 20 low speed UAVs 119
D.4 Average delay in a tracking scenario including 20 medium speed UAVs 120
D.5 Average packet delivery ratio in a tracking scenario including 20 medium speed UAVs 120
D.6 Average routing traffic in a tracking scenario including 20 medium speed UAVs 121
D.7 Average delay in a tracking scenario including 20 high speed UAVs . 121
D.8 Average packet delivery ratio in a tracking scenario including 20 high speed UAVs 122
D.9 Average routing traffic in a tracking scenario including 20 high speed UAVs 122

List of Acronyms

AANET	Aeronautical Ad-Hoc Network
AI	Artificial Intelligence
AODV	Ad-hoc On-demand Distance Vector
AWN	Aerial Wireless Network
CSAT	Cooperative Search, Acquisition, and Track
DDF	Decentralized Data Fusion
DSR	Dynamic Source Routing
EFDE	Enhanced Future Distance Estimator
FIFO	First In First Out
FSM	Finite State Machine
GeRaF	Geographic Random Forwarding
GPS	Global Positioning System
GPSR	Greedy Perimeter Stateless Routing
GRP	Geographic Routing Protocol
IEEE	Institute of Electrical and Electronics Engineer
ISR	Intelligence, Surveillance and Reconnaissance
IP	Internet Protocol
LAROD	Location Aware Routing for Opportunistic Delay-tolerant network
LLNK	Lost Link
MAC	Medium Access Control
MANET	Mobile Ad-hoc Network
MEMS	Micro Electro Mechanical System
N/A	Not Available
OH	Overhead
PDR	Packet Delivery Ratio
PHY	Physical layer
PPCP	Probabilistic Planning with Clear Preference
QoS	Quality of Service
R&D	Research and Development

RERR	Route Error
RFC	Request For Comment
RGR	Reactive-Greedy-Reactive
RREQ	Route Request
RREP	Route Reply
RWP	Random Waypoint
SCF	Store-Carry-Forward
TA	Task Assignment
TTL	Time To Live
UAANET	Unmanned Aeronautical Ad-hoc Network
UAV	Unmanned Aerial Vehicle

Chapter 1

Introduction

An Unmanned Aerial Vehicle (UAV) is defined as any aerial vehicle that does not carry a human operator. In many state-of-the-art applications, UAVs must cooperate in order to decrease mission delay and increase reliability in highly critical operations. This cooperation, which is accomplished using wireless communication, allows UAVs to share information. These UAVs can be modeled as wireless communicating nodes that form what we call an Unmanned Aeronautical Ad-Hoc Network (UAANET). In UAANETs, the relatively low number of UAVs, their high mobility and ensuing constantly changing topology challenge network connectivity. Therefore, UAANETs should have an efficient networking architecture to combat these limiting topological features. In fact, the specific features of UAANETs make it challenging to use traditional routing protocols. The main focus of this thesis is on proposing a new routing protocol for UAANETs based on the available mechanisms in the literature.

The proposed routing protocol in this thesis is a combination of a reactive routing protocol and greedy geographic forwarding. Reactive routing operates based on an on-demand route discovery that is executed before data dissemination. As mobility increases, the performance of the network degrades due to the fact that route interruptions may cause data retransmissions and errors. In the literature, mechanisms such as local repair or backtracking have been introduced to partially combat this problem [1].

The performance of geographic routing mostly relies on up-to-date location information [2]. In fact, providing reliable location information in an infrastructure-less environment (such as UAANETs) is complicated specially when the mobility increases. Before data dissemination, the source node requires to have the location information of the destination. In the literature, it is mostly assumed that the location information is accessible by deploying a separate location service module [3]. This assumption is challenging in an environment such as UAANETs, when there is no such location service. In such scenarios, data/location information dissemination can be a chicken/egg causality dilemma. In order to communicate, UAVs need to disseminate location information. Likewise, to be able to send/receive location information, a communication structure is required. One method to access location information is to use a flooding-based location service, which a source uses to obtain destination location by flooding the whole network.

1.1 Research Objective

The main goal of this thesis is to propose a new routing strategy based on reactive and geographic routing protocols available in the literature. The motivation for introducing such a design is to benefit from the advantages of the geographic and reactive mechanisms. More precisely, in this thesis, we will:

- Evaluate the performance of greedy geographic forwarding in UAANETs via a Monte Carlo simulation framework and estimate its success probability
- Propose a combinational routing protocol, called Reactive-Greedy-Reactive (RGR)
- Implement RGR protocol in a network simulator
- Compare the proposed RGR with existing reactive and geographic routing protocols

1.2 Thesis Contributions

After a definition of UAANETs, as a state-of-the-art wireless networking architecture, the focus of the thesis is to propose a combinational reactive-greedy-reactive routing. To that end, the performance of the greedy geographic core of the protocol is separately simulated in a first step and the success probability is approximated via a quadratic polynomial. This performance evaluation is helpful to give an idea on the performance of the greedy geographic core of the protocol in UAANETs. This part of the thesis was published as a conference paper as follows:

- Rostam Shirani, Marc St-Hilaire, Thomas Kunz, Yifeng Zhou, Jun Li, and Louise Lemont,"The Performance of Greedy Geographic Forwarding in Unmanned Aeronautical Ad-Hoc Networks," *9th Annual Conference on Communication Networks and Services Research Conference (CNSR 2011)*, Ottawa, Ontario, Canada, May 2011.

The next part of the thesis is to design and simulate the proposed RGR. We considered the Ad-hoc On-demand Distance Vector (AODV) routing protocol as the core reactive mechanism that is combined with a greedy geographic part. The fact that we choose AODV is based on the popularity of the protocol. However, the combinational RGR can also be implemented by using different reactive routing protocols as well. In RGR, the reactive part of the protocol sends route request to find a route to a destination. In the route reply, however, the destination not only includes the route information to the source but also sends its location information as well. After the route is established, data forwarding is accomplished similar to AODV until the time that the route is broken. In case of a broken route, the protocol will switch to its greedy geographic part. The greedy forwarding continues till a new reactive route is found or the destination is reached. Please note that route interruptions are dealt with without requiring a local repair by the greedy geographic part in RGR. In the meantime, the greedy geographic part operates without requiring an independent location service due to the use of the route request/reply mechanism originally available in AODV as the location service. Our simulation results show the improvement of RGR in terms of packet delivery ratio and end-to-end delay compared to AODV. The cost of achieving such an improvement is the higher overhead of RGR, as shown in the simulations.

1.3 Organization of the Thesis

In Chapter 2, related work on UAANET architecture is discussed and networking research is highlighted. Chapter 3 presents a performance analysis of the core greedy geographic forwarding in UAANETs where success probability of the scheme is evaluated based on Monte Carlo simulation. In Chapter 4, the details of the proposed RGR mechanism are explained followed by OPNET simulation results in Chapter 5. Finally, Chapter 6 highlights the conclusions drawn from the work and discusses future work.

Chapter 2

Related Work on UAANET

2.1 Introduction

In this chapter, we first review different aspects of aerial systems and networks. The goal is to have a clear picture of the components and networking architecture of aerial systems. Then, we review the network layer and routing protocols for ad-hoc networks applicable to aerial systems.

In recent years, there has been a growing interest in combining and sharing information through wireless channels. One important state-of-the-art application of wireless networks is Aerial Wireless Networks (AWNs). Based on the presence (or absence) of humans in an aerial vehicle, AWNs are divided into two categories: manned and unmanned. In this work, we focus on unmanned aerial vehicles (UAVs). Many applications of UAVs, such as searching and tracking, require communication among different UAV units. When several UAVs are communicating with each other via wireless links, they form an UAANET.

In the literature on Mobile Ad-hoc Networks (MANETs), researchers have proposed different categorizations for routing techniques [4, 5]. In this thesis, we categorize routing techniques in three major groups: proactive routing, reactive routing, and geographic routing protocols [6]. This simple categorization not only contains the majority of available techniques in the literature, but also provides a sufficient structure for the materials that we later discuss in the thesis.

The rest of the chapter is organized as follows. In Section 2.2, an UAANET as a state-of-the-art networking architecture is defined. In Section 2.3, cooperation, control, and path planning as the major operations of an UAANETs are discussed.

Specific networking characteristics of UAANETs are the topic of Section 2.4. Reactive and geographic routing protocols as two main components of RGR are reviewed in Section 2.5 and 2.6 respectively. Section 2.7 reviews the necessary backgrounds on MANET routing protocols and discusses the necessity of a combinational routing architecture for UAANETs. Finally, a summary of the chapter is presented in Section 2.8.

2.2 Unmanned Aeronautical Ad-hoc Networks

Before discussing about the network layer and routing protocols for UAANETs, the major components of the UAANET are introduced. To that end, this section contains a brief discussion on UAV operations and applications, followed by a qualitative comparison of AANETs and UAANETs.

2.2.1 UAV Operations

There is a wide variety of UAV shapes, sizes, configurations, and characteristics. Therefore, there exists many different classifications for UAVs. The United States army has provided a classification for UAVs based on their applicabilities and sizes [7,8]. As an example, military UAVs such as the Predator, which is able to carry missiles, are much larger than sensor UAVs which are used in studying the behaviour of a volcano. In general, the three main functions of UAV operations are: Intelligence, Surveillance and Reconnaissance (ISR) [9]. Intelligence, which improves mission efficiency [10], can be provided by many diverse sources such as human intelligence, measurement intelligence, signal intelligence, and imagery intelligence. Information from these different sources is processed by an Artificial Intelligence (AI) unit. The surveillance function is used when an application (such as border protection or enemy surveillance) requires tracking of a specific target on the ground. Finally, the reconnaissance function deals with discovery missions in which knowledge about the map of a region is not already available.

Figure 2.1 shows ISR functions in a cooperative UAV mission. Each UAV is equipped with a camera to obtain color images (the camera is a typical sensor in this example, but other kinds of sensors can also be considered). Then, the obtained image is translated into useful information using a vision processing unit. Based on the information derived from the vision processing unit, a likelihood of observation

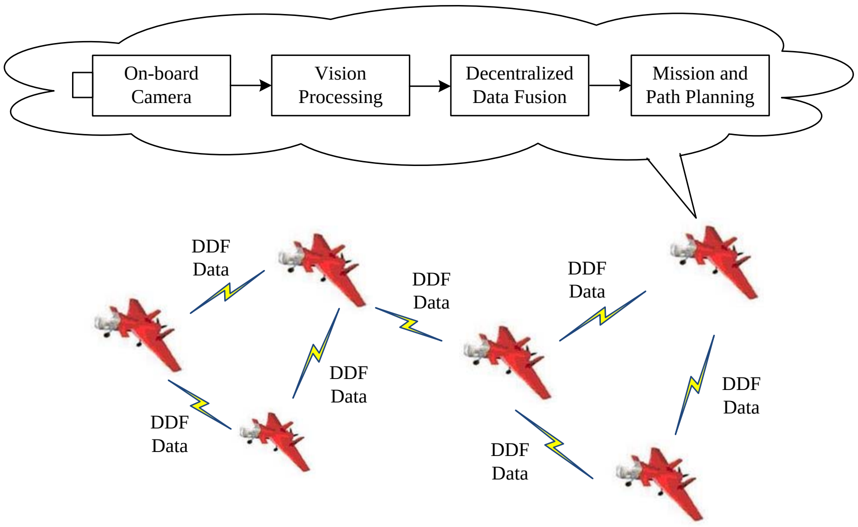

Figure 2.1: Block diagram of ISR mission

is determined and sent to a Decentralized Data Fusion (DDF) unit [9]. Figure 2.1 illustrates that DDF units of different UAVs communicate with each other, and they need to update information on a regular basis and produce state estimates. These estimates are sent to the mission control and path planning unit and are used to make decisions regarding the next trajectory of the UAV mission.

2.2.2 Applications of UAANETs

UAVs and UAANETs can be useful for several different applications. In this section, major civilian and military applications are described.

Civilian Applications

In civilian applications, UAVs are used for missions such as remote sensing, transportation, scientific research and rescue operations. Monitoring large agricultural areas, transporting goods (especially for critical missions), monitoring a natural phenomenon like volcanic eruptions or meteorological research, or even taking artistic photos from a location otherwise unreachable are all potential applications of these UAVs. For example, UAVs flew into the 2008 hurricanes in Louisiana and Texas to gather and send near-real-time data directly to the terrestrial station [11]. UAVs (especially UAVs that have low weight and small size) can also be used in search and rescue operations to find lost or trapped humans in a building after an earthquake or other natural disasters.

Military Applications

Many research groups around the world are designing suitable network architectures for military applications [12]. In the twentieth century, the main mission of air forces has been to defend against an onslaught of bombs or to attack a target. Nowadays, a collection of asymmetric fights like terrorist threats are more difficult to handle. Also, another role for the military is its peacekeeping duties, which requires surveillance and monitoring. To support this shift in military paradigm, air forces require a collection of rapidly transportable UAVs [12]. These UAVs, which are cheap and reliable, have different shapes, sizes, configurations and characteristics. A more comprehensive survey of the military perspective of UAVs can be found in [13].

2.2.3 Differences between UAANETs and AANETs

In the literature, the idea of Aeronautical Ad-Hoc Networks (AANETs) has already been proposed [14,15]. In fact, AANETs can be categorized as a pseudo-linear highly mobile ad-hoc network. Unlike commercial aircrafts in AANETs, which have transportation applications, UAVs in a UAANET are usually used for applications such as searching or tracking. These specific applications typically impose a random non-linear trajectory on UAVs (e.g. the tracked object could have unpredictable non-linear mobility). Therefore, pseudo-linearity, which results in a specific design strategy in AANETs [16], is not a feasible assumption in UAANETs. Thus, the majority of proposed AANET networking techniques are not directly applicable to more random UAANETs.

2.3 Cooperation, Control and Path Planning

The main research problem of cooperation, control and path planning is to propose algorithms for optimum path planning, to select the best trajectories, and to have minimum energy consumption. In general, the trajectories that these different UAVs are taking in a mission can be either pre-planned or dynamically planned during the mission. However, the applications of pre-planned trajectory designs are limited because, in most of the missions, unknown phenomena may affect the trajectory of one or several UAVs. As a result, considering pre-planning contradicts the ad-hoc self-organizing nature of the UAANET. For the rest of this thesis, an UAANET means a network of several UAVs in which path planning is needed to be done during the mission, unless otherwise mentioned. Having this characteristic in mind, the final goal is to establish a cooperating network to achieve a specific mission. For cooperative missions, it is necessary to have mutual communication and data transfer among UAVs. The mission is sometimes threatened by natural phenomena or enemy fighters. In this case, it is necessary for the UAVs to share critical data before being destroyed or disabled. In [17], an implementation of a reliable sensor data collection network is presented in which the UAV control is based on wireless mesh networking. The goal of this implementation is to deliver data to a monitoring station for ISR functions. Disaster management and mission critical applications can also be managed by UAVs [18]. Cooperative UAV missions use many different airborne communication technologies [12]. Design and implementation of embedded avionics using commercial of-the-shelf Micro Electro Mechanical System (MEMS) sensors and computing platforms are proposed in [19, 20]. Experimental designs are beyond the scope of this chapter. The rest of this section reviews several state-of-the-art works on cooperation, control and path planning.

Recently, there has been interest in the coverage and surveillance capabilities in the deployment of UAVs. In [21], a decentralized Cooperative Search, Acquisition, and Track (CSAT) architecture is proposed to combine searching and tracking during a mission. Searching is done during revisit times, when the UAV is not actively tracking a vehicle. Using the RAVEN [22] testbed, the results demonstrate a balance between searching and tracking, two missions that are normally conflicting [21].

Ding et al., in [23], have considered two modes of operation for UAVs: autonomous and pilot-controlled. In pilot-controlled mode, the pilot (as a separate component of a feedback loop) can change the leader UAV in a leader-follower model. The system switches to autonomous mode to do mission execution, only when no threat is perceived. Their method can be improved if the switching procedure happens more intelligently using mobility information of UAVs as a switching criterion. The idea is based on the fact that velocity and acceleration of a UAV have some embedded information. For example, when there are lots of variations in velocity, it can be concluded that an unknown event is going on, and we need to switch to pilot-controlled from autonomous mode. Fortunately, the velocity and acceleration information are available in each UAV. However, more research is required to define a criterion that can be a function of acceleration, velocity, and other mobility parameters.

Ben et al., in [24], propose a distributed algorithm for Task Assignment (TA), coordination and communication of multiple UAVs. A health-aware task assignment algorithm is developed in [25]. Vehicle health deals with fuel management, vehicle failures, sensor performance, and actuator failure modes. The idea is to propose a feedback loop that uses health state information to update the performance model in real time.

Tisdale et al. worked on a practical control strategy for a team of vehicles performing cooperative sensing [26]. They surveyed different aspects of a cooperative mission such as path planning, sensing, filtering, information measures, and cooperation. Cole et al. [27] introduced a comprehensive control structure for UAV cooperative control for information-gathering missions. It was shown that the cooperative team completed the task more than twice as fast as a team with no communications and 1.4 times faster than a team that communicated only feature-state information to make individual decisions.

In [28], Probabilistic Planning with Clear Preferences (PPCP) is proposed. In path clearance, the goal of a UAV is to traverse environments as quickly as possible without being detected by an adversary. The PPCP algorithm is scalable and efficient for scenarios in which multiple UAVs are available. The core of PPCP is an iterative algorithm that tries to find the best path starting with an initial point. The initial point is assumed to be inaccurately estimated, for example by a satellite receiver. The idea is to modify the estimation iteratively. Also, the path is changed as soon as the UAV senses a possible adversary. A disadvantage of PPCP is that the complexity of the iterative algorithm is not clearly evaluated. It should be shown that the imposed complexity does not overshadow the performance of the UAANET. In other words, the processing time of the algorithm should be in such an order that it could be helpful for a UAV that is moving with a speed in the order of hundreds of kilometres. If it takes too much time for the algorithm to process and propose a path, it probably will not be useful for the UAV anymore.

An extensive dynamic model that captures the stochastic nature of the cooperative search and task assignment problems is developed in [29]. Three different cooperative algorithms based on the stochastic nature of the cooperative search and task assignment are introduced. Although it is shown that prediction improves performance of cooperative UAVs, the prediction accuracy depends on the team size, the number of targets and the UAV's knowledge of target locations.

A decentralized control architecture for multiple UAVs, based on in-vehicle sensor data and communicated information, is designed for search, detection and localization of mobile ground targets [30]. In this design, both the overall time to search for a group of targets, and the final target localization error, are minimized.

In summary, cooperation, control and path planning are major aspects that need to be addressed in UAV missions. An important conclusion here is that cooperation and path planning can be better implemented if UAVs share information using wireless channels. As an example, consider several UAVs that are searching for an object. If they can inform other UAVs about the result of their observations, other vehicles can use this information to decide their future trajectories. Formation of UAANETs requires a specific communication infrastructure, from an accurate antenna design, to compatible network architecture and suitable layering interactions.

2.4 Networking Characteristics of UAANETs

In this section, important network characteristics of UAANETs are discussed to provide an insight into the networking requirements discussed in the next section. Some of the proposed UAANET architectures are based on MANET protocols, and the IEEE 802.11 standard [31–33]. Although generic MANET protocols could be applied to UAANETs, better solutions are obtained by considering the unique features of UAANETs. In the following subsections, the most important characteristics of UAANETs are listed to be considered in future UAANET protocol designs.

2.4.1 Rapid Topology Change

Unlike mobile nodes in a MANET scenario, UAVs can fly at speeds up to hundreds of kilometers per hour [34], and unlike AANETs [14, 16, 35] or satellite links, power is a limiting factor in UAANETs. Another difference between UAANETs and AANETs, or satellite links, is in the antenna technology that these networks can use [36, 37]. UAVs are much smaller than satellites or commercial aircrafts. UAVs need to use antennas that are not only easily transportable but also financially affordable. These factors cause the transmission range of UAVs to be much shorter than AANETs or satellite links. As a result, the ratio of a UAV's velocity to radio link is not low enough to keep the UAVs connected and prevent variations in topology. The effects of high UAV mobility are intensified when applications such as searching is considered. The faster the topology changes, the harder it is to establish relatively stable end-to-end connections.

2.4.2 Application-based Mobility

Mobility models characterize user movement patterns to simulate the network and evaluate protocol performance in MANETs. Typical mobility models for evaluation of MANETs are random walk, random waypoint, random direction, and brownian motion [38, 39]. Using random walk or any other random mobility model is inconsistent with the assumption of intelligent UAVs. However, the advantage of widely used models such as random walk or random waypoint is that they are well parameterized. In random walk, for example, there are parameters for adjusting randomness. UAV search missions have more random behaviors than tracking. To support these different randomness requirements for modeling different UAV missions, the parameters of the random walk model need to be well adjusted. Thus, it may be possible to modify original versions of MANET mobility models and make them more compatible with UAANET environments.

Effects of density variations on mobility and more generally on connectivity of the network is another crucial feature of UAANETs. Density not only depends on the number of UAVs shaping the UAANET but it will also be affected by different applications. Consider two widely-used applications of UAANETs: searching and tracking. On one hand, when UAVs are searching an area, they should be distributed in the field such that they can cover the whole area. If the region is large, the distribution of UAVs may lead to a sparse network. On the other hand, when tracking an object, a large number of UAVs follow the target. As a result, they do not need to spread out and therefore, they remain in each others' transmission range.

2.4.3 Medium Access Control Requirements

Although Medium Access Control (MAC) is not the main focus of this thesis, we briefly introduce some of the aspects of UAV MAC. A crucial task of the MAC layer is to satisfy the latency requirements of data packets that have different priorities. Admission control and differentiated services are needed at this layer to address prioritized packet scheduling in UAANETs. In most current UAANET prototypes, traditional wireless MAC protocols such as IEEE 802.11 are used [31–33,40]. In order to support priorities for different applications, different classes of Quality of Service (QoS) need to be defined. In addition, MAC protocols address not only QoS but also power saving strategies to schedule sleep modes for nodes, and assign transmission power levels.

Dynamic transmission range and power assignment can be used to manage power consumption [41]. The idea of dynamic transmission range comes from the fact that the density of UAVs is changing in time and space. Thus, it is not necessary to use the same transmission range everywhere, all the time. Power assignment is helpful when different tasks with different priorities and broadcasting requirements are assigned to different members of an UAANET. In this situation, higher transmission powers are assigned to the nodes that have accepted more duties. Other UAVs that have less responsibility are assigned lower transmission power in order to preserve energy.

2.5 Reactive Routing

After defining UAANET architecture and its networking requirements, the focus of this section is to review some of the basic literature on ad-hoc routing protocols that are used later in the thesis as the core process of UAANET routing architecture. As discussed in Section 2.1, there are three main categories of routing protocols in MANETs: pro-active, reactive, and geographic. Pro-active protocols, in which all nodes maintain up-to-date routes to all destinations have been shown to perform poorly in highly-dynamic networks [6]. As UAANETs are such networks, we exclude them from the discussion and instead focus on the more promising categories. Based on the fact that the proposed RGR protocol in this thesis is a combination of a reactive protocol and a geographic forwarding, we discuss the reactive routing in this section. Then, geographic routing protocols are reviewed in the next section.

In reactive routing protocols, a source node finds a route to destination by flooding route request packets into the network. Because the process is on-demand, the route discovery imposes some latency on the overall performance of the network. Also, flooding of route requests may cause buffer overflow and network congestion. Dynamic

Source Routing (DSR) and Ad-hoc On-demand Distance Vector (AODV) are two well-known examples of reactive routing protocols that are briefly discussed in the following paragraphs.

In DSR [42], a source node (that does not have a path to a destination in its cache) broadcasts a route discovery packet. Intermediate nodes examine their cache before forwarding the packet. When there is no route available in the cache, the node inserts its IP address in the packet and forwards the packet to the neighbors. When the destination (or any intermediate node) sends a reply packet back to the source, nodes on the selected route cache the source route. In the case of several replies from neighboring nodes, local interference may occur. To avoid this, each node delays its reply according to its distance to the source.

In AODV [1], the source broadcasts a route request to its neighbors, which in turn forwards it to their neighbors, until the destination is reached. The neighbors record the node from which the route request came. The destination chooses the route with minimum link cost and sends a route reply back along the chosen path. The nodes along the path enter the forward route entry into their routing table. The route request may take multiple paths to reach the destination, but the destination chooses the optimum path. If an intermediate node moves away, its neighbors detect the link failure and send a link failure notification to their upstream neighbors, which then forward the information until it reaches the source.

There are many proposals available on improving AODV and DSR [43]. Also, in addition to AODV and DSR, there are many other proposals available in the literature on reactive routing [44], but DSR and AODV have become Internet standards.

2.6 Geographic Routing

Geographic routing uses location information rather than network addresses to establish source-destination communication in a MANET environment. Every node in the network is aware of its own location, and location information of neighboring nodes is collected via periodic packet exchanges. Also, a source node knows the location of its destination. The source node is using these information to route the data towards the destination. For data dissemination, a node use a greedy forwarding mechanism in which a traditional geometric rule, mostly based on Pythagoras theorem, is employed. The source node sends data packets to the neighbour with minimum distance to the destination [3, 4]. In other words, greedy forwarding helps in bringing the message closer to the destination in each step by using local information. Other criteria such as minimum angle to destination can also be used to improve the performance of such approaches [45]. Greedy forwarding may encounter a problem: a dead-end (i.e. void node, local maximum, blocked node) where there is no neighbor closer to the destination. Face routing is used in recovering a dead-end. In face routing (perimeter strategy), a node that encounters a dead-end changes its forwarding strategy to a planar subgraph, using the right-hand rule until it reaches a node that is closer to the destination than the node where greedy forwarding first fails due to the local maximum [3]. Many questions may arise on constructing planar graphs for face routing in a realistic environment where the transmission range is not just a simple disc [46, 47]. Also, face routing does not work in a 3D environment, which makes the mechanism not applicable for UAANETs [48]. For the rest of this thesis mostly we consider a transmission disc, unless otherwise mentioned. For further information, a survey of geographic routing techniques in MANETs can be found in [4].

In geographic routing protocols, a node requires its neighbour's location as well as destination location information to relay data messages. Neighbour location information can be collected by geographic hello messages. For destination location information, however, the network should employ a location service module. Since, the accuracy of destination location information is an important parameter in the performance of greedy geographic forwarding, in the following subsections, we review the available location service modules in the literature. We then briefly discuss how RGR provides destination location information in the next section.

2.6.1 Location Service Module

Based on [49], location services in the literature can be classified in three major groups: flooding-based, quorum-based and home-based. A flooding-based service is the traditional one that can be proactive or reactive. In a proactive service, a node disseminates its location periodically. In a reactive service, when a node does not have the updated information of a target, a search message is flooded into the network. Location and mobility information can be used to narrow the scope of flooding.

In quorum-based approaches, the destination node sends the location updates and the source node is responsible for sending search requests. Location updates and search updates are generally sent to two different subsets of network nodes that are respectively called *update quorum* and *search quorum*. These two subsets should be selected such that their intersection is not empty. At the rendezvous points, update and search quorums can provide the location information to the querying nodes. Quorum-based approaches are more efficient with regards to overhead; however, the challenge is how to use appropriate nodes to carry search and location updates. Quorum-based approaches can be flat or hierarchical.

In the home-based approach, every node has a home region that is known to others, and location updates are proactively sent to the nodes that are in or closest to that region. Other nodes send search messages towards the home region of the destination and the message is redirected from the home region to the current location of the actuator. The concept of home in this design can be beyond the geographic location and can be a hash function of the destination ID. Home-based approaches can be further categorized as flat or hierarchical. The details of these different methods can be found in Chapter 8 of [49].

2.6.2 Location Prediction in Geographic Routing

Some mobility prediction techniques were used to improve the performance [50]. In [50], the effects of node mobility is modeled by two problems namely Lost Link (LLNK) and LOOP. The LLNK problem happens when the selected next hop (the neighbor that is closest to the destination) is not within the radio range even if it is listed as a neighbor. The LOOP problem is when a destination node moves away from its original location and another becomes a node located closest to the original coordinate of the destination. This situation is misunderstood as local maxima by traditional geographic routing such as Greedy Perimeter Stateless Routing (GPSR) [51].

Location prediction has also been proposed for QoS routing in MANETs [52]. In [52], a geometric approach is used to provide a pseudo-linear estimation of the mobility. A learning automata based approach for adaptive mobility prediction is proposed in [53]. The proposed enhanced future distance estimator (EFDE) estimates its coefficients in each step. Future distance of two nodes can be predicted for different mobility models, speeds and sampling rates.

2.7 Towards a Routing Protocol for UAANETs

In the literature, researchers mostly consider UAV swarming applications and tactical cooperative features and try to improve cooperation, control and path planning, e.g. [21]. In those types of work, the details of the communication architecture for such purposes are neglected [12]. On the other hand, another category of research exists in which the attempt is to use UAVs as communication relays among ground ad-hoc nodes [54, 55]. The main application of such relaying is specifically in battlefields in which ground-based multi-hop communication is not easy to establish.

Another class of the proposed routing protocols in UAANETs are for situations with network disconnectiviy. The idea in these types of works is to mix geographic routing with Store-Carry-Forward (SCF) procedures to be adapted to delay tolerant networks [45, 56]. A geographic routing algorithm for intermittently connected UAANETs is introduced in [57]. The routing algorithm, called LAROD (Location Aware Routing for Opportunistic Delay-tolerant networks), is a geographic beaconless routing algorithm based on the SCF principle. The UAV that holds the packet (the custodian) uses greedy packet forwarding when there are other UAVs nearby. The custodian should make sure that the packet has been received by other UAVs. If several nodes in the forwarding area receive the packet, the first expired-timer node is selected as the next forwarder to rebroadcast the packet. Overhearing the transmission by other UAVs, custody of the packet is also relinquished. The focus of this thesis is not to propose solutions in intermittently connected situations, although it can be the focus of future research.

In the current research, we specifically focus on the routing protocol for connected UAANETs. There are multiple UAVs performing a mission, they require to communicate for some tactical reason (cooperation, control, and path planning). In fact, the goal is not to optimize the mission in terms of aeronautical parameters. But, the focus is on the communication part and specifically the routing protocol for such a specific networking architecture. As UAVs in a mission require connectivity to do real-time cooperation and control, we propose a routing protocol for scenarios that the network is connected (the connectivity of aeronautical ad-hoc networks is analyzed in more details in [58] by some simplifying assumption). Based on that assumption, our goal is to combine traditional reactive routing and greedy geographic forwarding to have a compatible routing design for UAANET architecture.

2.7.1 Greedy Geographic Core in Data Forwarding

One of the main pitfalls of the proposed geographic routing protocols is that the performance of the network degrades as the network is sparser [2, 46]. The main proposed geographic mechanisms are based on the idea that greedy forwarding is used when a neighbour closer to the final destination exists [4, 49]. When the average number of neighbours is high enough, successfully establishing greedy geographic forwarding from each source to different destinations is highly probable. However, there are scenarios where the average number of neighbours is not that high. If there is no closer neighbour, a fallback mechanism such as face routing is deployed [47]. In face routing, based on the constructed Gabriel graph [49], a right hand rule is performed in which the packet is forwarded to the first right neighbour. Face routing reverts to greedy forwarding when a geographically closer node to the destination is visited.

The problems of geographic routing can be tackled from three different perspectives. First, one can explore whether it is possible to find a better method rather than blind face routing when greedy routes are not available (the term *blind* here is used for a routing mechanism that does not use the available geographic information for data forwarding) [59]. Second, we can explore how to modify available face routing mechanisms. This category includes different designs for constructing neighbourhood graphs such as Gabriel graph, relative neighbourhood graph and circular neighbourhood graph [60]. Third, we should determine how reliable greedy geographic forwarding in different scenarios is. Although the first and the second questions can be targeted for more research, the focus of Chapter 3 is on the third problem: evaluating the performance of greedy geographic forwarding in UAANETs. If greedy forwarding has low success probability, then the next steps can be on proposing new or modifying available forwarding mechanisms.

2.7.2 Reactive-Greedy Combination

One version of reactive-greedy combination has recently been proposed for wireless mesh networks [61]. Their idea is to reactively detect voids to reduce the packet loss. A node chooses a neighbor closer to the destination to forward data packets. If a node fails to forward a packet to a given destination, it will consider itself as blocked (void) for that direction. It will reactively advertise backwards a list of blocked directions so that its neighbors will not choose it as a next hop for these directions. In this method, the reactive mechanism is used only for reporting a blocked node.

In this thesis, the combination of reactive-greedy-reactive is used. As a result of RREQ/RREP, not only is a reactive route established, but the geographic location of the destination is also obtained. The data packets use at first the reactive route to forward data. In case, a route breaks, a switch to greedy geographic forwarding occurs. The novelty of the approach is the fact that the reactive-greedy combination are both used for transferring data packets. In contrast to traditional geographic routing protocols, an independent location service is not required. To the best of our knowledge, the proposed combination is the first of its kind.

2.8 Summary

In this chapter, the background literature was discussed. At first, UAANETs were defined and the specific characteristics of these networks were explained. After a brief review of the main applications of UAANETs such as coperation, control and path planning, the focus of the chapter was on specifying networking characteristics of UAANETs. Then, we elaborated the routing techniques already proposed for UAANETs or the ones that can be adapted from MANET routing protocols. A brief overview of reactive routing protocols and geographic routing protocols in the literature was also proposed. After that, we motivated the idea of combining a greedy geographic mechanism with reactive protocols. The literature on these two category of proposals were reviewed.

In the rest of the thesis, we first evaluate the performance of greedy geographic forwarding as the main bypass mechanism in Chapter 3. Then, the implementation of RGR using a network simulator is explained in Chapter 4 followed by the simulation results presented in Chapter 5. Finally, the thesis is concluded in Chapter 6.

Chapter 3

Greedy Geographic Forwarding in UAANETs

3.1 Introduction

An UAANET, as described in the previous chapter, consists of several cooperative UAVs in order to decrease mission delay and increase reliability in highly critical aerial operations [12]. The cooperation of UAVs, which is accomplished using wireless communication, allows UAVs to share information and coordinate their behaviours in order to achieve optimized performance. In this chapter, we assume that the UAVs are connected and form an ad-hoc wireless network.

In UAANETs, the relatively low number of UAVs, their high mobility and ensuing constantly changing topology cause sparse network connectivity. However, due to the fact that UAVs are equipped with positioning systems, geographic information is available in the system and can be used for data forwarding purposes as well. In other words, UAANETs can be treated as a sparse networking architecture where geographic information can be used for routing purposes [62]. Consequently, adapting available geographic routing is highly desirable in UAANETs to propose scalable packet forwarding and maintain a low latency in the routing process. Therefore, it is necessary to understand the effects of node sparsity in UAANETs on the performance of geographic forwarding mechanism.

In this chapter, the goal is to evaluate the performance of greedy geographic forwarding in UAANETs. To that end, Section 3.2 covers the related work on the performance of geographic forwarding. Section 3.3 describes simulation environment followed by the simulation results in Section 3.4. An observation of the simulation results is presented in Section 3.5. Based on the simulation results, we then propose a quadratic estimation of success probability in Section 3.6. Finally, Section 3.7 provides a summary of the chapter.

3.2 Background

Geographical routing was introduced to provide scalable routing in mobile ad-hoc networks by exploiting location information of mobile nodes in the network [4]. Many of the current literature on geographic routing protocols have borrowed ideas from deterministic graph theory to analyse the protocols. The most commonly used models are: Gabriel graph, relative neighborhood graph, and circular neighbourhood graph [60]. Another area of research in geographic routing is the level of local knowledge needed to accomplish different goals such as spectral efficiency or energy efficiency. The problem of energy efficiency has been formulated in [63]. The proposed schemes are shown to provide near optimal energy consumption while outperforming former greedy forwarding mechanisms. Path pruning is another topic of research on geographic routing protocols [59, 64]. In path pruning, the idea is to provide a more efficient method rather than blind face routing.

In [65], a statistical method is introduced for analysing the previously proposed geographic random forwarding (GeRaF) [66]. In [65], wireless nodes are distributed in the region via a Poisson distribution and therefore the remaining distance to the destination has an exponential distribution. A multi-hop mechanism is proposed that ensures the best node (geographically closest to the destination) is chosen. In this chapter, the results are not just limited to a specific forwarding mechanism such as GeRaF. In fact, the aim of this chapter is to analyse a wide range of available greedy geographic routing protocols and evaluate the effects of different parameters such as density and transmission range. Since the simulation scenarios are not limited to specific cases, the results are more general and apply to all greedy geographic forwarding mechanisms.

A statistical evaluation of the properties of greedy geographic routing in spontaneous wireless mesh networks has been presented in [67]. In the proposed architecture, nodes are distributed in the scenario under a Poisson distribution assumption and, similar to [65], they evaluated the amount of progress in each step. The Poisson assumption simplifies analytical formulation; however, the results might not be that realistic. Different from [67] and [65], this chapter considers uniformly distributed nodes, which is more realistic for a UAANET scenario.

Although geographic routing has been the target of research from different perspectives, most of those studies evaluate the performance of geographic mechanisms in dense situations. In this work, we consider different densities from very sparse to sparse and relatively dense to evaluate the effects of changing the number of nodes, the average number of neighbours and the transmission range on the packet success probability. To that end, a completely random sparsely connected UAANET is considered. The focus of the study is on the performance of snapshots of the network in such a scenario. These snapshots then can also be generalized to all other states of the network, addressing the protocol performance in the presence of mobility.

3.3 Simulation Environment

We used Monte Carlo simulations to evaluate the performance of greedy geographic forwarding in UAANETs. In the simulations, one snapshot of the network is assumed containing a source node S and a destination node D. The environment is a square with length $L = 100$. Also, a fixed transmission range of T_r for all the nodes in the network is considered. It is important to note that the size of the network and transmission range are unit-less. In fact, this unit-less environment does not affect the generality of the proposed results, because the ratio of Tr/L is important, both of which are considered unit-less.

In each simulation, N nodes are generated. These nodes are uniformly distributed in the environment. Based on the distance between each pair of nodes, the assigned transmission range for individual nodes in the network is selected. To that end, the M shortest links are considered. In order to find the M shortest links in the simulations, a distance matrix is defined. $D(i,j)$ is the Euclidean distance between node i and j, which is also the $(i,j)th$ element of the distance matrix. Then, all of the distances from each node i to all other nodes are sorted in ascending order in an array called the distance vector. In the next step, the first M entries of the vector are selected as the M shortest links in the network. Since each link is connecting 2 nodes and there are N nodes in the network, the corresponding average number of neighbours per node (UAV) is $2M/N$.

The aim of the simulations is to evaluate the performance of the greedy part

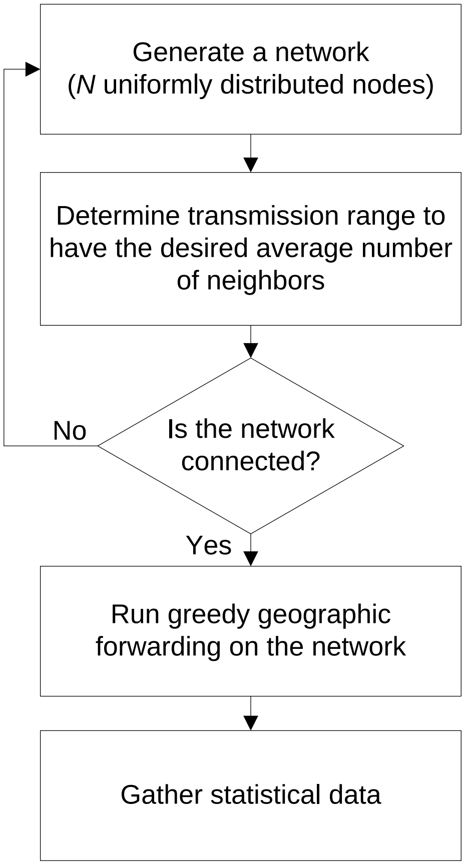
Generate a network
(*N* uniformly distributed nodes)
Determine transmission range to have the desired average number of neighbors
Is the network connected?
No
Yes
Run greedy geographic forwarding on the network
Gather statistical data

Figure 3.1: The algorithm for gathering greedy geographic forwarding statistics

of the geographic routing mechanism with respect to the number of nodes in the network. After generating a network with the desired average number of neighbours per user, we make sure that the network is connected. The goal in this section is to test the performance of greedy geographic forwarding on connected networks (i.e. disconnected networks will be dropped). As soon as a connected network is found, then the greedy geographic routing is run on it. After that, we compute the percentage of successful source-destination communications that can be established. A modular representation of the algorithm is depicted in Figure 3.1.

The algorithm depicted in Figure 3.1 generates connected networks with a constant average number of neighbours and evaluate the success probability of greedy geographic forwarding. We also conducted another simulation to evaluate the effects of changing the transmission range. In this case, the target parameter is the transmission range and therefore the second block of Figure 3.1, which is used for determining transmission range, will not be needed and the algorithm will be slightly different.

3.4 Simulation Results

In order to gather the relevant statistics, 10 different networks are considered and the results presented here are the averages over those 10 repetitions. The value of the average number of neighbours is initially equal to 2, which is close to the minimal value of the average number of neighbours that is required for a connected network. Therefore, the simulation time for this value is high, as several networks need to be generated in order to find one where all the nodes are connected. In the first stage, we keep the average number of neighbours equal to 2 in order to generate extremely sparse yet connected networks. However, in additional experiments, we also increase the average number of neighbours to 3, 4 and 5 in order to evaluate the network performance in denser scenarios.

For an average number of neighbours equal to 2, 3, 4 and 5, simulation results for networks of size $N = 5, 10, 15, 20, 25, 30$ are collected. In Figure 3.2, the results of increasing the number of nodes on the success rate of mutual source-destination communications are depicted. For "$N = 5$, average number of neighbors = 2", the average percentage of correct receptions exceeds 92% and decreases to less than 55% when the number of nodes is increased to $N = 20$. For "$N = 5$, average number of neighbors = 4", the average percentage of correct receptions is 100% (all nodes are direct neighbours of each other). That value decreases to less than 85% for "$N = 30$, average number of neighbors = 5".

It is worth mentioning that for a constant number of neighbours, the transmission range adapts accordingly. Intuitively, by increasing the number of nodes, the transmission range should be shrunk to maintain a constant average number of neighbours. In such a scenario, where the average number of neighbours is kept constant, increasing the number of nodes has significant effects on the percentage of successful communication pairs. The intuition behind this reduction of successful communication pairs is that when the average number of nodes increases, and therefore the transmission range decreases, many source-destination pairs will be connected by longer paths in the network. A longer path means that there are more nodes on the path from a random source to a random destination. If one of the nodes is not able to find a neighbour closer to the destination, the greedy forwarding cannot establish a

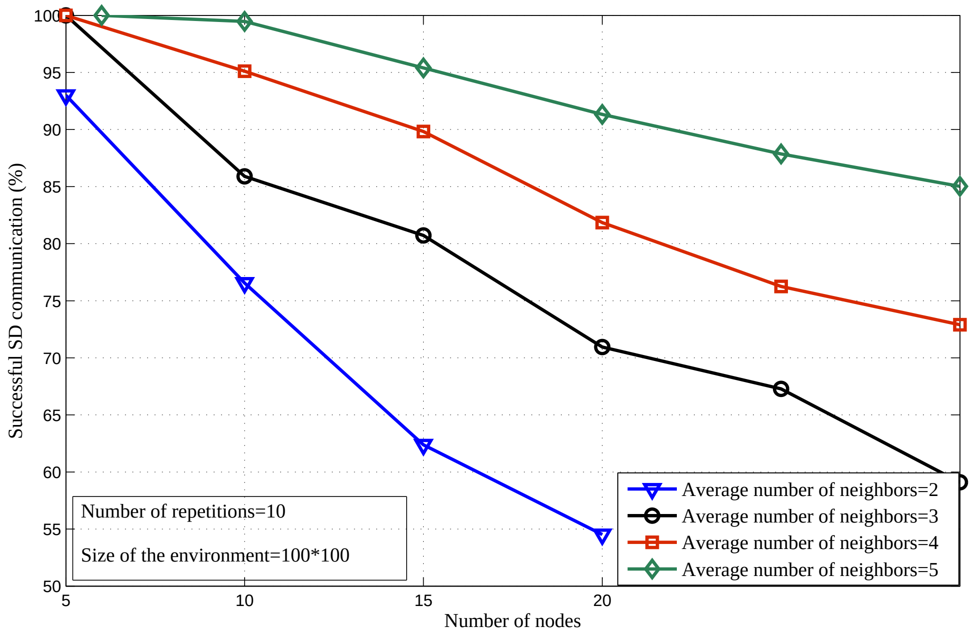

Figure 3.2: Effects of increasing the number of nodes on successful source-destination communications

path from source to destination. Consider P_i is the probability that the $(i-1)th$ node can find a neighbour that is closer to the destination. Assume that there are k nodes between S to D in a greedy path. When we increase the number of nodes and keep the average number of neighbours a constant, the average number of hops that needs to be traversed from source to destination increases accordingly. Based on the fact that $P_i \leq 1$, the product of P_i for all hops decreases as the number of terms in the product increases. Therefore, the higher the number of hops (k), the lower is the probability of establishing a path from S to D.

In a second simulation, we analyzed the effects of increasing the average number of neighbors on the percentage of correct source-destination communications. The result is depicted in Figure 3.3. In this simulation, we increase the average number of neighbours from 2 to 10 for a scenario in which (10, 15, 20, 25) nodes are uniformly distributed in a 100×100 environment. For a network of ten nodes, the percentage of successful source-destination pairs increases from almost 76% to 100% when average number of neighbours are increased from 2 to 9 (a very sparse scenario to a very dense one, where every node is connected to every other node). For larger networks, we see

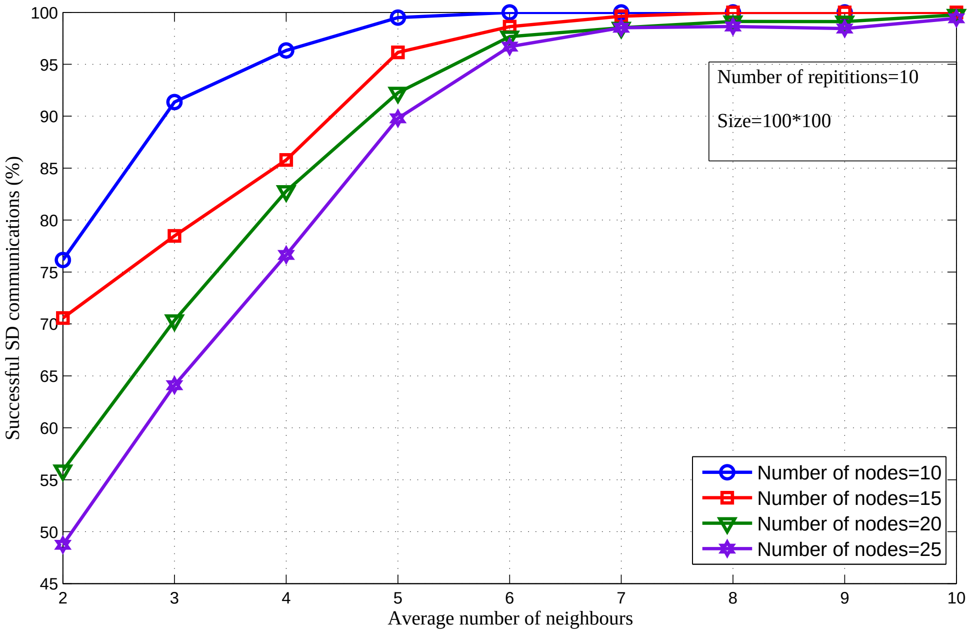

Figure 3.3: Effects of increasing the average number of neighbors on successful source-destination communications

a similar increase with an increase in network density.

In Figure 3.2 and 3.3, the parameters that we worked on were the number of nodes and the average number of neighbours respectively. The effects of increasing transmission range on the successful communication pairs are not clearly shown in the previous figures. As a result, Figure 3.4 shows the percentage of successful communication pairs for different transmission ranges. For a similar squared environment of size 100 × 100, the number of nodes is changed from 10 to 30. For each of these values, the transmission range is changed from 20 to 50 (1/5 and 1/2 of the length of the environment respectively) in steps of 5. Even a ratio of 1/5 for transmission range over length of the region can be considered to be quite a high value; however, even for this value, more than 35% of the source nodes cannot reach their destination nodes via greedy forwarding in a network of 30 nodes. Another observation from Figure 3.4 is that curves for different values of number of nodes intersect each other. The probability of success for a larger number of nodes using small transmission ranges is lower than the probability of success for a smaller number of nodes using small transmission

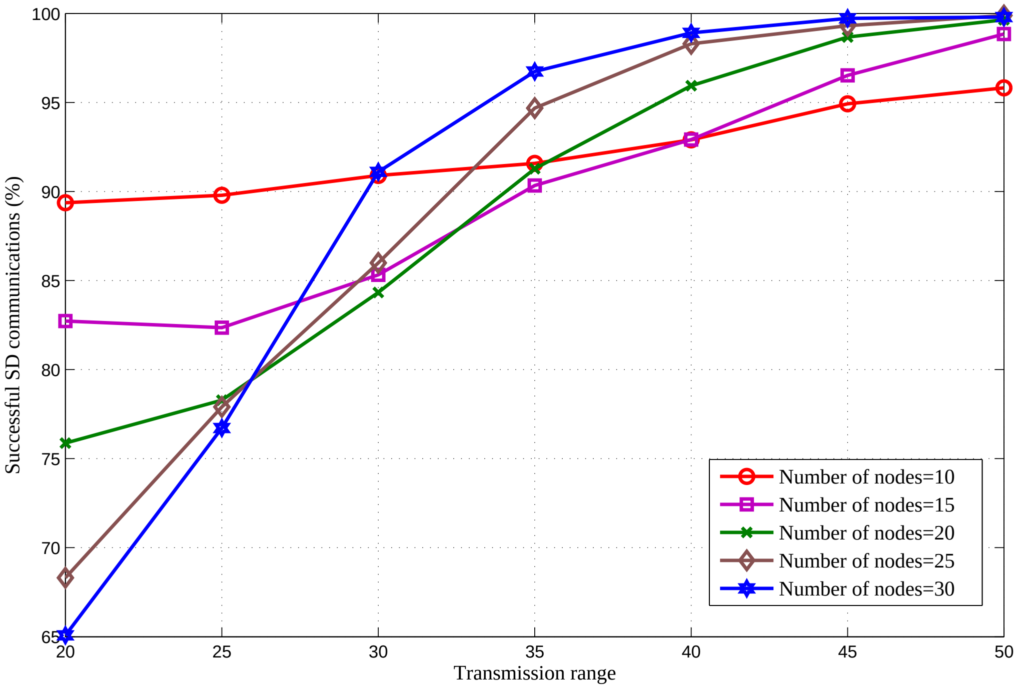

Figure 3.4: Effects of transmission range on successful source-destination communications

ranges. However, for large transmission ranges, the probability of success for a large number of nodes is greater than that for a small number of nodes. This observation is intuitively true, because when there are a lot of nodes in the network, small transmission ranges have more negative effects due to the existence of longer paths in the network. As the transmission range increases, in denser scenarios (i.e. larger number of nodes), the situation will reverse, with many source-destination pairs now connected through paths only one or a few hops long, increasing the chance of greedy geographic forwarding being successful. Therefore, at a point, the curve of larger number of nodes intersects the curve of smaller number of nodes and tends to 100% sooner.

3.5 Observations From Simulation Results

The main observations from the simulation results can be summarized as follows:

1. While keeping the average number of neighbours fixed, increasing the number of nodes reduces successful end-to-end communications greatly. This results from the accompanying reduction in transmission range, which causes more source-destination pairs to be connected over paths with an increased number of hops.

2. Geographical routing (that is a combination of greedy forwarding and face routing) will encounter a lot of switches from greedy mechanism to face routing and vice versa when the network is sparse. This may impose inefficiencies in network performance.

3. Simulation results show that for networks with an average number of 2 neighbours (i.e. a very sparse network), and a network with 20 nodes, successful source-destination communications can be established via greedy geographic forwarding only in less than 60% of all cases.

4. For 20 nodes, when the average number of neighbors is equal to (or more than) 6, the percentage of successful communication pairs are close to 100%. For an average number of neighbors from 2 to 5, this percentage will range from less than 60% to 90%.

5. For 20 nodes, a transmission range of 40 (which is quite a large value relative to our simulated scenario area) or more is needed to ensure that the greedy geographic forwarding success rate is close to 100%. A transmission range of 20 only has a success rate of approximately 75%.

3.6 Quadratic Estimation of Success Probability

In this section, based on the simulation data, a quadratic function for computing the probability of success is proposed. Denote $P_{success}$ as the success probability of greedy geographic forwarding and h as the number of hops. We use quadratic polynomials to fit the simulation data. The coefficients of the fitting quadratic polynomials are determined by minimizing the sum of the squared errors between the function and the Monte Carlo simulation data. Goodness of fit of each of the estimated functions, which has a value in the interval [0,1], is also calculated based on the definition in [68].

When the number of nodes is 10, the greedy forwarding success probability can be fitted to the quadratic function shown in (3.1) and plotted in the top curve of Figure 3.5. All curves in Figure 3.5 were generated by averaging the probability of success for 4 different simulations in which the average number of neighbours was varied from 2 to 5.

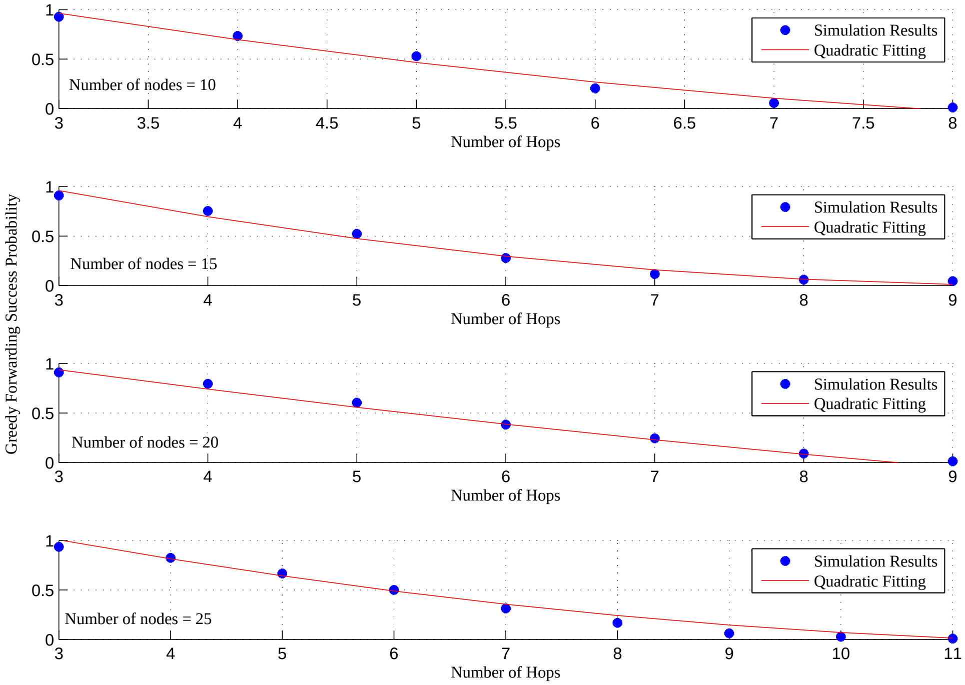

Figure 3.5: Greedy geographic forwarding success probability (for number of UAVs=10, 15, 20, and 25)

$$P_{success} \approx 0.0172h^2 - 0.387h + 1.97 \qquad 3 \le h \le 7 \tag{3.1}$$

The goodness of fit for Equation (3.1) is 0.9924. In order to have a wider observation of the greedy geographic success probability for different number of hops, the following approximation function can be used:

$$P_{success} \approx \begin{cases} 1 & h < 3 \\ 0.0172h^2 - 0.387h + 1.97 & 3 \leq h \leq 7 \\ 0 & h > 7 \end{cases} \tag{3.2}$$

In Equation (3.2), please note that for number of hops equal to 1 or 2, the probability of success is 1 due to the fact that in a connected network, greedy geographic forwarding is always successful for one and two hop neighbours.

The polynomial estimation for the success probability of greedy geographic forwarding for number of nodes equal to 15 is shown in (3.3). The goodness of fit for this curve is 0.9947. Monte Carlo simulation results and equivalent fitting function are depicted in the second curve of Figure 3.5.

$$P_{success} \approx 0.021h^2 - 0.41h + 2 \qquad 3 \leq h \leq 9 \tag{3.3}$$

Equivalently, we can come up with the following piecewise function:

$$P_{success} \approx \begin{cases} 1 & h < 3 \\ 0.021h^2 - 0.41h + 2 & 3 \leq h \leq 9 \\ 0 & h > 9 \end{cases} \tag{3.4}$$

The greedy geographic success probability for a network of 20 UAVs can be estimated by the quadratic polynomial shown in (3.5). The goodness of fit for the curve is also 0.9972. Monte Carlo simulation results and values of the quadratic fitting function are plotted in the third curve of Figure 3.5.

$$P_{success} \approx 0.0063h^2 - 0.24h + 1.6 \qquad 3 \leq h \leq 8 \tag{3.5}$$

Similarly, (3.5) can be expanded to include the number of hops from 3 to 8 as:

$$P_{success} \approx \begin{cases} 1 & h < 3 \\ 0.0063h^2 - 0.24h + 1.6 & 3 \leq h \leq 8 \\ 0 & h > 8 \end{cases} \tag{3.6}$$

Finally, for a UAANET containing 25 UAVs, the quadratic fitting function, with a goodness of 0.996, is derived as

$$P_{success} \approx 0.0097h^2 - 0.26h + 1.7 \qquad 3 \leq h \leq 11 \tag{3.7}$$

The Monte Carlo simulation results and values computed using (3.7) are plotted in the bottom curve of Figure 3.5. Similarly, for an arbitrary h, we have the following piecewise approximation:

$$P_{success} \approx \begin{cases} 1 & h < 3 \\ 0.0097h^2 - 0.26h + 1.7 & 3 \leq h \leq 11 \\ 0 & h > 11 \end{cases} \tag{3.8}$$

In order to have a good understanding of the behaviour of the success probability of greedy geographic forwarding, we compute the overall probability of success versus the number of hops and average them over the number of hops.

$$P_{success} \approx 0.012h^2 - 0.29h + 1.7 \qquad 3 \leq h \leq 10 \tag{3.9}$$

Equation (3.9) is the estimated function for the overall value for the probability of success versus the average the number of hops. The goodness of fit for this function is 0.9958. In (3.10), the approximated $P_{success}$ is shown, and its values versus the number of hops are plotted in Figure 3.6.

$$P_{success} \approx \begin{cases} 1 & h < 3 \\ 0.012h^2 - 0.29h + 1.7 & 3 \leq h \leq 10 \\ 0 & h > 10 \end{cases} \tag{3.10}$$

It is worth-noting that $P_{success}$ in (3.10) is estimated for number of hops from 3 to 10 for UAANETs that contain either 10, 15, 20, or 25 UAVs. These are typical values that are used in UAV missions that require a team of UAVs.

As indicated in Equation 3.10, the success probability of greedy geographic forwarding is 1 for the one and two hop neighbours. For $h \geq 3$, it is shown that the constant term of the quadratic equation is larger than the other coefficients. In other

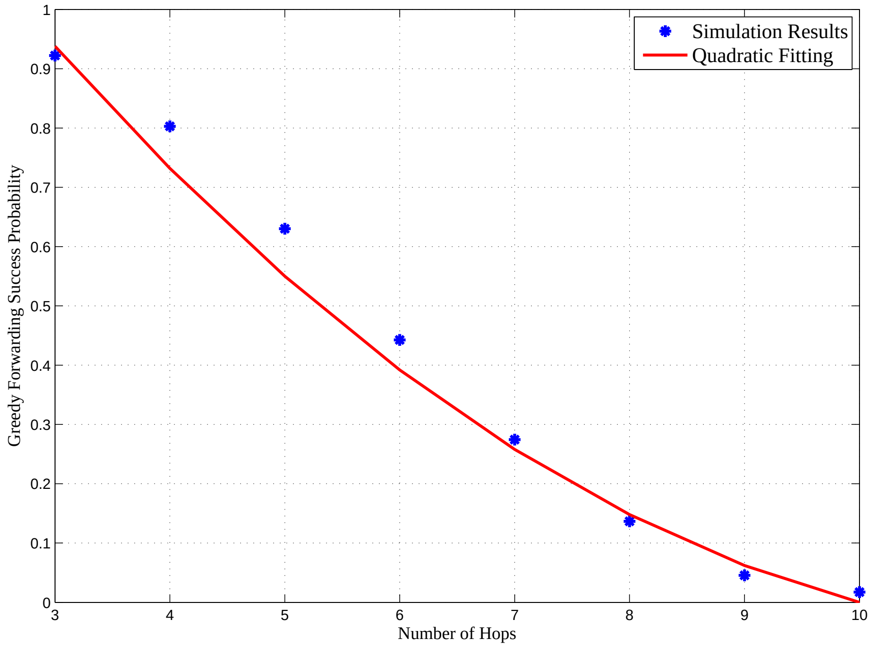

Figure 3.6: Average success probability of greedy geographic forwarding

words, the constant term is dominant in shorter hop routes. As h increases, other terms play a more significant role. This is also shown in Figure 3.6, where the success probability degrades to less than 50% for 6-hop neighbours. One idea that arise here is to use the greedy geographic forwarding in shorter routes while the destination is not more than 2-3 hops away. In such a case, the success probability of greedy geographic forwarding is either 1 or at least more than 0.9. The rest of this thesis is based on the described intuition. We want to combine greedy geographic forwarding with other available literature such that we use greedy geographic forwarding when the number of hops to the destination is likely to be smaller.

3.7 Summary

Based on the fact that each UAV is aware of its location, a geographic-based routing can be used to provide routing in UAANETs. In this chapter, a simulation framework for studying greedy geographic forwarding in UAANETs was proposed. Simulation results illustrate that using only greedy geographic forwarding in sparse situations is not 100% sufficient and a combination of other methods is required. For less critical applications of UAANETs, greedy geographic routing can be used. But, for applications that require higher packet delivery ratios, other mechanisms must be combined with greedy geographic forwarding. One possible avenue would be to add a void handling technique, such as face routing. But, applying face routing for a naturally 3D network such as UAANET is challenging in reality. The other disadvantages of a greedy-face-greedy combination is the necessity of an independent location service, which requires another communication architecture rather than the greedy geographic forwarding and face routing.

Determining a destination's location, in the absence of a separate location service, could be done in an on-demand fashion, similar to the route discovery in reactive protocols. As a matter of fact, that is exactly the approach to explore in the rest of the thesis. The reactive route discovery can be extended to learn about the destination location. Then, greedy geographic forwarding, which works well over shorter hops (see Figure 3.6) can be exploited as the way to recover from reactive route failures.

Chapter 4

Reactive-Greedy-Reactive Routing for UAANETs

4.1 Introduction

Traditional routing protocols in ad-hoc networks may have some difficulties in handling communication in UAANETs. The performance of different ad-hoc routing mechanisms, such as geographic and reactive protocols, degrades as the mobility in the network increases [50]. In order to deal with unwanted route interruptions in reactive protocols in high mobility (especially when there are a considerable amount of traffic), reactive mechanisms sometimes provide detouring algorithms or local repairs at intermediate nodes [1]. In geographic routing, as mobility increases, the number of hello messages for exchanging a neighbour's location information increases to keep up-to-date location information available for the communication session. In fact, the performance of greedy geographic forwarding in UAANETs has been simulated in Chapter 3 and the results show that the performance degrades in sparse scenarios (when average number of neighbours is equal to 2, almost half of the packets are dropped).

One idea to make traditional routing mechanisms more applicable for UAANETs is to combine two or several of them in order to exploit the benefits of different individual schemes. In this work, the idea is to combine a reactive routing protocol with the greedy geographic scheme. The proposed RGR mechanism is a combinational design in which the reactive and the greedy geographic parts may act as each other's complement. The goal is to design the RGR mechanism such that it improves end-to-end data packet delivery, especially for scenarios in which neither the reactive scheme nor the greedy geographic mechanism performs well.

The reactive part of the RGR is based on the AODV routing protocol. In RGR, during the process of route request/route reply (RREQ/RREP), a route is established on which the destination location information is also available in the source node. The source node sends the data towards the destination, similar to AODV. In the event of a link breakage (a case that is more likely in UAANETs based on the fact that the velocity of UAVs can be relatively high in some applications), the reactive mechanism cannot continue to transfer data until the route has been repaired, either locally or globally. The RGR mechanism, in such a case, switches to greedy geographic forwarding towards the destination. In the next hop, the process uses the reactive mechanism to see if there is a route to the destination. If such a route exists, the data is forwarded on the route. Otherwise, the greedy geographic forwarding continues its operation. The same process continues to deliver the packet to the destination. In RGR, a packet can possibly be dropped if there is neither a reactive route to the destination nor a geographically closer neighbour in the table of the current node towards the destination.

The adaptive nature of the RGR protocol makes it compatible for different applications. In case where the UAV trajectories have low relative speeds and they remain in each others' vicinity for a while, the process of message forwarding mostly relies on the reactive part. On the other hand, if there are many interruptions due to high relative velocity of UAVs, more reactive routes will be broken. Therefore, the mechanism switches more often to greedy geographic forwarding and the data dissemination will be based on the geographic location of the nodes rather than the route information. The adaptive property of the scheme makes it robust and compatible to different scenarios for UAANET missions.

The main intuition for using greedy geographic forwarding as an alternative to the reactive path is that it is likely that, while the reactive route breaks, there is a geographic neighbour that can be used for data forwarding when the UAVs are kept connected. As a result of the relatively high mobility of nodes in a UAANET scenario, a node on the source-destination path may move away, which causes a broken route. However, it is also likely to have another node close to the current intermediate node which can be used as a geographic forwarder towards the destination in a connected cluster.

The rest of this chapter is organized as follows. Section 4.2 discusses different possible combinational designs and motivates the idea of combining a reactive routing protocol with the greedy geographic forwarding. In Section 4.3, an overview of the OPNET implementation of AODV and the proposed RGR is presented. In Section 4.4, the details of RGR's functionality based on the AODV core and the additional greedy scheme are explained. In Section 4.5, RGR is qualitatively compared with reactive and geographic routing as presented in the literature. Finally in Section 4.6, the chapter is summarized.

4.2 Motivation

The main idea for proposing the combinational RGR is for scenarios in which every node is aware of its location, e.g. via a global positioning system (GPS). In UAANETs, every UAV is equipped with GPS for the mission. Therefore, the location information is available at no extra cost in every UAV. In order to use the available location information, the reactive protocol is combined with greedy geographic forwarding to improve the end-to-end packet delivery. One question that may arise here is why do we propose this reactive-geographic combination when there are also other alternatives available in the literature. To answer this question, we discuss some other alternatives for combining different routing/forwarding protocols in the literature. We then compare RGR with other possible proposals.

4.2.1 Blind Broadcasting

The combination of broadcasting and geographic routing is the simplest alternative that one may consider. Each node broadcasts its location periodically and all other nodes have a table in which they update the location information of different nodes (i.e. UAVs) that can be a potential destination. The update process is done while the communicating data sessions are running in different parts of the network.

4.2.2 Geographic Reverse-route Forwarding

In geographic reverse-route forwarding, the source broadcasts the location request. Intermediate nodes rebroadcast the request until it reaches to the destination and then, the destination node sends the location information back to the source geographically, based on the fact that the source and destination have included their corresponding location information in the packet. The source also uses the geographic location of the destination for the next packet to geographically forward it towards the destination.

4.2.3 Reactive Reverse-route Forwarding

A RREQ/RREP procedure is used in reactive reverse-route forwarding to obtain the location information of the destination. The data forwarding is then based on the geographic mechanism. In the meantime, the reactive route is kept as long as fresh location information of the destination is required. While the data is sent via the geographic part of the protocol, the reactive part is used for transmitting control messages containing location information. The idea is to send location information and control messages on a more reliable path (i.e. the reactive path).

4.2.4 Proactive Reverse-route Forwarding

Unlike reactive reverse-route forwarding, the established route is not based upon the request in proactive reverse-route forwarding. The routes are available before the geographic data forwarding starts. In other words, all potential sources establish routes to potential destinations before the actual communications start. The difference here is the online availability of the location information, which is helpful for time-critical applications.

4.2.5 Discussion on the Proposals

In this section, a comparison of these different proposals is presented and then the motivation for developing RGR is discussed. The blind broadcasting scheme is the simplest proposal that can be used. The main disadvantage of that scheme is that a lot of unnecessary packet broadcasting may happen in the network. However, due to the non-geographic nature of the scheme, updating neighbour location information by hello messages is not required.

Geographic reverse-route forwarding, reactive reverse-route forwarding, and proactive reverse-route forwarding are proposed to limit the number of broadcast packets in the network. The source node first broadcasts location requests. The common part of all of these proposals is that the destination uses a unicast mechanism towards the source after receiving the location request.

Reactive and geographic reverse-route forwarding mechanisms are more scalable than proactive reverse-route forwarding due to the fact that in a proactive mechanism, the source-destination route is maintained even though the route is not used. However, the location information in proactive reverse-route forwarding is available without delay once a source starts transmitting data to a destination. The trade-off here is between online availability of location information and maintenance cost of the routes.

The independence of location service mechanism and data forwarding is a characteristic of reactive and proactive reverse-route forwarding mechanisms compared to geographic reverse-route forwarding in which the data and the control packets are both sent geographically. One disadvantage of geographic reverse-route forwarding is the possibility of delay imposed by alternative mechanisms in the literature such as face routing or backtracking [61, 69].

The disadvantage of reactive and proactive reverse-route forwarding mechanisms is that we pay the cost of establishing a source-destination route, but we do not use it for data forwarding. In order to compensate for this disadvantage, the idea in RGR is to use the established route not only for control packets but also for data forwarding. In contrast to reverse route forwarding methods, here the established route is used as much as possible. As soon as a failure (i.e. route interruption) occurs, the forwarding mechanism switches to greedy geographic forwarding until another reactive route is found or the destination is reached. In the next sections, the details of the RGR protocol are explained.

4.3 Implementation of RGR in a Network Simulator

The proposed RGR protocol was implemented in OPNET Modeler [70]. OPNET Modeler is designed to accelerate the R&D process for analyzing and designing communication networks, devices, protocols, and applications. The proposed modular access to different network components makes it possible to design a protocol independent of other modules in the network. The other motivation for using OPNET in this research was that AODV has already been implemented there. Besides AODV, a Geographic Routing Protocol (GRP) is also available in OPNET, which can be used for comparison purposes. Due to the fact that the reactive core of the proposed RGR is based on AODV, an overview of AODV and its implementation in OPNET are first presented in this section before going into the details of the RGR implementation.

4.3.1 Overview of AODV in OPNET

The overview of AODV in this section is only targeting key features of the protocol. The details of AODV can be found in RFC 3561 [1]. The core of AODV routing is based on two mechanisms: route discovery and route maintenance. The route discovered, as a result of the RREQ/RREP phase, is used by the source to send data to the destination. In the meantime, a request table is used to keep track of RREQ messages generated or forwarded by the current node. The request table is used to discard duplicate RREQ messages. In OPNET, the packet arrival function handles the arrival of a packet by deploying several sub-functions, which individually handle the arrival of application packets, RREQs, RREPs, and route errors (RERRs).

In order to handle a packet arrival, the routing table is checked to determine if there is a route to the destination. If there is no such route, a RREQ is generated and sent via several available sub-functions. The RREQ contains source and destination IP addresses, which are stored in the source and destination address fields of the AODV route request message header respectively. The previous node's IP address is also saved in the RREQ. The routing table of AODV contains destination IP address, destination sequence number, and next hop address. The routing table is updated by a RREQ only if fresher information is available in the current message. The freshness of a RREQ is determined by a monotonically increasing sequence number maintained by the nodes. Upon reception of a RREP, the node updates its table if the sequence number in the RREP is greater. In case the sequence numbers are equal and the new hop count is smaller, then the routing table is updated as well.

AODV employs an expanding ring search technique to prevent unnecessary network wide dissemination of RREQs. In expanding ring search, the TTL field of the IP header of the RREQ is set to a certain value. If the route discovery process fails to find a path to the destination, the source increases the value of the TTL field and repeats the process. The process continues until either the source finds a path or the whole network has been searched and no path has been found.

Local repair is another feature of the available AODV implementation in OPNET. In the event of a link breakage in AODV, a local repair mechanism may be deployed. In local repair, the node upstream of the broken link broadcasts a RREQ to find the destination. In AODV, local repair is an option and intermediate nodes locally decide whether or not to use that (using that option is not always beneficial in some scenarios). Further discussion on the functionality of local repair can be found in [1].

C structures are used for AODV (and also for other MANET routing protocols) in OPNET to define message headers. As an example, the structure of RREQ and RREP messages are defined in **AodvT_Rreq** and **AodvT_Rrep** headers in an external C file. For routing and request tables, also different data structures are used. The AODV routing table is implemented as a hash table indexed by an IP address.

4.3.2 RGR Implementation Overview

We have implemented RGR in OPNET, based on the AODV process model available in OPNET. Node model, process model, and proto-C codes of AODV model should be changed to implement RGR. Designing the right finite state machine (FSM) model and changing/adding the necessary C code corresponding to each of the processes in the FSM to be able to support RGR are the key points in the OPNET implementation.

After designing the RGR process model based on the available AODV process, the structure should be attached to IP. The IP process model (**ip_dispatch**) is a part of every MANET routing protocol in OPNET. The **ip_dispatch** identifies and invokes the routing protocol that is configured in the network layer. MANET routing protocols are using **ip_dispatch** via **manet_mgr**, which is designed to parse attribute values to identify and configure the desired routing protocol.

An important goal of this work is to see the performance of the protocol for different trajectories. As an example for UAANETs, tracking and searching applications can be used. The first step in this part is to design such trajectories that can realistically model UAV trajectories in a searching or tracking application. The trajectory design for UAVs in OPNET is accomplished via the random waypoint (RWP) model. Using the RWP is useful due to the fact that is a standard mobility model that can be modified to represent both searching and tracking missions of UAANETs.

After implementing the RGR protocol and required UAV mobility models, we compare RGR with traditional routing protocols in ad-hoc networks. For such a purpose, AODV and greedy geographic routing protocols are considered for comparison in different scenarios.

4.4 RGR Functionality

In this part, a high-level overview of RGR functionality is provided relevant to the proposed OPNET implementation. In the following subsections, different design alternatives of RGR are discussed and the OPNET process model is briefly reviewed. The assumption is that the procedure of AODV and greedy geographic forwarding has been explained in previous sections, and the reader has a general knowledge of the functionality of these schemes. The focus of this section is then more on the parts which are added to the original AODV to change it to RGR. Therefore, the reactive-geographic interactions are discussed and the process taken after switching to greedy geographic forwarding is explained in more detail.

4.4.1 Elements of RGR Mechanism

In this subsection, we briefly introduce the main elements of RGR. In the next sections, the RGR mechanism is discussed in more detail. Figure 4.1 represents a scenario in which a source node (S) is going to communicate with a destination node (D). There are 6 other nodes in the scenario which are used to implement a source-destination route.

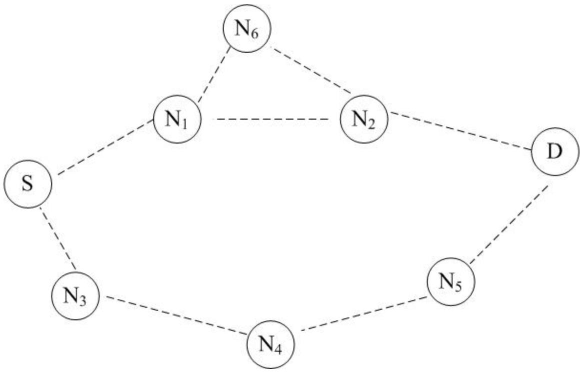

Figure 4.1: The scenario to establish a reactive route from S to D

In the reactive part, the source checks if there is a route available in its routing table. In that case, data is sent over the route (this step is similar to AODV, therefore the details are skipped). If there is no such route, the RREQ/RREP mechanism is used. As a result of the RREQ/RREP exchange, location/route information is fetched back to the source. The source node then forwards the data packet towards the destination on the route similar to the AODV process. The intermediate nodes also receive the packet and forward it to the destination.

In the second part, assume that a packet is received by intermediate node (N_1) as shown in Figure 4.1. First, the node checks if there is a reactive route to the destination. If such a route does not exist (due to the mobility of the neighbouring nodes), then the packet is forwarded geographically. The following algorithm presents the greedy geographic forwarding deployed in the node.

1. Check the node tables and calculate $Dist_i^{neigh} = Dist(neigh_i, Dest)$ for $i = 1, 2, ..., N$.
2. Find $Dist_{min} = min\{Dist_i^{neigh}\}$ for $i = 1, 2, ..., N$.
3. If there exists a geographically closer neighbour ($Dist_{min} < Dist(node, Dest)$), forward the packet to neighbor j, where $j = Arg\{Dist_{min}\}$.
4. Else, packet is dropped.
5. End

In the algorithm described here, N is the total number of neighbours. $Dist(node, Dest)$ is the distance of the current node to the destination and $Dist(neigh_i, Dest)$ is the distance of the i^{th} neighbour to the destination. $Arg\{Dist_{min}\}$ is the argument (which neighbour) that has the minimum value of the distance to the destination. In a nutshell, the algorithm, knowing the location of all of a nodes' neighbours (propagated via periodic hello messages), selects the neighbour that is closest to the destination and forwards the packet to this neighbour. If no neighbouring node is physically closer to the destination, greedy geographic forwarding fails and the packet is dropped.

In Figure 4.2, a flow chart of the packet arrival function in RGR is depicted. When a packet arrives, it can be either a control packet or a data packet. Control packets are handled based on their types as RREQs, RREPs, and RERRs. When a data packet arrives, it is checked if there is a reactive route to the destination. If such a route exists, the packet is then forwarded towards the destination on that route. In case there is no reactive routes, data packets can be either from the application layer of the current node, or a message from other neighbours. If the packet is from the application layer of the current node (i.e. the node is the source of data), the

RREQ/RREP mechanism is exploited to find a reactive path. If the packet is a message relayed by neighbours (i.e. the current node is an intermediate node), a switch to greedy geographic forwarding occurs to route the packet geographically towards the destination. In the rest of this section, the details of RGR are explained in different subsections.

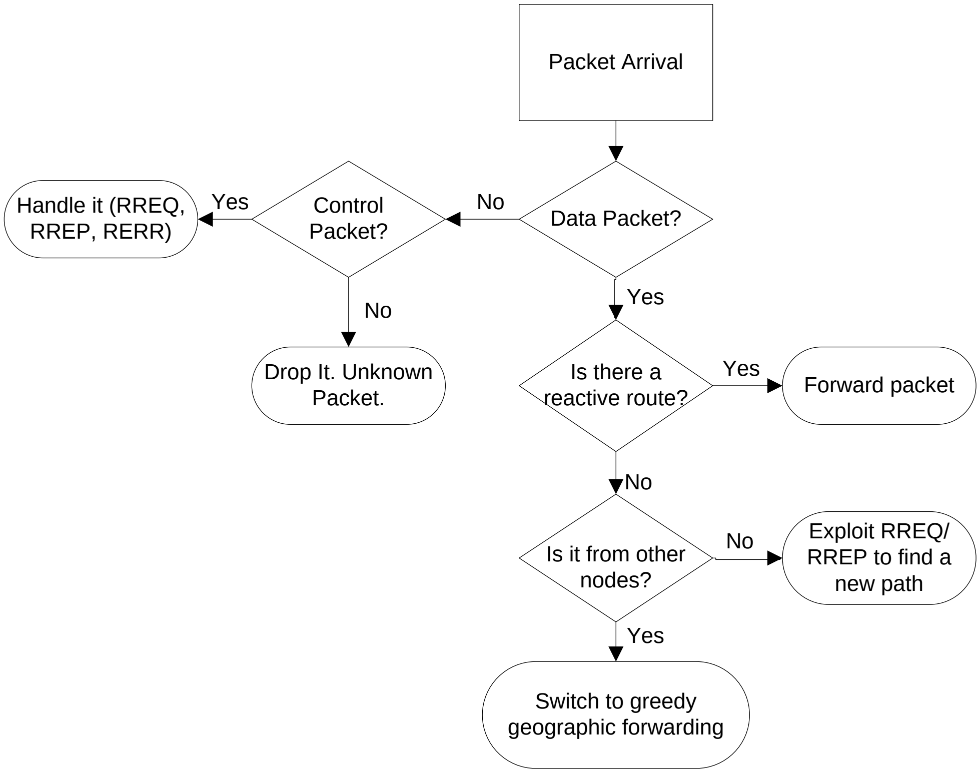
Packet Arrival
Data Packet?
No
Control Packet?
Yes
Handle it (RREQ, RREP, RERR)
No
Drop It. Unknown Packet.
Yes
Is there a reactive route?
Yes
Forward packet
No
Is it from other nodes?
No
Exploit RREQ/ RREP to find a new path
Yes
Switch to greedy geographic forwarding

Figure 4.2: Handling a packet arrival in RGR

4.4.2 Node Tables

In MANET routing protocols, routes towards different destinations are kept in a routing table. Based on the information available in the routing table, a node decides to relay a data packet to one of its neighbours. As an important part of a routing process, we discuss the two different routing tables available in RGR architecture. The RGR routing tables are based on AODV tables. In AODV, a routing table, indexed by destination IP address, keeps the information about a specific destination. In order to keep a list of neighbours, a neighbour table also exists in AODV. In the following subsections, these two tables are explained and the modifications made to change them to RGR table are discussed.

Neighbour Table

The neighbour table is periodically updated by the hello messages received by the neighbours. The neighbour table is a hash table indexed by the IP address of neighbouring nodes. In AODV, when a node receives a hello message from a node, if the node is not already available in the table, the node is added as a neighbour there. If the node is already available in the neighbour table, the entry's time tag is updated to the current time (since timeouts of neighbour entries in the table is similar to AODV, more detail can be found in [1]). The difference of RGR is that each hello message also contains the geographic location of the node. In OPNET, the neighbour table of RGR is implemented as a hash table, indexed by neighbour's IP addresses, which also includes the location information of the neighbours.

Routing Table

A node keeps all of the routes that are passing over the node. The routing table is indexed by the destination IP addresses. RREQs and RREPs in RGR are modified to contain source and destination location information respectively. The routing table indexed by the destination IP address then also contains the location information of the destination that is obtained by the control messages. Thus, every node keeps the destination location information of each of the routes that are using that node. In this thesis, we assume that all the UAVs are flying at the same altitude, similar to [57,71]. This assumption can be helpful in dealing with two parameters as location information (longitude and latitude). However, the discussions proposed here can easily be elaborated to 3D in future work.

4.4.3 Reactive Route

The control packet structure of RGR, rather than using the usual AODV headers, should also contain the destination location information to be able to handle possible switches to greedy geographic routing at intermediate nodes. In order to include the required location information in OPNET, some changes are required in header files and the function block. The detail of these changes are explained in Appendix A.1.

When the source node has a route available towards the destination, the source node uses that reactive route to send data towards the destination. The intermediate nodes also run the reactive part of the RGR. After receiving a packet, the local routing table is checked and the data is forwarded exactly similar to AODV. In the meantime, the neighbour table is updated by hello messages to have up-to-date location information of the neighbouring nodes available in the neighbour table. If there are no link breakages in the network, the RGR mechanism would behave as AODV. The differences occur when there are path interruptions, which trigger the greedy geographic function of the arrival handle function in the process model.

4.4.4 Packet Forwarding in an Intermediate Node

When a packet arrives in an intermediate node, the routing table is checked to see if there is a valid reactive route to the destination. If yes, the packet is sent over the reactive path towards the destination. If the route towards that destination in the routing table is broken (i.e. the next hop neighbour of the path is not in the vicinity anymore), the greedy geographic function is triggered. In OPNET, the process of switching from reactive to greedy geographic forwarding is implemented as a sub-function. Upon reception of a data packet, if the reactive route is broken, the greedy geographic function is triggered.

In the greedy geographic part, the destination location information is obtained from the routing table, as explained in Section 4.4.2. Also, the neighbour location information is distributed via the hello messages and maintained by the neighbour table. The algorithm explained in Section 4.4.1 is then used to forward the data by finding a geographically closer neighbour to the destination. If the node cannot find such a neighbour, the packet is dropped. It is note-worthy that when a packet is dropped by the greedy mode of the protocol, no route error is created. The conditions of generating a route error are discussed in Section 4.4.5. Please note that after a message is handled (either sent or dropped) by the greedy geographic forwarding, the next node first attempts to find a reactive path. If such a path cannot be found, then the greedy geographic forwarding is triggered.

4.4.5 Route Errors

When an AODV route breaks, a route error is sent back to the source. In the OPNET implementation, knowing about RERRs is helpful due to the fact that the switch to greedy geographic forwarding should occur when a route breaks (which can be recognized by the route error functions). Route errors are created and sent in AODV (and equivalently RGR) if one of the three following events happen:

1. The node detects a link break for the next hop of the route. In OPNET, such an error is recognized by a route error type called **AodvC_Link_Break_Detect**.

2. The node receives a data packet for a destination for which it does not have an active route, which is defined as **AodvC_Data_Packet_No_Route** type in OPNET.

3. A route error is received in the node from a neighbor on a route. This is the third type of route errors which is identified by the **AodvC_Rerr_Received** type in OPNET .

In the first two cases, after the RERR process is accomplished, the data forwarding may be handled by the greedy geographic forwarding. The third case, however, occurs when the route error is received from another node, therefore no switching to greedy geographic forwarding is required. In fact, the switch to greedy geographic forwarding should occur when there is a packet in the intermediate node and the route breaks. In other words, even if one of the first two cases happen, the algorithm takes no action. The switch to greedy geographic forwarding only takes place when a packet arrives and a previously available link in neighbour table is broken. The detail of the switch to greedy geogaphic forwarding function is explained in Section 4.4.6.

The route error process function in OPNET is processing a route error depending upon link breaks, no routes, or received route errors. The process of handling either of these three cases is done by checking the route error type. In order to manage the validity of different routes, a lifetime module available in the original AODV messages is also used to delete invalid routes. The functionality of lifetime in OPNET is briefly discussed in Appendix A.3.

4.4.6 Switching to Greedy Geographic Forwarding

Switching to greedy geographic forwarding may take place in intermediate nodes, when the reactive route to destination breaks. The process is implemented in OPNET by using a sub-function that is called by the main packet arrival function to handle the arrivals. As shown in the flow chart in Figure 4.2, when a data packet arrives, the node checks if there exists a reactive path in its routing table indexed by destination IP address. If the route is already broken (due to neighbour movements), rather than dropping packets (or deploying local repair in AODV), RGR executes another sub-function in which the node tries to geographically forward the packet to the destination. To that end, destination location information is extracted from the table and also the neighbours location are extracted from the neighbour table. Based on the Euclidian distance, the node then finds the closest neighbour to the destination. In Appendix A.2, the switching function is explained in more detail.

4.4.7 Handling a Received Packet via Greedy Geographic Forwarding

Consider the case that a packet is received via a greedy geographic forwarder. No matter whether the node received a data packet via reactive routing or via greedy geographic forwarding, at first the reactive option is checked. In the reactive part, the node checks if there is a route to the destination from the current node. The process contains looking at routing table entries to find an entry that is pointing to the desired destination. In order to handle this arrival in OPNET, several possibilities may occur.

The algorithm attempts to find a route to destination. During the process of finding a route, one possibility is the occurrence of a route error in this new node (no reactive next hop to the destination is found). The route error in this case is the result of the reception of a data packet for a destination for which the current node does not have an active route. A RERR (for a specific route) has already been sent when the first switch from reactive to greedy geographic forwarding happened. The question here is to check whether or not the mechanism successfully avoids sending another error message regarding the same source-destination path.

In order to prevent the transmission of unnecessary RERRs in the network, every node keeps a list of its precursor neighbours (i.e. the list of IP addresses that are likely to use this node for a specific destination). Route errors are then sent in a unicast or broadcast manner based on the number of precursors. The node checks if there exists any precursor node. In case there are more than one precursor nodes, a broadcast route error is sent. Else if there is one precursor node, a unicast route error is sent. Finally, if there is no precursor node, no route error is sent.

For the case that a geographic message has been received in the current node, note that the reception of such a packet means that it is more likely that there is no unicast route back to the source node. Also, there is only one unreachable destination, which is the destination of the data packet that cannot be delivered. A RERR is only sent if there is a non-empty precursor list. The nodes receiving the RERR then decide to either forward or discard the RERR packet based on the sequence number and the IP address of the end-points. As a conclusion, without any notable change in the structure of RERRs in AODV, the mechanism can also be used to handle RGR route errors and implement interrupts for switching to greedy geographic forwarding.

4.4.8 Dropping a Packet in the Greedy Geographic Function

When a node has a packet to forward but no reactive route is available, the greedy geographic function is triggered via the main arrival handle function to search for its closest neighbour to the destination. If there is a geographically closer neighbour, the packet is sent to the MANET process model to be forwarded to the desired neighbour. The other case is when there is no geographically closer neighbour, which is discussed in this section.

One of the two following possibilities may occur if no geographic next hop can be found. Either there is no other neighbour (except the previous forwarder) to this node or there is no closer neighbour node than the previous distance to the destination. In either of these cases, the packet is destroyed and the greedy geographic function terminates to be back to the main arrival handle function. In the process model, the node waits for the next arriving packet to handle. Although the packet is dropped, no RERR is sent because the packet is dropped by greedy geographic forwarding (RERRs are only created when one of the conditions mentioned in Section 4.4.5 occurs). Of course, the route error regarding this path interruption has already been sent when the first switch from reactive mode to greedy geographic forwarding occurred.

Since local repairs are disabled in RGR, no attempt in intermediate nodes is made to resend the packet after the greedy geographic forwarding discards the packet due to the non-availability of a geographically closer neighbour towards the destination in its vicinity. The intuition for deactivating the local repair option in RGR is the existence of a greedy geographic alternative. The idea is that the greedy geographic forwarding can help in delivering the packets in case of a link breakage while a globally repaired path is being re-established by the source. The global repair is implemented in the source after reception of the RERR. In the next section, the details of the global repair are explained.

4.4.9 Global Repair by the Source

When a route error occurs, the node recognizing the error delivers the appropriate RERR to the affected neighbours after invalidating existing routes and listing affected destinations. The RERR packet contains all the unreachable destinations and their corresponding destination sequence numbers.

When a RERR is received by a source node, the source finds that a link breakage has happened. However, the source is not aware whether or not the packet has been forwarded (i.e. successfully delivered) by the greedy geographic forwarding. In other words, possibly the source receives a RERR regarding a path to a destination even though the data packet that triggered that message is successfully delivered.

Upon reception of a RERR, the source node first invalidates the route. The invalid route stores previously valid route information for an extended period of time. An invalid route cannot be used to forward data packets, but it can provide information useful for future RREQ messages. The route error processing function processes whether the error is a result of a link breakage, a non-available route, or a received RERR.

After route invalidation, the source broadcasts a new RREQ to the network to find a new path to the destination. While the source node is establishing a new path globally, the greedy geographic forwarding independently helps in forwarding the previously sent packets to the destination.

4.4.10 Destination Operations

The destination can receive a packet either on a reactive route or via greedy geographic forwarding by a neighbour. The packet received by a reactive route can be either from a source for which the destination has a known path to or from a source without any path to. The latter case is only possible when somewhere on the source-destination path at least one switch from reactive to greedy geographic forwarding has occurred. In any of the above cases, when a node recognizes itself as the final destination, the node delivers the packet to the application. If a response is needed, the routing table is checked to find if any fresh route to the required node is available. If yes, the responding packets are sent back over the route similar to AODV.

Note that there is a difference between intermediate nodes' operations and end-points' (i.e. source and destination) operations in handling a received packet. The question is how the model differentiates between an intermediate node and an end-point based on the available functions in OPNET. In intermediate nodes, if the reactive mechanism fails, a switch to the greedy geographic forwarding takes place (ultimately a packet is dropped if the greedy geographic forwarding cannot make a progress in packet delivery). On the other hand, an end-point employs the RREQ/RREP mechanism if there is no route to the destination available in its routing table. Making such a differentiation in OPNET implementation is possible due to the fact that two different types of functions are used in OPNET for creating a packet (in end-points) and relaying a packet (in intermediate nodes). In other words, **create** functions are independent of **arrival_handle** functions. When a node receives a packet, either geographically or reactively, based on the destination IP address specified in the packet, the node can recognize that the packet is destined for the current node. If the packet requires a response (e.g. a RREQ control packet that requires a RREP), the **create** functions of RGR are triggered to create the required response. If there is already a path to the source, the packet will be sent over the reactive path by checking the right entry of the routing table. In case there is no reactive path available in the local routing table, the node should establish a route to be able to send the packet. The process of establishing a route is as explained in Section 4.4.3. The destination in this case behaves as a node that has received a packet from an upper layer and does not have a path in its routing table towards the desired node to forward the packet. In such a case, the reactive part (AODV) will broadcast a RREQ to the network.

In Appendix A.4, some of the major OPNET functions regarding **create** and **arrival_handle** are explained. Greedy geographic forwarding may only be triggered in **arrival_handle** functions, which means that neither the source nor the destination will switch to greedy geographic forwarding even if there is no reactive path in their routing tables. Instead, the end-points, in this case, employ the RREQ/RREP mechanism to establish a reactive route.

4.5 RGR versus Reactive and Geographic Routing

In this section, we qualitatively compare the proposed RGR and the reactive and geographic routing protocols in the literature. The goal in this section is to provide some discussions on the applicability of the RGR mechanism. To that end, we first review the UAANET scenarios and discuss about the applicability of the RGR for those scenarios. Then, different characteristics of the RGR mechanism are briefly compared to their reactive and geographic counterparts.

As discussed in Chapter 2, the specific applications of UAANETs impose a semi-random non-linear trajectory on UAVs (e.g. the tracking object could have unpredictable non-linear mobility). Therefore, pseudo-linearity, which results in a specific design strategy in AANETs, is not a feasible assumption in UAANETs. The mobility trajectory of a UAV is neither completely random nor pseudo-linear. In order to have realistic scenarios, the mobility models should be applied such that the characteristics of a realistic UAANET are considered.

The motivation of our research is to deactivate local repair and use the greedy geographic alternative instead. In order to describe the benefits of such a change, the main specifications of the local repair in AODV and switching to greedy geographic forwarding in RGR can be compared and itemized as follows:

- During a local repair, the packets are buffered in the intermediate nodes. If no RREP is received, after a timeout a RERR is sent back to the original source. Hence, the packets are delayed or even dropped if a buffer overflow occurs. On the other hand, in RGR, as soon as a switch to greedy geographic forwarding occurs, the data packet is handled by the geographic function. Thus, the imposed delay on the packet is expected to be smaller. Also, packets are not required to be buffered, which reduces the possibility of a buffer overflow in the intermediate nodes.

- Upon reception of a RREP as a result of a local repair, the node first compares the hop count of the new route with the value in the hop count field of the invalid routing table entry for the destination. If the hop count of the newly determined route to the destination is greater than the hop count of the previously known route, the node should issue a RERR message for the destination. In RGR, the greedy geographic forwarding is used to handle the current packets while the route is globally repaired. Therefore, a potentially longer path is only used to salvage the current packets. Based on the fact that it is more likely to come up with a longer path as a result of a local repair, in many cases both AODV and RGR will send a RERR and a new path is established. In such a case, refreshing end-to-end routes via a global mechanism can enhance the overall performance of the protocol.

- Sometimes, more than one destination may be affected as a result of a link break. In AODV, the upstream node of the lost link can perform immediate local repair for only one of the destinations. In contrast, the greedy geographic alternative in RGR is handling all of the packets in a First In First Out (FIFO) manner even if they are for different destinations.

- The local repair process can be proactive or on-demand. In proactive, the upstream node tries to repair the route as soon as the break is recognized. In on-demand, the route is only repaired if there are incoming data packets requiring that route. The disadvantage of proactive local repair is that the repaired path may not be used (the intermediate node is not aware when the data session ends). The disadvantage of on-demand local repair is the higher delay that is imposed on the packets. In RGR, switching to greedy geographic forwarding has neither of these disadvantages because of the location-based nature of the forwarding.

- Local repair may occur only if the destination is not more than a linear fraction of the maximum possible number of hops between two nodes in the network. In RGR however, the switch to greedy geographic forwarding is independent of the location of the broken link on the route.

What we can conclude about the functionality of local repair in AODV is that the mechanism is more helpful when most of the nodes have low mobility and remain in each other's vicinity, and only one or a limited number of nodes move away. Local repair is ideal in such a case because the repaired route would not be much longer than the original path. Also, the repaired path can be used for a longer time as a result of the stability due to the low relative mobility in the network. One risk of local repair is the case when the repair process is unsuccessful. In other words, we pay the cost of the local repair (RREQ/RREP in an intermediate node, buffering the data packets, and etc.) while the whole process is useless and after a timeout, finally the packets are dropped and a RERR is sent back to the source. When the nodes have relatively higher mobility, the possibility of such a worst case in local repair performance increases. One other situation that local repair is not as beneficial as RGR is when there are different destinations in the network, the performance of local repair degrades due to the fact that the mechanism can support only one of the destinations at a given time.

The RGR design does not require local route maintenance similar to geographic protocols and in contrast to many reactive routing mechanisms. However, RGR is required to have access to neighbour location information to be able to perform greedy geographic forwarding. Unlike geographic routing protocols, an independent location service is not required in RGR due to the fact that the location information is provided by the AODV RREQ/RREP mechanism.

The RGR mechanism is more complicated and requires the dissemination of location information compared to AODV. The higher overhead is the cost that we pay in order to provide end-to-end connectivity for a density variable highly mobile network architecture without requiring reactive local repairs and independent geographic location service.

Another note-worthy point about RGR is that the design is for a networking architecture that has a higher relative mobility compared to traditional MANET scenarios. Therefore, the expectation is to have more route interruptions in the network compared to traditional MANETs. The fact is that switching to greedy geographic forwarding provides a best-effort alternative in cases that a route interruption occurs. In Table 4.1, the main characteristics of RGR mechanism are compared with those of AODV and geographic routing.

4.6 Summary

In this chapter, the fundamentals of a new routing protocol for unmanned aeronautical ad-hoc networks were introduced. RGR is a routing mechanism that can be adaptively matched to different topological scenarios in an ad-hoc network. Unlike the protocols proposed in the available literature that are only applicable to some specific scenarios,

Table 4.1: Comparison of RGR with reactive/geographic routing mechanisms

Parameter	AODV	Geographic Routing	RGR
Local maintenance	Required	-	-
Location Service	-	Required	-
Route request	Required	-	Required
Neighbour location	-	Required	Required
Motivation (Application)	Connectivity in MANETs	Scalability in dense MANETs	Handling higher mobility
Control messages	Route establishment /maintenance	Neighbour discovery	Route establishment /neighbour discovery
Routing choice made at	Source and intermediate nodes	Intermediate nodes	Source and intermediate nodes
Mobility specifications	Static/Low mobility	Low mobility	Fairly static to highly mobile

mostly dense low mobility scenarios, the proposed RGR can be adaptively matched to different scenarios by switching from reactive to greedy geographic forwarding or vice versa. When the network is more static (relative velocities are small), the number of switches to greedy geographic forwarding is less. When there are more route breaks in the network, more switches to greedy geographic forwarding occur. After the qualitative comparison presented in this chapter, the next chapter demonstrates the performance of RGR quantitatively.

Chapter 5

Simulation Results

5.1 Introduction

In this chapter, we compare the performance of RGR with greedy geographic forwarding and two different versions of AODV (with and without local repair). Similar to the available literature [45,72], we consider packet delivery ratio, end-to-end delay, and overhead to compare the performance of these three mechanisms. In this section, these metrics are discussed in detail. We also explain how we extract the statistics from the simulation environment and what simulation parameters are depicted in the figures. It is note-worthy that the accuracy of the simulation results presented in the thesis, are all evaluated by 90% confidence intervals.

The rest of this chapter is organized as follows. In the reminder of this section, we introduce the three performance metrics. Since the mobility model is an important factor in UAANET performance, a discussion on the mobility scenarios exploited in the simulations is included in Section 5.2. The attempt is to set the configurations such that it can correctly model plausible UAV deployments. The other important parameters in the simulation scenarios are related to the physical layer and MAC, which are explained based on the OPNET model specifications in Section 5.3. The end-to-end traffic of the network is explained in Section 5.4. Since we simulate four different routing protocols in the network, the routing parameters for each of the protocols are explained in Section 5.5. In Section 5.6, the simulation results for a searching scenario are presented, followed by the results in a tracking mission in Section 5.7. Statistics showing the number of switches from the reactive part to the greedy geographic mechanism are provided in Section 5.8. Finally, the chapter is summarized in Section 5.9.

5.1.1 Packet Delivery Ratio

In RGR, the greedy geographic forwarding salvages some of the data packets that are possibly dropped by the original AODV protocol. Therefore, the expectation is that RGR will achieve a better packet delivery ratio. RGR automatically switches to greedy geographic forwarding when a path breaks, thus the mechanism can handle high mobilities more efficiently without affecting delivery ratio in a connected cluster.

In order to measure packet delivery ratio, 10 independant scenarios are generated in OPNET. Each of those scenarios is generated using a different seed of the pseudo-noise sequence generator available in the OPNET core. We consider the same 10 seeds for each routing protocol to gather 10 sets of pseudo-independent results. Then, instantaneous packet delivery ratio (PDR) for a seed i is called $PDR^i(t)$ and is calculated as:

$$PDR^i(t) = \frac{T^i_{rec}(t)}{T^i_{sent}(t)} \tag{5.1}$$

$T^i_{rec}(t)$ and $T^i_{sent}(t)$ are respectively the traffic received and the traffic sent in the interval $[t-10, t]$ (note that in the discrete event model of OPNET, we gathered the discrete data every 10 sec for the past interval). Averaging $PDR^i(t)$ over all 10 seeds, the result is called PDR(t) and is calculated as:

$$PDR(t) = \frac{1}{10}\sum_{i=1}^{10} PDR^i(t) \tag{5.2}$$

The figures in this chapter depict average PDR ($APDR(t)$), which is the time average of $PDR(t)$, and is calculated as:

$$APDR(t) = \frac{1}{t}\sum_{x=1}^{t/10} PDR(10 \cdot x) \qquad t = 10, 20, \ldots, 1000 \tag{5.3}$$

As shown in Equation 5.3, $APDR(t)$ provides the average delivery ratio from the beginning of the simulation until time t. In other words, the delivery ratio in the interval $[0, t]$ is depicted for each of the three protocols from t=0 to t=1000 in increments of 10 seconds.

5.1.2 Delay

Delay is another important parameter which is used to evaluate RGR compared to AODV and greedy geographic forwarding. The delay in AODV is mostly the result of route establishment/maintenance. RGR is using an AODV-style RREQ/RREP process to establish a route, therefore the delay behaviour of RGR during the RREQ/RREP process is expected to be similar to AODV. When a route failure occurs, the behaviour of RGR in terms of RERR generation is similar to AODV without local repair, as discussed in Chapter 4. The delay of greedy geographic forwarding, however, is mostly the result of processing delay at intermediate nodes, which is much smaller than the delay of reactive route establishment/maintenance. The other important fact about measuring delay in OPNET is that we can only only compute the delay for packets that can reach their destinations. Otherwise, the delay parameter in OPNET is a Not Available (N/A) parameter.

The derivation of the delay calculation in the figures is similar to the packet delivery ratio. We consider the same 10 seeds, and then compute the average delay as follows:

$$D_{avg}(t) = \frac{1}{10t} \sum_{x=1}^{t/10} \sum_{i=1}^{10} D^i(10 \cdot x) \qquad t = 10, 20, \ldots, 1000 \tag{5.4}$$

In Equation 5.4, $D_{avg}(t)$ is the average delay imposed from the begining until time t, and $D^i(x)$ is the delay imposed by the protocol using the seed i at time interval $[x - 10, x]$.

5.1.3 Overhead

We also compare the overhead of RGR with AODV and the OPNET implementation of greedy geographic forwarding. Consider $OH^i(t)$ as the overhead of the routing protocol using the seed i to generate mobility scenario, which is defined as the number of control packets such as RREQs, RREPs, RERRs, and hello messages.

$$OH^i(t) = RT^i_{sent}(t) \tag{5.5}$$

Please assume $RT^i_{sent}(t)$ as the total routing traffic sent (either by endpoints or intermediate nodes) for seed i during the time interval $[x - 10, x]$ including RREQ, RREP, RERR, and hello messages. In other words, the routing traffic is the amount of extra traffic that is distributed in the network in order to provide the possibility of sending data packet.

The overhead in the past 10 sec, as shown in Equation 5.5, is the count of routing packets sent in that interval. The average overhead in the last t sec can be calculated as:

$$OH(t) = \frac{1}{10t} \sum_{x=1}^{t/10} \sum_{i=1}^{10} RT^{i}_{sent}(10.x) \qquad t = 10, 20, \ldots, 1000 \tag{5.6}$$

OH(t) is the average overhead of the protocol in the past t seconds over all 10 seeds.

5.2 Mobility Modelling

An important part of the simulation configuration is to have suitable mobility scenarios, which represent plausible UAANET deployments. In general, proposing accurate mobility scenarios for UAANETs is an open research problem (which is beyond the scope of this research) [26, 27]. In this thesis, the attempt was to use one of the available OPNET mobility models, which can be adapted to fit searching and tracking missions by changing several OPNET settings. In the following subsections, we discuss the mobility scenarios that were derived for both searching and tracking missions.

One characteristic of the scenarios in this chapter is that we do not force the UAVs to be in each other's vicinity. This means that at some point in time, the network can be possibly disconnected as this is the case in realistic UAANET applications (and potentially forming clusters). In fact, in a searching application, we expect to have more interruptions because the UAVs should be expanded in a region. The tracking application, however, is expected to have a smaller number of network partitions. This important characteristic is one of the key features to consider in modelling searching and tracking applications. In the following subsections, searching and tracking scenarios are separately explained.

The mobility scenarios are derived from the RWP model. In the RWP mobility model, a node randomly chooses a location and speed, moves from its current position to that location in a straight line, stays at that destination for a *pause time* seconds. Then the process repeats. The main difference of the mobility model used in our case compared to the original RWP is the fact that the speed is constrained by two parameters: minimum and maximum values. In the original RWP, only the maximum speed is used as a parameter (i.e. the minimum speed is always assumed to be 0).

The other difference with respect to the original RWP is the fact that all the nodes start in a small constrained initial area, modelling a common launch site for the UAVs. In the original RWP model, nodes are initially uniformly distributed in the whole region. As time passes, nodes move around based on the RWP parameters. In particular in the search scenarios, eventually the UAVs will spread over the complete deployment area.

5.2.1 Mobility Parameters for Searching Mission

In order to model a searching mission of an UAANET, a square area is considered. Since we have assumed two different sizes for the mission, the exact size of the region is explained for each set of simulation results separately. The assumption in a search mission is that every UAV is looking at different places to find the desired object. Therefore, the RWP can model the mobility of the UAANET, especially when UAVs move independently (in fact, our model does not consider UAVs' statistical inter-dependencies). In such a case, a node chooses a destination and speed, and then moves from its current location at that speed towards the destination. A node then remains at that location for *pause time* seconds and the process repeats. We consider a continuous flight mission in which the UAVs never come to a rest, which is why *pause time* is set to 0. Also, the mission starts at time 0 and ends at t=1000 sec, which is the end of the simulation. The mobility characteristics for search applications are summarized in Table 5.1. Please note that UAVs are expanded all over the region independently to look for the object. The independent random mobility may divide the UAANET into several partitions.

Based on the mobility parameters described in Table 5.1, we definded three different UAV scenarios. The first scenario models a low speed searching UAANET in which the UAV velocity is changing based on a uniform distribution in the $[10, 20]$ m/s interval. In the medium velocity model, the UAVs uniformly select a velocity in the $[30, 40]$ m/s interval, and finally our high speed scenario has UAV speeds uniformly distributed in $[50, 60]$ m/s. Please note that these range of speeds are typical values for a UAANET including medium size UAVs [7].

Table 5.1: Mobility parameters of a searching scenario

Parameter	Value
Mobility Model	Random Waypoint
Low Speed Scenario	Uniform(10, 20) m/s
Medium Speed Scenario	Uniform(30, 40) m/s
High Speed Scenario	Uniform(50, 60) m/s
Initial Region	1×1 km^2 square with a vertex on (0, 0)
Size	4 km^2, 25 km^2
Number of UAVs	10, 20, 30
Pause Time	0
Start Time	0
Stop Time	End of Simulation
Simulation Time	1000 s

5.2.2 Mobility Parameter for Tracking Mission

For modelling a flock of UAVs participating in a tracking mission, another modification of the RWP can be used in OPNET. In this model, all UAVs are moving towards a target region. This target region is a 2000 × 2000 m square, which is 125 km away from the origin (where the UAVs start their mission). The region is considered 125 km away to make sure that the implemented scenario correctly models the tracking mission. If the region is somewhere closer, the UAVs would possibly reach there before the simulation ends, which is not desirable for modelling a tracking mission (when the UAVs reach to the region, the model will become a searching model, as discussed in Section 5.2.1). In this model, UAVs travel towards the region while there is a randomness in their trajectories. The details of the mobility parameters for an UAANET in tracking mission are summarized in Table 5.2.

In a tracking mission, UAVs' speed are changing based on a uniform distribution in the range [17, 20] m/s, [36, 40] m/s, and [55, 60] m/s for low speed, medium speed, and high speed scenarios respectively [7]. Compared to a search mission, the uniform interval for the velocities is smaller. The reason for such a selection is that in a tracking mission, UAVs are assumed to follow a target on the ground, therefore they would have smaller deviations in their speeds and directions. In other words, the target would cause the UAVs to have more correlated mobility vectors. In OPNET, we model this phenomenon by a smaller uniform interval for the velocity vectors. The other fact is that as the target moves faster, we expect to have more deviation in UAV mobility. Hence, we increase the uniform interval width from 3 for low speed scenarios to 4 and 5 for medium and high speed scenarios respectively. One other difference of the tracking scenarios compared to searching scenarios is that the number of UAVs in tracking scenarios is usually less than the number of UAVs in a searching scenario. The intuition is that we need more UAVs to search an unknown area rather than tracking a known object. Based on this intuition, we only consider the values of 10 and 20 UAVs in tracking mission. Table 5.2 shows the velocity specifications of the tracking scenarios.

Table 5.2: Mobility parameters of a tracking scenario

Parameter	**Value**
Mobility Model	Flocking UAVs
Low Speed Scenario	uniform(17, 20) m/s
Medium Speed Scenario	uniform(36, 40) m/s
High Speed Scenario	uniform(55, 60) m/s
Number of UAVs	10, 20
Initial Region	1×1 km^2 square with a vertex on (0, 0)
End Region	125 km away from Origin
Size of End Region	4 km^2
Pause Time	0
Start Time	0
Stop Time	End of Simulation
Simulation Time	1000 s

5.3 MAC/PHY Specifications

The propagation model considered in our simulations is a free space path loss, which models the propagation as a disc around the transmitter. Although channel modelling of UAANETs can be a research topic based on the specific channel characteristics imposed by UAV mobility and the environment [73], the effects of channel impairments are not addressed in this research. Instead, the focus of this thesis is on routing protocol design, thus assuming a simple channel model with a predictable set of specifications is acceptable. Based on the packet reception power threshold, which is -95 dBm, the transmission range of the UAVs will be 1000 m. Transmit power for acquiring such a range is 0.00322798735385 W. MAC layer specifications are also listed in Table 5.3. The values for such a setting are assumed to be typical values for medium size UAVs based on [74].

Table 5.3: MAC layer specifications

Parameter	**Value**
Protocol	IEEE 802.11
Data Rate	1 Mbps
Transmission Range	1000 m
Packet Reception Power Threshold	-95 dBm
Buffer Size	256000 bits

5.4 Network Traffic

The definition of traffic flows in the UAANET should also be such that it can be as close as possible to a realistic mission. In our simulations, two nodes of the scenario are called head UAVs. These nodes communicate with each other bi-directionally. Every other nodes in the network has uni-directional flows towards each of these nodes.

Considering such traffic flows are compatible with a UAANET scenario in which several UAVs are receiving data from other nodes. Although it may be required to have a flow among any pair of nodes in a realistic UAANET mission, in our model we assume that the nodes in such a case will communicate via two head UAVs available in the network. The reason for considering such a structure for traffic flows is two-fold. First, this will help to have more statistically accurate simulation results because we average all of the flows in the network in terms of the metrics introduced in Section 5.1. For example, in a network of 10 UAVs, we run the simulation over 10 seeds. The flows available in the model are 18, 9 flows to each head UAVs. The results that we extracted from the simulation results then are for 180 flows, although we have run 10 seeds. Due to the randomness of the mobility and randomness of traffic generation, when we average over all of the seeds we can come up with a better results. In this case, with smaller number of simulation runs, we can obtain more accurate results, which can decrease the simulation time. The other important fact to use such flows in the network is that we can test adaptability of the proposed RGR protocol in dealing with multi-flows in the network. One important feature of a routing protocol is the ability to handle multi-flows. Different protocols may impose different routing delays, and processing time in intermediate nodes. In order to have a realistic comparison of the protocols, assuming such an environment is required. Thus, unlike many literature available on the topic which only consider very limited number of flows, we test the protocols in a more realistic scenario assuming several flows in the network.

We gathered sent and received traffic, delay, and overhead. Each flow has an exponential packet size with an average length of 1024 bits and an expected packet inter-arrival time of 0.2 sec for searching missions and 0.5 sec for tracking missions. The reason for assuming different inter-arrival time for searching and tracking missions are explained in the simulation results section. Considering exponential interarrival time and packet size are typical in the literature due to the memoryless property of the exponential distribution. The generated traffic specifications for searching and tracking missions are available in Table 5.4.

It is worth-noting that the packet inter-arrival time for searching mission is 2.5 times higher than tracking mission in Table 5.4. This assumption is to confirm the fact that in a realistic mission, the traffic sent in a searching mission is expected to be higher than a tracking mission. The other assumptions that we made about traffic flows are regarding packet inter arrival time distribution and packet length distribution. For both of searching and tracking missions, these two values are exponential. These two values may have any other distribution other than exponential in some realistic examples. The main reason for assuming exponential packet length and inter-arrival time is the memory-less property of exponential distribution, which makes it popular for observing statistical behaviour of network flows.

Table 5.4: Traffic parameters

Parameter	Searching scenario	Tracking scenario
Start Time	0 sec	0 sec
Packet Inter-arrival Time	Exponential (0.2)	Exponential (0.5)
Packet Size	Exponential (1024) bits	Exponential (1024) bits
Stop Time	End of Simulation	End of Simulation

5.5 Routing Protocol Setting

In this section, the default values for all the routing protocols discussed in the rest of the chapter, are explained. Due to the fact that RGR is implemented based on the available AODV process model in OPNET, we have used the same sets of configurations for these two protocols. Also, the values we considered for RGR and AODV are all the default values in OPNET [70]. AODV and RGR parameters are available in Table 5.5. Please note that the values defined in Table 5.5 are based on the standard definition in [1]. As discussed in Section 4.3.1, AODV is using an expanding ring search technique to limit the number of RREQs in the network. In expanding ring search, a node sets its TTL value to TTL Start and sets the timeout for receiving a RREP to Ring_Traversal_Time[1] sec. If a RREQ times out without a corresponding RREP, the source broadcasts the RREQ again with a value incremented by TTL Increment.

The values considered for greedy geographic forwarding are all the default OPNET values, and are summarized in Table 5.6. As you can see from these two tables, the hello interval in greedy geographic forwarding is almost 5 times longer than in AODV and RGR. One may ask why not to consider the same value of hello packet intervals for greedy geographic forwarding and RGR. The reason for such selection was that

[1] RING_TRAVERSAL_TIME= 2×NODE_TRAVERSAL_TIME×(TTL_VALUE+TIMEOUT_BUFFER)

Table 5.5: AODV/RGR configurations

Parameter	Value
Active Route Timeout	5 sec
Hello Interval	Uniform (1, 1.1)
Allowed Hello Loss	3
Net Diameter	35
Node Traversal Time	0.04
Route Error Rate Limit	10 pkts/sec
TTL Start	1
TTL Increment	2
TTL Threshold	7
Timeout Buffer	2

RGR is a combination of both, therefore we considered the minimum value of these two values which was the hello interval of AODV.

Table 5.6: Greedy geographic forwarding configurations

Parameter	Value
Hello Interval	Uniform (4.9, 5)
Neighbour Expiry Time	3
Number of Initial Floods	2
Backtrack Option	Disabled

5.6 Simulation Results for a Searching Mission

In this section, simulation results for a searching mission with 10, 20 and 30 UAVs are reviewed. We consider two sizes for the search region. The first size is a 2000×2000 m square (which is only tested for number of UAVs equal to 10) and the other is a 5000×5000 m square. In the following subsection, the results for number of nodes equal to 10, 20, and 30 are separately discussed. Also, the parameters used in different scenarios are briefly mentioned and the results are evaluated and explained.

5.6.1 Small Network with 10 UAVs

In Figure 5.1, the average delay for all routing protocol is depicted for a low speed UAANET in a region of size 4 km^2. The improvement in average delay of RGR compared to AODV with local repair is due to the fact that the waiting time in intermediate nodes is smaller in RGR. Please note that local repair queues the packets while searching for a new route to the destination. Compared to AODV without local repair, RGR has a slightly better delay, although it is not as significant as in the case with local repair. This improvement is due to the fact that in RGR, when a switch to greedy geographic forwarding occurs, the packet is geographically forwarded to the next hop. At the same time, a global repair is initiated to establish a new path.

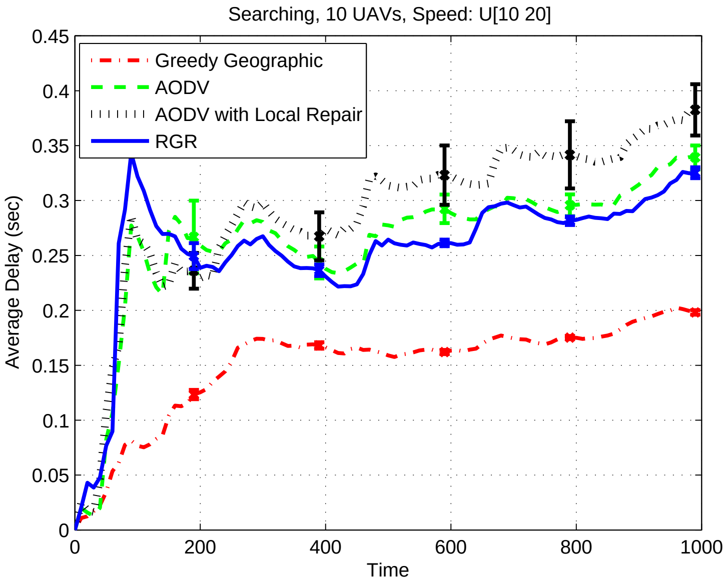

Figure 5.1: Average delay in a searching scenario of size 4 km^2 including 10 low speed UAVs

Figure 5.2 shows the delivery ratio of RGR versus other protocols for a low speed UAANET in a region of size 4 km^2. In terms of delivery ratio, the performance of RGR, AODV and AODV with local repair are close to each other. The reason is

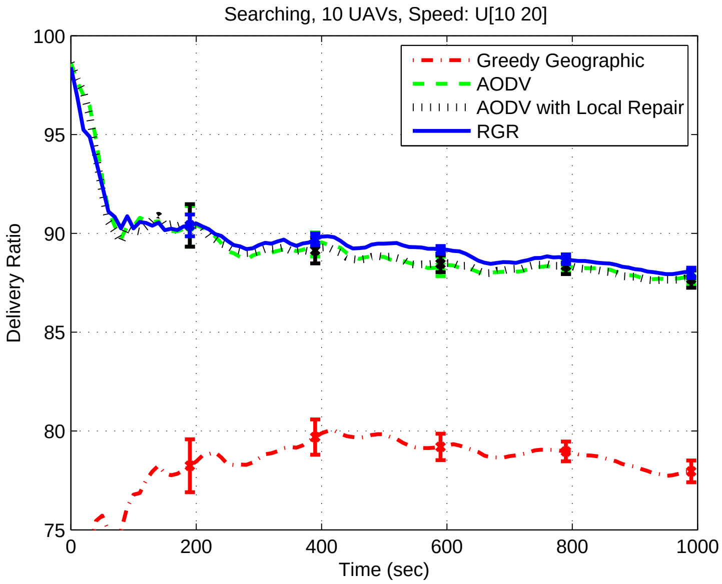

Figure 5.2: Packet delivery ratio in a searching scenario of size 4 km^2 including 10 low speed UAVs

that when the number of nodes is equal to ten, there is not enough options to find a better node when a route breaks. Also, route breaks are sometimes the result of a partitioned network for which no routing approaches can achieve packet delivery. Unlike in Chapter 3, since we do not force UAVs to be in each others' transmission range, there exists the possibility of such an interruption. Evaluation of such a phenomenon in UAANETs is left as a future research. In Figure 5.2, also note that the greedy geographic forwarding packet delivery ratio is considerably lower than other protocols, which reconfirms insights from Chapter 3 that the greedy forwarding alone is not a sufficient approach.

Figure 5.3 represents the average number of routing traffic for a low speed UAANET in a region of size 4 km^2. The number of routing packets in AODV and RGR is much higher than that of greedy geographic forwarding. A part of this result is due to the hello intervals which are sent almost 5 times more frequently in RGR and AODV than greedy geographic forwarding. Besides that, in RGR and AODV, the routing functionality is based on sending RREQs, RREPs, and RERRs. It is shown that the routing traffic for RGR and both versions of AODV are very close to each other, which is based on the fact that many times they require the same control traffic. The difference in the value of routing traffic in these protocols happens in

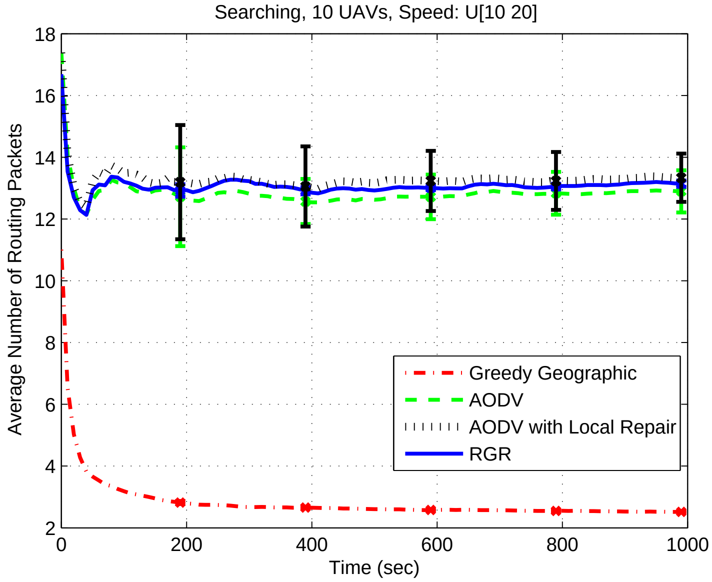

Figure 5.3: Average routing traffic in a searching scenario of size 4 km^2 including 10 low speed UAVs

case of a link break, where each of the mechanisms handle the problem differently, as discussed in Chapter 4.

In Figure 5.4, the average delay in a searching scenario of size 4 km^2 including 10 medium speed UAVs is depicted for all four protocols. The average delay of RGR is similarly better than that of AODV. When the velocity increases in the network, more reactive routes may break, which causes a switch to greedy geographic forwarding in RGR. Due to the fact that the queueing time in RGR is smaller than AODV with local repair, we can see an improvement in terms of delay. Also, as explained in the previous part for low speed scenario, the improvement of RGR is also the result of the fact that while greedy geographic forwarding salvages the packet, the process of global repair is performed.

Figure 5.5 shows the delivery ratio in a searching scenario of size 4 km^2 including 10 medium speed UAVs. As the mobility increases in the network, more reactive routes break [75]. In such a case, RGR employs the greedy geographic forwarding. Therefore, we can expect that a part of the packets are recovered by using greedy geographic forwarding and a better packet delivery ratio can be achieved in the long term.

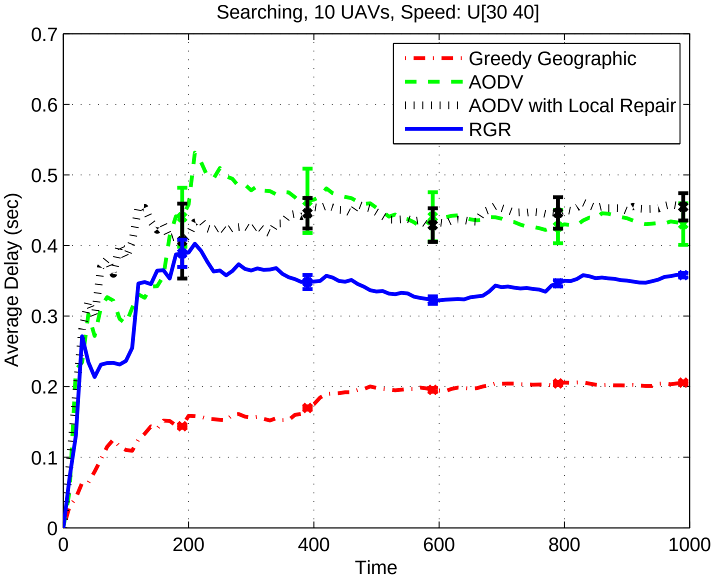

Figure 5.4: Average delay in a searching scenario of size 4 km^2 including 10 medium speed UAVs

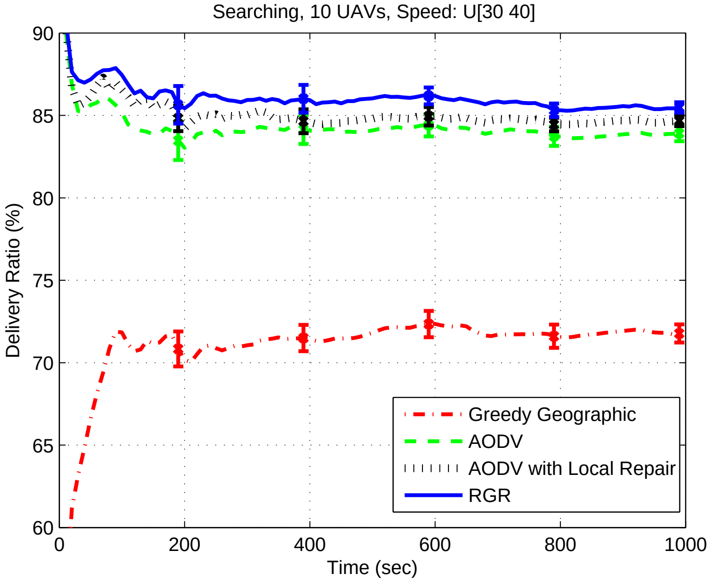

Figure 5.5: Packet delivery ratio in a searching scenario of size 4 km^2 including 10 medium speed UAVs

Figure 5.6 shows the average routing traffic in a searching scenario of size 4 km^2 including 10 medium speed UAVs. As expected, the number of control messages in

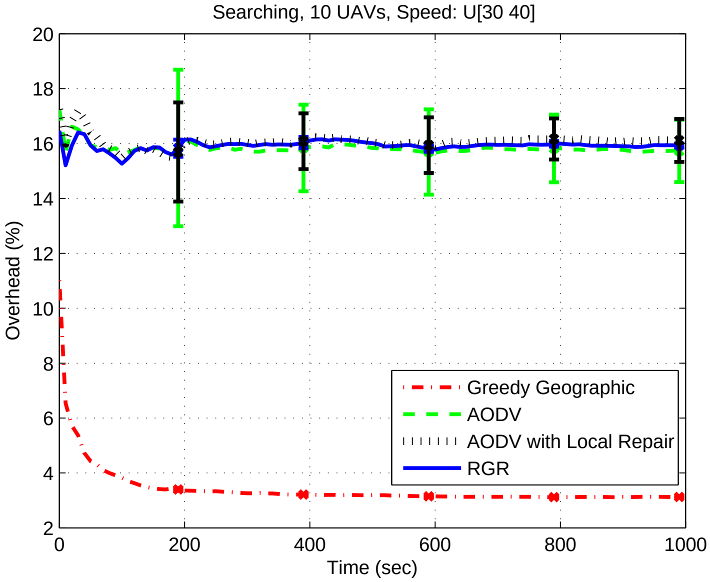

Figure 5.6: Average routing traffic in a searching scenario of size 4 km^2 including 10 medium speed UAVs

RGR and AODV is higher than for greedy geographic forwarding.

The other point to notice about Figure 5.1 to Figure 5.6 is the performance of greedy geographic forwarding. In terms of average delay and overhead, greedy geographic forwarding is better. The reason is that no waiting time is imposed in intermediate nodes. The imposed overhead of greedy geographic forwarding is smaller due to the fact that no route discovery/maintenance is required. In general, geographic routing protocols are more scalable than reactive routing protocols [3]. However, the disadvantage of greedy geographic forwarding is the fact that the packet delivery ratio is much lower. This lower delivery ratio is because in greedy geographic forwarding a packet is dropped if a better neighbour cannot be found. Since the same trend is observed for the performance of greedy geographic forwarding, in the rest of the chapter, we focus more on RGR compared to AODV, unless we observe a phenomenon that needs more explanation.

Figures 5.7, 5.8, and 5.9 represent average delay, average delivery ratio and average routing traffic for an UAANET deployment including 10 high speed UAVs. The intuition about the figures is similar to the low and medium speed cases. In Figure 5.7, the average delay of AODV with local repair is more than AODV and RGR, which is the result of the increase in unsuccessful local repair attempts (RREQ/RREP procedures) in intermediate nodes as intuitively explained in Section 4.5.

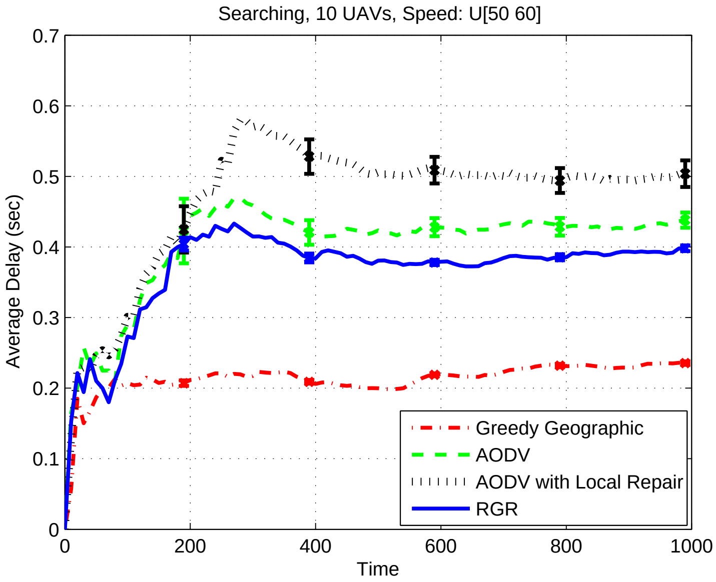

Figure 5.7: Average delay in a searching scenario of size 4 km^2 including 10 high speed UAVs

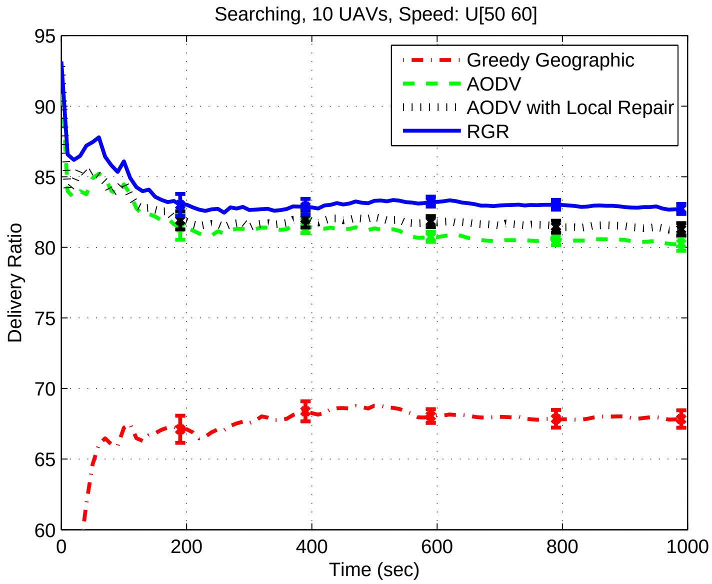

Figure 5.8: Average delivery ratio in a searching scenario of size 4 km^2 including 10 high speed UAVs

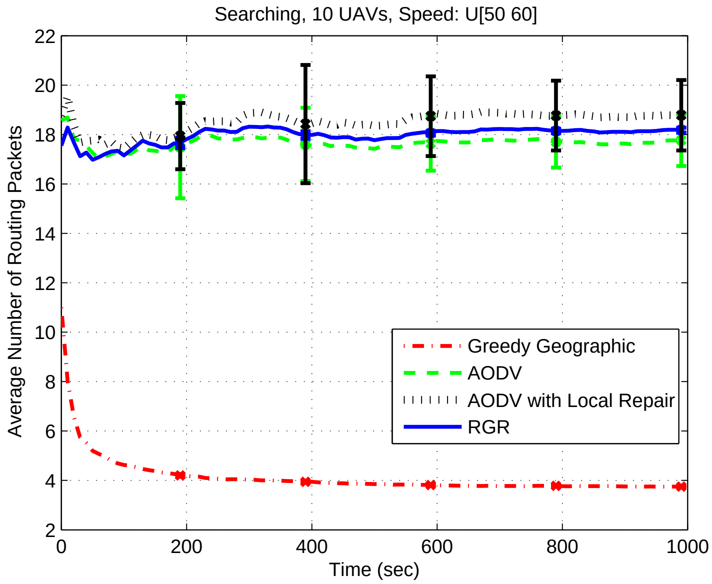

Figure 5.9: Average routing traffic in a searching scenario of size 4 km^2 including 10 high speed UAVs

Figure 5.9 represents the average number of routing packets. The number of routing packets in RGR is closer to AODV without local repair compared to AODV with local repair. In RGR and AODV without local repair, when a route breaks, a RERR is sent back to source. In AODV with local repair, however, the intermediate node sends RREQs locally to resolve the broken route. When the network has higher mobility, more links break and more routing packets are sent in intermediate nodes, which helps explain the difference between AODV with and without local repair.

We also have used 10 UAVs in an area of 25 km^2. Since the scenario is sparser in this case, it is more likely to result in a partitioned network. As a result of the fact that the same number of UAVs are used in a much wider area compared to previous configuration, we expect to have more link breaks. In order to keep the main body of the thesis more readable, the results of a searching mission including 10 UAVs are shown in Appendix B.

At the beginning of the simulation, all the nodes start their movement from an initial region of 1×1 km^2 square with a vertex on (0, 0). In other words, when simulation starts, the UAVs are close to each other. Therefore, the delivery ratio is higher due to the fact that every UAV can be accessed. After some time, the performance degrades, which is the direct result of the topology because all the protocols have similar behaviour.

One observation about the average delay of greedy geographic forwarding is the number of routing packets, which is much higher in the initial phase of all the simulations. This phenomenon is based on the fact that in OPNET implementation of greedy geographic forwarding, an initial flooding is done before commuincation starts. In our simulations, the number of initial floods is set to 2, as shown in Table 5.6. During this initial flooding phase, nodes distribute their IP addresses and location information in the network. Therefore, in the beginning of the simulation, the greedy geographic forwarding has a higher average number of routing packets. As the average number of nodes decreases, an increment in the delivery ratio can be observed. Also, the average number of routing traffic will be smoother after sometime, which is the result of periodic hello messages, which leads to a constant overhead after the initial time.

5.6.2 Large Network with 20 UAVs

We also collected results for 20 UAVs in a region of 25 km^2. The UAVs are deployed in three different scenarios with low, medium and high speeds. The delay of RGR outperforms AODV, as explained for the previous scenarios. The improvement of RGR delivery ratio compared to both versions of AODV increases as the number of velocity of UAVs increases, which confirms the previous discussion on the applicability of RGR at higher speeds. Another important observation is the fact that the delivery ratio of AODV with local repair is not significantly improved compared to AODV without local repair, especially at higher velocities (while the delay and overhead of AODV with local repair are worse). As discussed in Chapter 4, the current implemenation of local repair in AODV is more useful for scenarios in which the network has low mobility and only one node (or a limited number of nodes) moving away. In more dynamic networks such as UAANETs, local repair can even be worse than the original AODV due to the fact that the attempt in intermediate nodes, to locally find a path to destination, is more likely to be unsuccessful. Further discussion on the performance of local repair compared to RGR is presented in Section 4.5. The results that we observe for a scenario with 20 UAVs are similar to the results that we observe for a network of 10 UAVs in a 4 km^2 area. Therefore, we do not provide the same explanation for these results. In the rest of this subsection, those figures with some interesting characteristics are explained in more detail. Figures 5.10, 5.11, and 5.12

represent the average delay, average delivery ratio, and average routing traffic for a low speed searching mission for a 25 km^2 region with 20 UAVs.

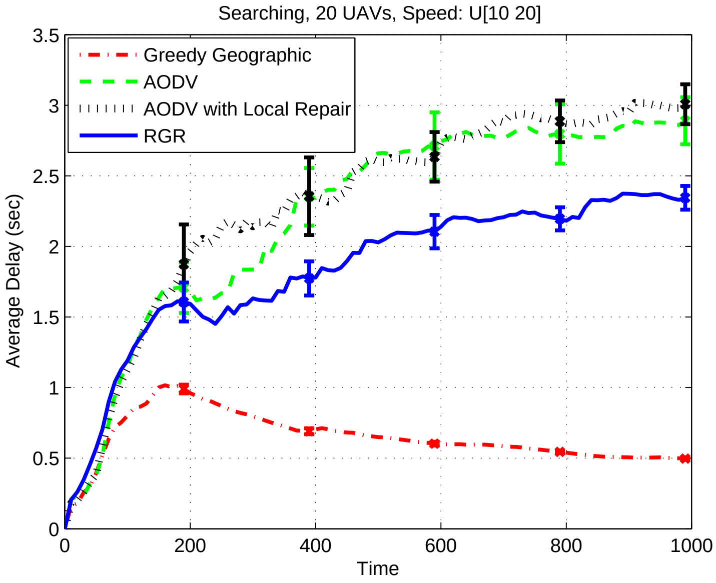

Figure 5.10: Average delay in a searching scenario of size 25 km^2 including 20 low speed UAVs

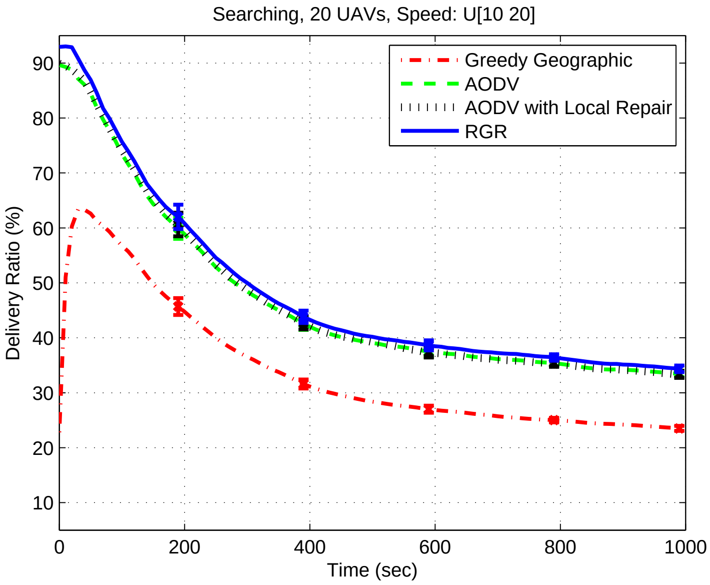

Figure 5.11: Average delivery ratio in a searching scenario of size 25 km^2 including 20 low speed UAVs

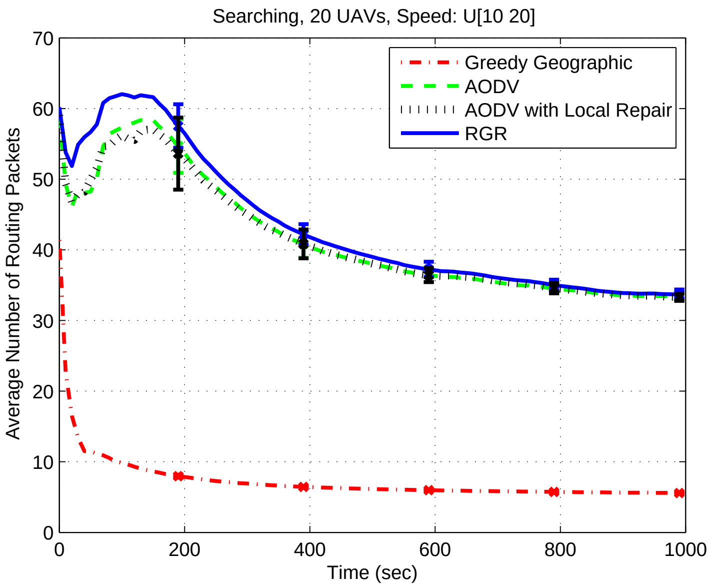

Figure 5.12: Average routing traffic in a searching scenario of size 25 km^2 including 20 low speed UAVs

Figure 5.13 represents the average delay in a searching scenario with 20 medium speed UAVs. In a medium speed scenario with 20 UAVs, the delivery ratio of RGR is close to AODV as shown in Figure 5.14. Also, the average routing traffic of the scenario is depicted in Figure 5.15, confirming that the average number of routing traffic in RGR is close to AODV without local repair in the long term. Despite the attractive low overhead and delay of greedy geographic forwarding, as shown in Figure 5.13, its lower packet delivery ratio make it less applicable for scenarios that higher delivery is required.

By observing Figure 5.13, Figure 5.14, and Figure 5.15, we can conclude that there are some scenarios in which the performance of RGR does not have a significant improvement in terms of delivery ratio. However, the good point is that with a comparable amount of overhead, RGR can improve the delay, which is vital for delay-critical applications. The reason for the improvement of RGR delay compared to AODV without local repair is in scenarios when an intermediate link breaks. In such a case, both RGR and AODV sends a RERR back to source. RGR can immediately salvage some of the packets in the intermediate nodes. Please note that the packets in the source node, in both cases, should wait until a new route is found by

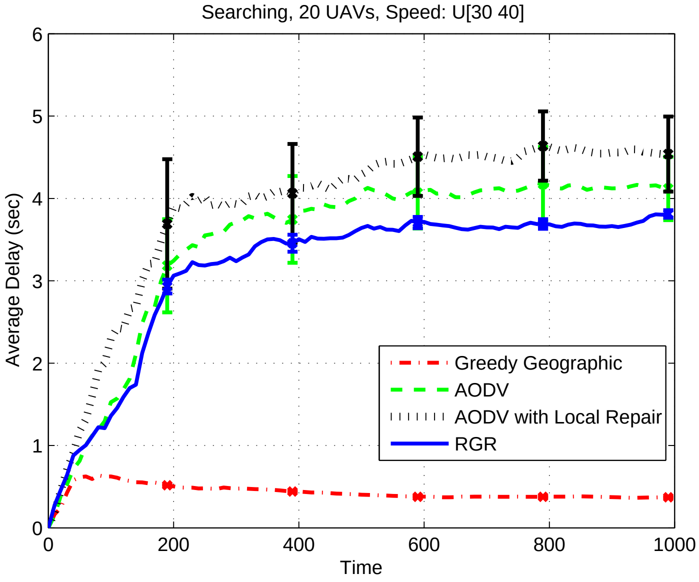

Figure 5.13: Average delay in a searching scenario of size 25 km^2 including 20 medium speed UAVs

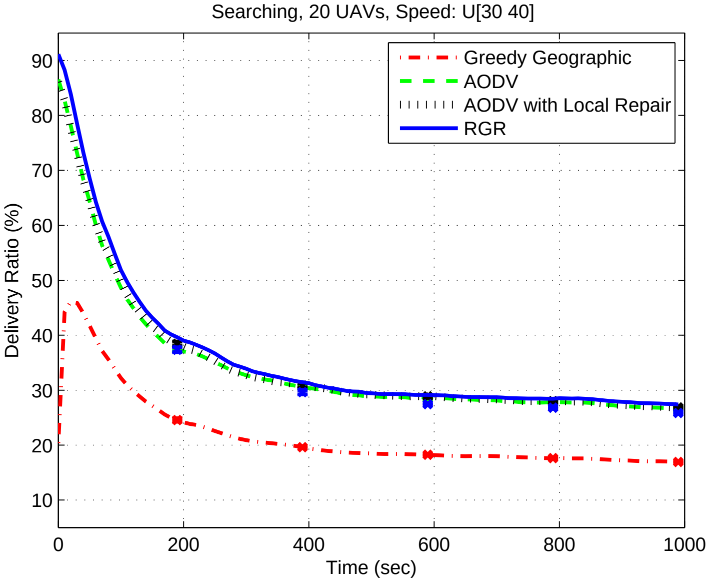

Figure 5.14: Average delivery ratio in a searching scenario of size 25 km^2 including 20 medium speed UAVs

the RREQ/RREP procedure. The improvement in delay, however, is because of the packets in intermediate nodes. In other words, the number of packets waiting for a

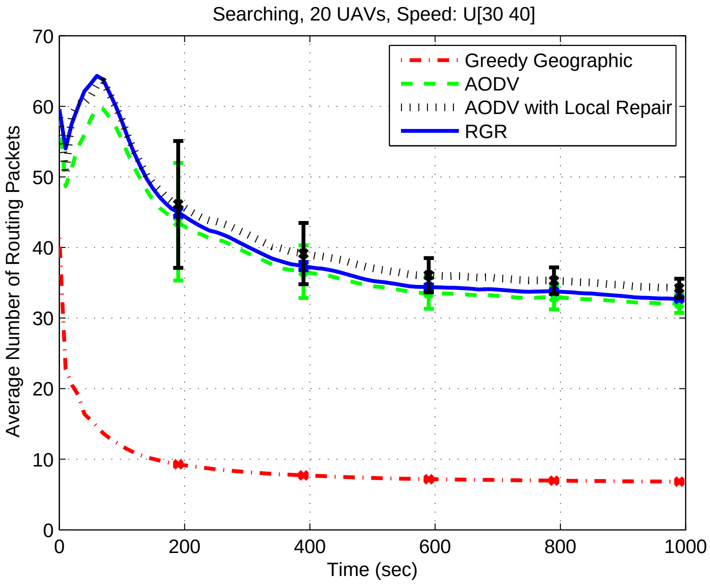

Figure 5.15: Average routing traffic in a searching scenario of size 25 km^2 including 20 medium speed UAVs

RREQ/RREP in intermediate node decreases in RGR, which cause a delay improvement in RGR compared to AODV in most scenarios. Although the same behaviour for delay of RGR versus two versions of AODV is observed in a searching scenario of size 25 km^2 including 20 high speed UAVs, as shown in Figure 5.16, the delay here is higher than lower speed scenarios. The improvement of RGR delivery ratio in higher velocities is also confirmed by Figure 5.17.

When the network has higher velocities, more reactive routes breaks. As discussed, the main difference between RGR and AODV is in the case of link breakage. In case of a link breakage, RGR transmits the packets greedily, if such a geographically closer neighbour to the destination is available. Due to the fact that reactive route is more vulnerable in higher speeds, the protocols that relies more on reactive routes will be more vulnerable. As a result, RGR that employs a mechanism independent of reactive route, can salvage more data packets on average.

Another observation in Figure 5.18, the average number of routing traffic sent by RGR is still comparable to AODV, although the packet delivery ratio of RGR has a more significant improvement. When there are more link breaks, we can expect the

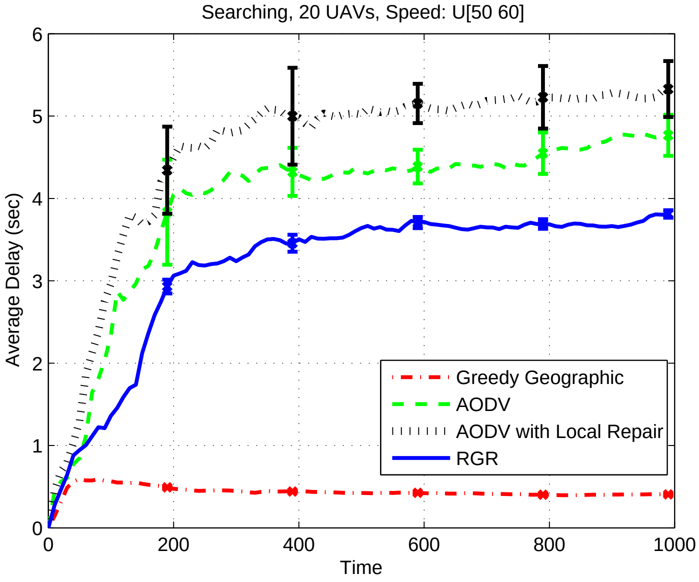

Figure 5.16: Average delay in a searching scenario of size 25 km^2 including 20 high speed UAVs

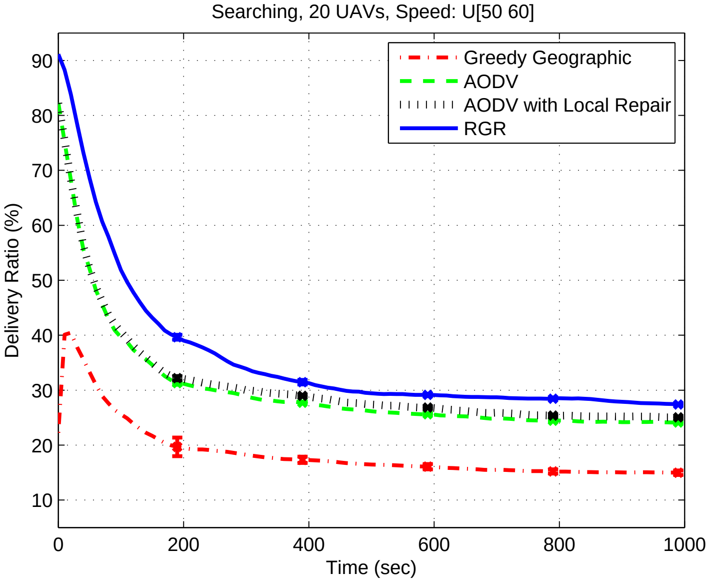

Figure 5.17: Average delivery ratio in a searching scenario of size 25 km^2 including 20 high speed UAVs

routing traffic to be different for RGR compared to different versions of AODV. In such a case, RGR and AODV without local repair, send a RERR as the link breakage is observed. Therefore, we do not expect to have a significant different in number of

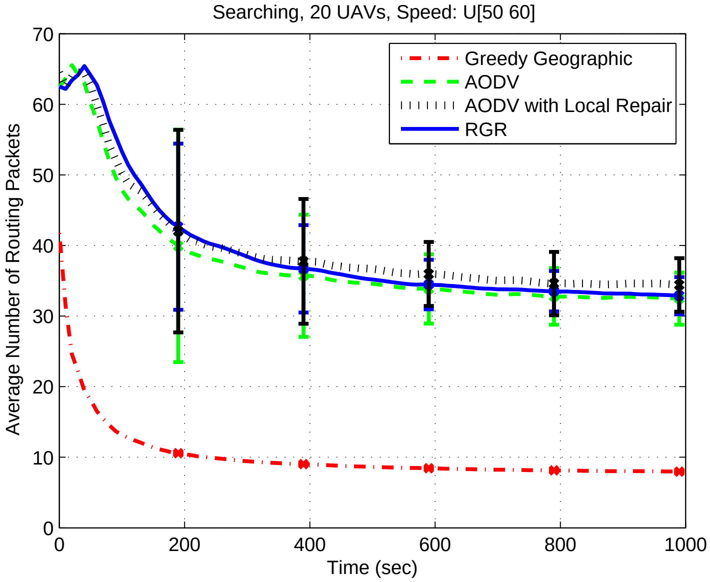

Figure 5.18: Average routing traffic in a searching scenario of size 25 km^2 including 20 high speed UAVs

control packets in RGR versus AODV, as shown in the results so far.

In order to have a better representation of the performance of RGR compared to AODV without local repair in low speed, medium speed and high speed scenarios, we evaluated the percentage of improvement in packet delivery ratio and delay. The increment in overhead is also evaluated for different scenarios. In order to derive each of those parameters, we consider the average values of packet delivery ratio, delay, and overhead over 1000 sec. In (5.7), the improvement of packet delivery ratio of RGR is defined as PDR_{imp}.

$$PDR_{imp}(\%) = \frac{APDR_{RGR} - APDR_{AODV}}{APD_{AODV}} \times 100 \tag{5.7}$$

$APDR_{RGR}$ and $APDR_{AODV}$ are average packet delivery ratio of RGR and AODV over the 1000 sec respectively. Equivalently, the percentage of improvement of RGR improvement compared to AODV is formulated in (5.8).

$$D_{imp}(\%) = \frac{D_{AODV} - D_{RGR}}{D_{AODV}} \times 100 \tag{5.8}$$

In (5.8), D_{RGR} and D_{AODV} are the average delay of RGR and AODV respectively, and D_{imp} is the delay improvement of RGR compared to AODV in percentage.

Since the imposed overhead of RGR is higher than AODV, the imposed overhead of RGR is calculated as:

$$OH_{imp}(\%) = \frac{OH_{AODV} - OH_{RGR}}{OH_{AODV}} \times 100 \tag{5.9}$$

OH_{imp} is the percentage of imposed overhead of RGR compared to AODV. Also, OH_{RGR} and OH_{AODV} are the average overhead of RGR and AODV respectively.

Figure 5.19 illustrates the packet delivery ratio improvement of RGR compared to AODV. The improvement of RGR compared to AODV increases as we go to a high speed scenario from a low speed one. As the speed of UAVs increase, we expect more switches from reactive to greedy geographic forwarding.

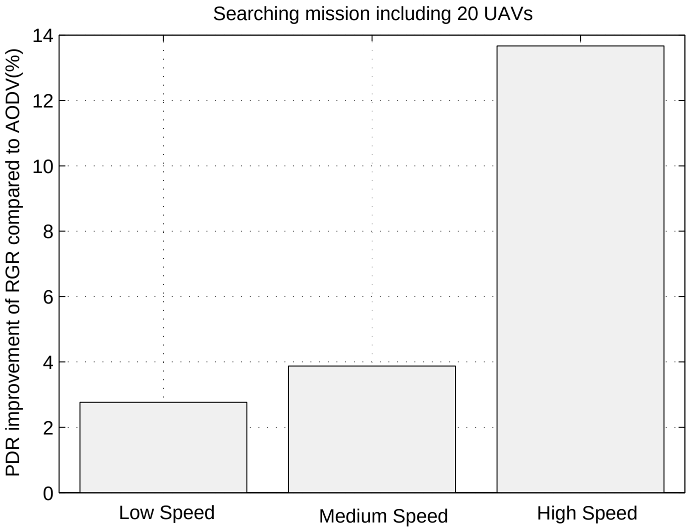

Figure 5.19: Packet delivery ratio improvement of RGR compared to AODV in different scenarios

Figure 5.20 represents the delay improvement of RGR compared to AODV without local repair. As shown, the delay improvement can be up to 20%. The other important observation is the randomness of delay behaviour compared to delivery ratio. Delay improvement has a more random nature as we see that the medium speed scenario has a lower improvement compared to low speed and high speed scenarios.

In Figure 5.21, the imposed overhead for acquiring such improvements in delay and packet delivery ratio is depicted.

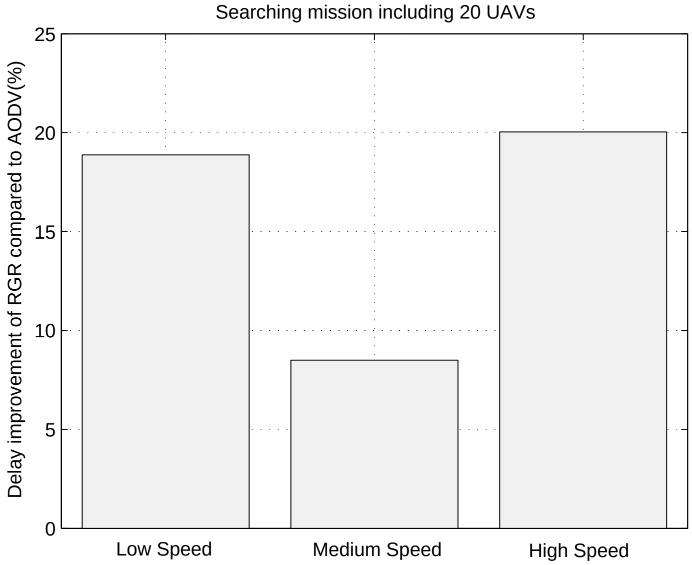

Figure 5.20: Delay improvement of RGR compared to AODV in different scenarios

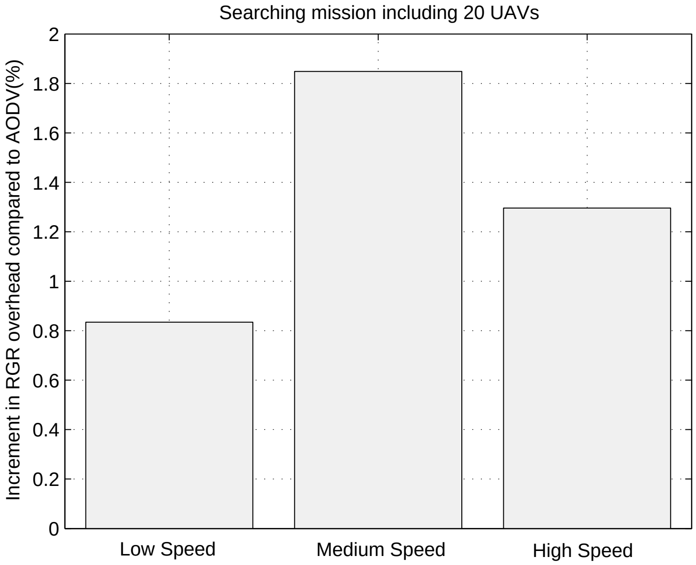

Figure 5.21: Increment in RGR overhead compared to AODV in different scenarios

In another set of simulations, we evaluated the performance of the protocols in a searching mission including 30 UAVs in an area of 25 km^2. Since the trends of the simulation figures are similar to the scenarios discussed for number of UAVs equal to 10 and 20. Simulation results for 30 UAVs in a searching mission can be found in Appendix C.

5.7 Simulation Results for a Tracking Mission

In this section, simulation results for a tracking mission are presented. We consider two different values for the number of UAVs in a tracking scenario. Unlike searching missions that include 10, 20, and 30 UAVs, we simulate two tracking scenarios including 10 and 20 UAVs. The reason for such selection is that usually the number of UAVs performed in a tracking mission is smaller than UAVs in a searching mission due to the specific requirements of the application. The other difference of tracking scenarios compared to searching scenarios is the packet inter-arrival time, which increases from 0.2 to 0.5 in tracking scenarios.

In this section, we organize the figures by simulation metrics rather than simulation scenarios (i.e. number of nodes). Three different sets of simulation results are presented in this section for a tracking scenario including 10 low speed, medium speed, and high speed UAVs. Figures 5.22, 5.23, and 5.24 present the average delay for the different protocols. The observation is that the delay of RGR is better than AODV with local repair in most scenarios, which is the result of a smaller delay imposed in intermediate nodes when a route breaks. Delay of RGR does not have a significant improvement compared to AODV without local repair. Even in some scenarios, the delay of RGR is worse.

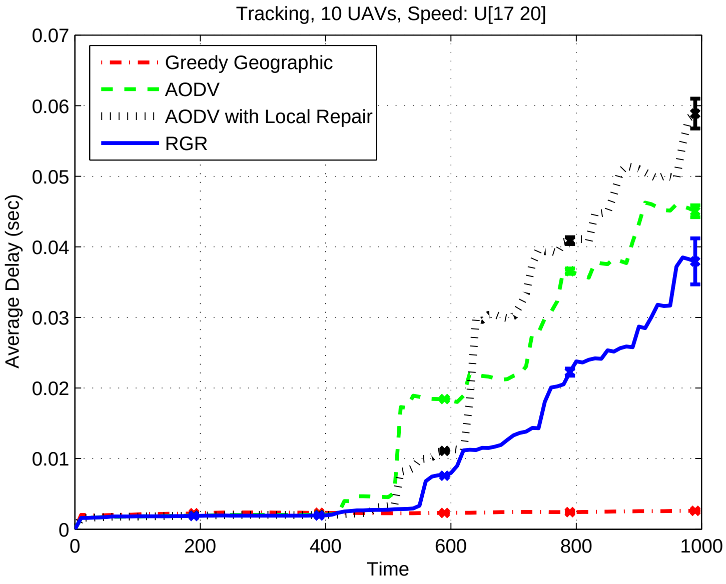

Figure 5.22: Average delay in a tracking scenario including 10 low speed UAVs

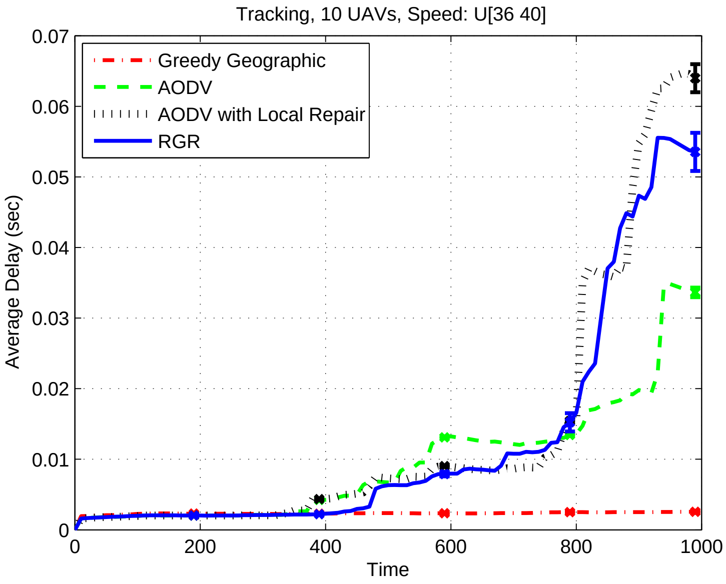

Figure 5.23: Average delay in a tracking scenario including 10 medium speed UAVs

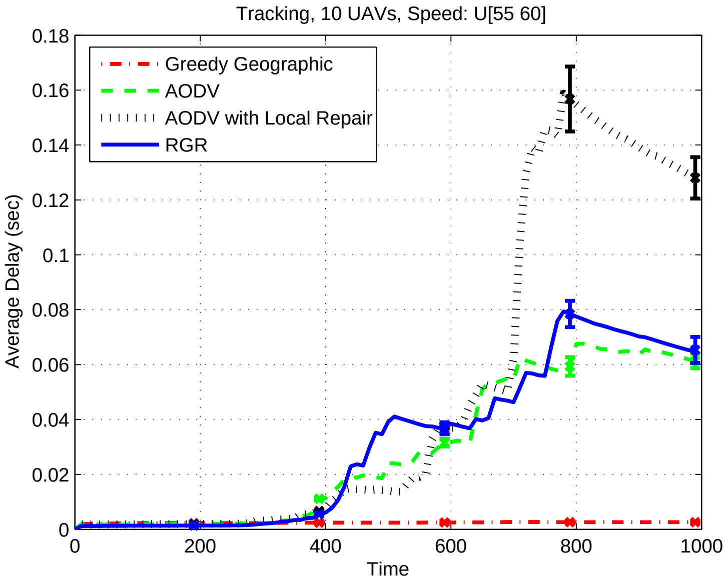

Figure 5.24: Average delay in a tracking scenario including 10 high speed UAVs

The packet delivery ratio of RGR and AODV are presented in Figures 5.25, 5.26, and 5.27 for low speed, medium speed, and high speed scenarios of the tracking mission respectively. It is shown that the performance of RGR and both versions of AODV are similar (very close) in terms of packet delivery ratio. The reason for such a phenomenon is that in tracking missions, the relative velocity of UAVs are much smaller. The fact is that UAVs' velocity interval for tracking mission is much smaller than searching missions, as characterized in Tables 5.1, and 5.2. Also, all UAVs in a tracking mission are following a single trajectory. Thus they are all approximately towards a single direction. In such a scenario, there are not many link breakage due to mobility of nodes. Based on our design, RGR is useful in scenarios where there are path interruptions and a geographically closer neighbour is available to salvage the packets buffered in intermediate nodes' queue. If the scenario is semi-static (due to low relative mobility of UAVs), such cases do not happen and the packet delivery ratio of RGR and AODV are similar. The good point is that the delivery ratio for tracking mission is high for both RGR and AODV in all three scenarios. As shown in Figures 5.25, 5.26, and 5.27, the delivery ratio for all scenarios are overlapped for all three protocols and is almost 10% better than greedy geographic forwarding. Also please note that the delivery ratio of greedy geographic forwarding in the beginning of the simulation is much lower, which is the result of the time needed for the initial flooding in geographic routing protocol in OPNET.

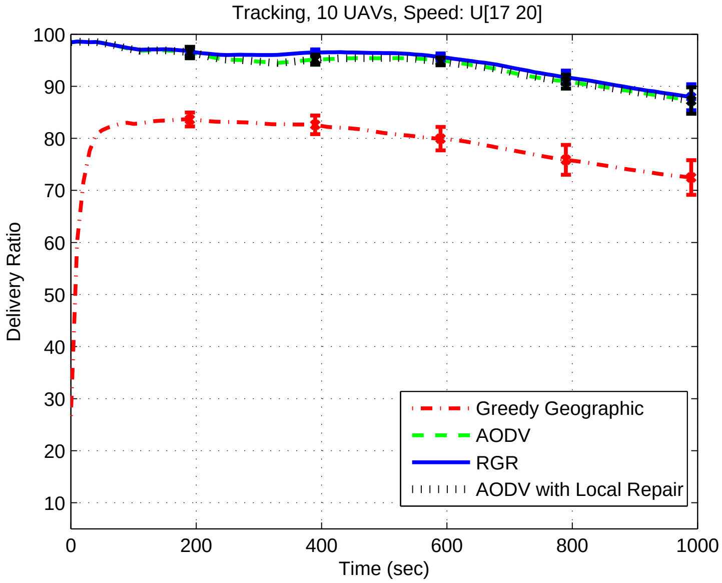

Figure 5.25: Average packet delivery ratio in a tracking scenario including 10 low speed UAVs

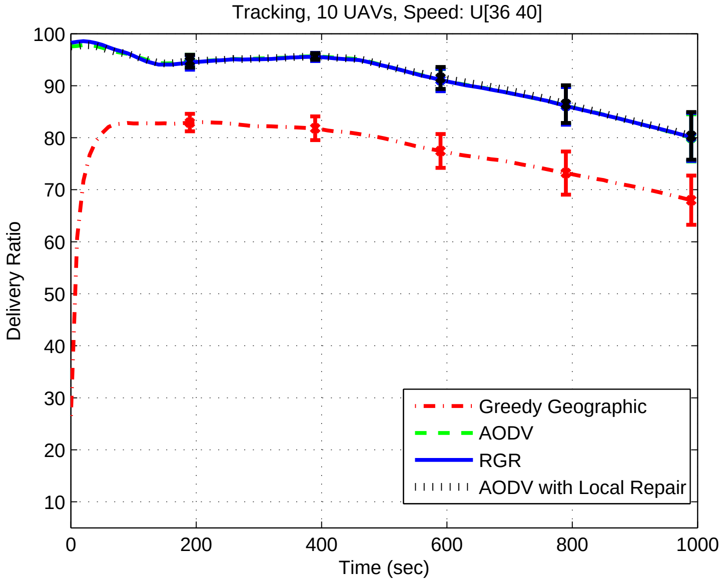

Figure 5.26: Average packet delivery ratio in a tracking scenario including 10 medium speed UAVs

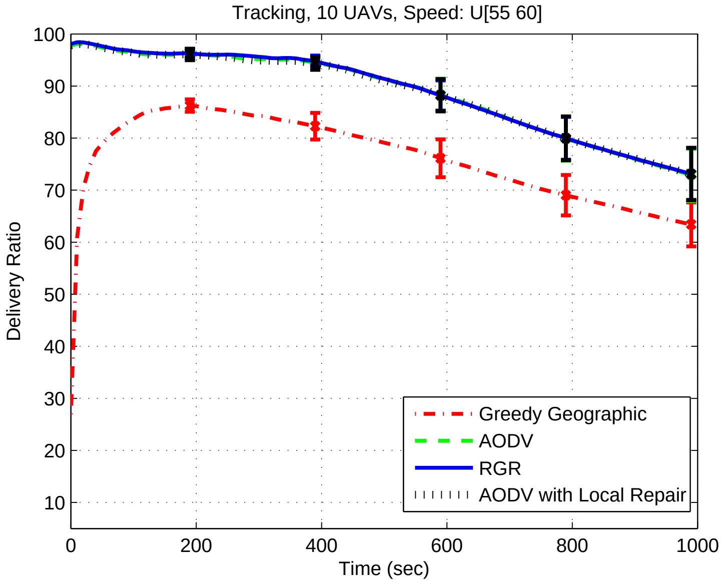

Figure 5.27: Average packet delivery ratio in a tracking scenario including 10 high speed UAVs

In terms of overhead, RGR and the two different versions of AODV have very similar behaviour (i.e. the curves are overlapped or very close). The interesting point is that for different velocities in the protocols, we have almost the same value for the average number of routing packets. Hence, we can conclude that the velocity of UAVs does not have a great impact on number of routing packets. The overhead of all four protocols in low, medium and high speed tracking scenarios are depicted in Figures 5.28, 5.29, and 5.30 respectively.

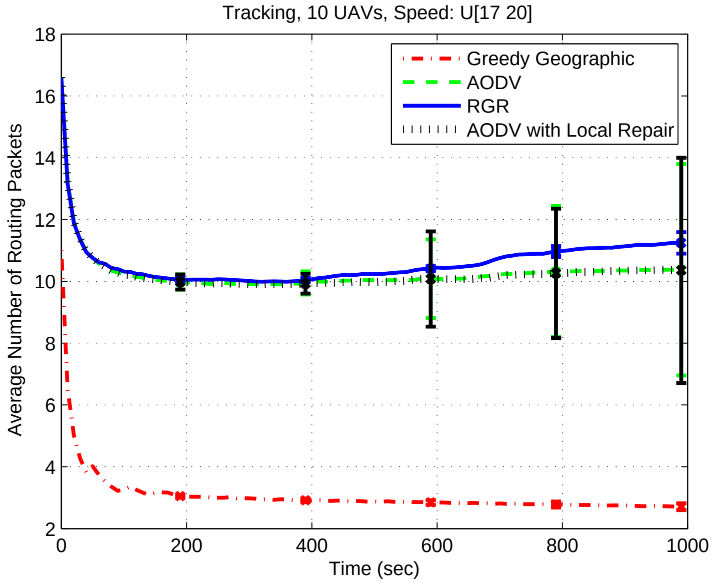

Figure 5.28: Average routing traffic in a tracking scenario including 10 low speed UAVs

Compared to the results for searching missions, different protocols in a tracking mission have a more similar behaviour, especially in terms of packet delivery ratio and overhead. One reason for such a phenomenon is that the UAVs are in each others' vicinity due to the low relative speeds in the network. The other reason is the topology of the network. Since UAVs are following a similar trajectory, the topology can be modelled by a line where all UAVs follow each other with slightly different random speeds. In such a scenario, if a reactive route breaks, the possibility of finding a new reactive route to the destination from an intermediate node via a local repair process is not high. With the same deduction, the probability of finding a geographically closer neighbour to the destination would not be considerable, compared to two dimensional searching scenarios. As a result, in tracking missions, the performance of RGR and two different versions of AODV would be more similar because in many cases when

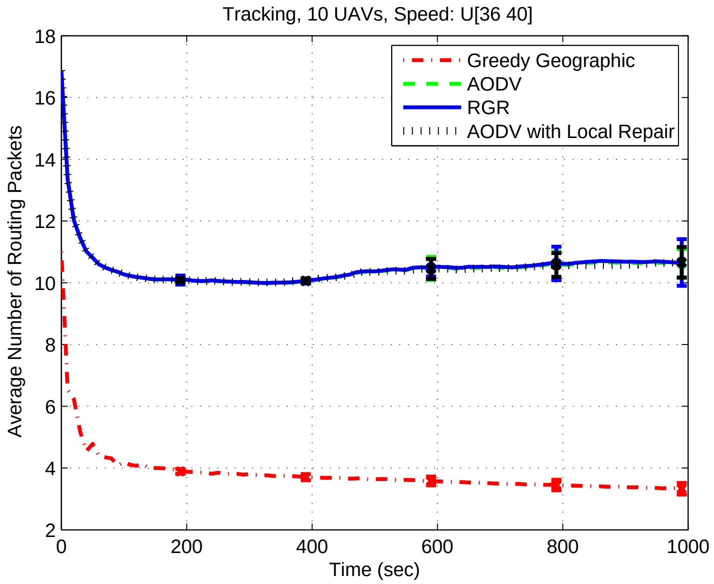

Figure 5.29: Average routing traffic in a tracking scenario including 10 medium speed UAVs

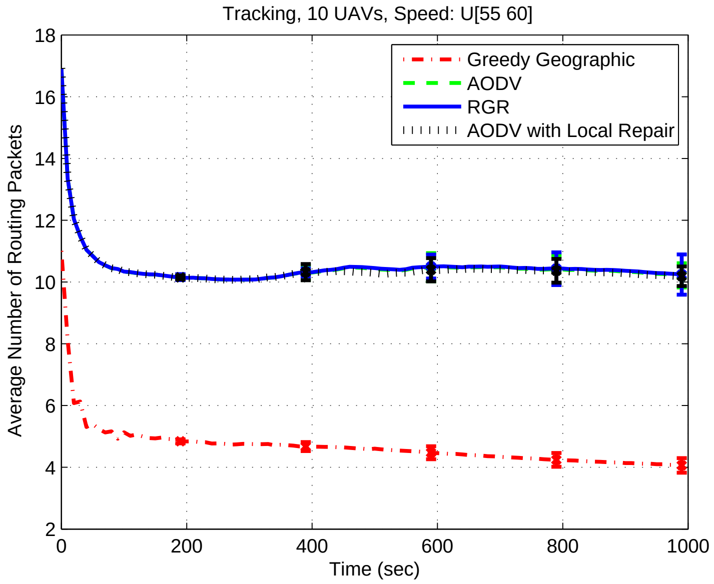

Figure 5.30: Average routing traffic in a tracking scenario including 10 high speed UAVs

a link breaks, neither local repair nor greedy geographic forwarding can salvage the packets. This phenomenon may cause an improvement compared to AODV without local repair.

We also collected simulation data for a tracking scenario with 20 UAVs. Since the results follow the same trends as shown for 10 UAVs, we do not present them in this section. Please refer to Appendix D for further information.

5.8 Number of Switches

The other statistics that we collected during the simulation of RGR is the number of switches from the reactive part to the greedy geographic forwarding. The goal in gathering statistics on the number of switches was to have a measure of the use of greedy geographic forwarding in different scenarios of tracking and searching missions. Please note that when a packet is forwarded by the greedy geographic forwarding, the value for the number of switches increased. In our OPNET implementation, when a switch to greedy geographic forwarding function takes place, the variable counting the number of switches is incremented by one. Please note that in such an implementation, if two consecutive neighbours use the greedy geographic forwarding, two switches are counted. Further information on our OPNET implementation for collecting the number of switches is presented in Appendix A.5.

The average number of switches per flow over 10 seeds for [0, 1000] sec interval for search missions is presented in Table 5.7. The first observation from the results is the rapid increment in the number of switches as we increase the number of UAVs from 10 to 30. As the number of UAVs increases in a scenario, we can expect longer routes in the network. At the same time, the possibility of route breakage increases because we have a larger number of links in the network, which makes it more likely to switch to greedy geographic forwarding.

Table 5.7: Average number of switches per source-destination flow in search missions

Scenario	10 UAVs (4 km^2)	10 UAVs (25 km^2)	20 UAVs	30 UAVs
Search U[10 20]	201.4111	217.644	818.1947	1489.3
Search U[30 40]	196.022	282.889	944.8947	1600.2
Search U[50 60]	206.172	299.4278	986.5465	1694.1

Table 5.8 represents the number of switches to greedy geographic forwarding in tracking missions. For tracking missions, we see the increment in number of switches

as we increase the number of UAVs in the scenario. Also, please note that the number of switches is smaller compared to searching scenarios. The reason is because of the specific mobility which impose a smaller relative velocity among each pair of UAVs and also the smaller amount of traffic that is shared among different UAVs in a tracking mission compared to searching scenarios.

Table 5.8: Average number of switches per source-destination flow in tracking missions

Scenario	10 UAVs	20 UAVs
Track U[17 20]	20.99	89.3452
Track U[36 40]	22.2	95.7474
Track U[55 60]	28.31	107.0342

5.9 Summary

In this chapter, the simulation results on the performance of RGR were presented for two main applications of UAANETs, searching and tracking. Due to the fact that greedy geographic forwarding has a lower average packet delivery ratio, we focused more on the comparison of RGR versus AODV. We showed that the proposed RGR protocol is comparable to AODV, with and without local repair, in most scenarios. For searching missions, we see that RGR outperforms AODV in terms of packet delivery ratio and delay. The overhead of RGR is lower than AODV with local repair, but more than AODV without local repair. Therefore, the overhead is the cost that we pay to achieve a better average end-to-end delay and packet delivery ratio.

An important observation is that the comparison of RGR with AODV highly depends on the network parameters, such as topology, velocity, and amount of traffic. The improvement of RGR compared to AODV is more clear in a scenario of 20 searching UAVs in a 25 km^2 environment compared to either 10 or 30 searching UAVs in the same area. We can conclude that the improvement of RGR is more clear when the network is not dense or very sparse. In the meantime, RGR performance is at least comparable to other mechanisms in worst case scenarios.

The other important fact about different protocols is the implementation issues.

Since we implemented the idea based on the available AODV, the proposed RGR can be easily emulated and finally implemented in realistic UAANETs. The simulation environment was OPNET, which has realistic models for network environments. Rather than the implementation issues, the current version of RGR has this potential to be modified more. Further discussion on possible steps for future work on RGR are discussed in more detail in the next chapter.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Motivated by the availability of location information in an UAANET, we have proposed a reactive-greedy-reactive mechanism for routing in density variable UAANETs. As a first step, the performance of greedy geographic forwarding was simulated in UAANETs, and also a quadratic polynomial estimation of the success probability of the greedy geographic forwarding in UAANETs was presented. The proposed quadratic function provides a mathematical representation of the success probability of greedy geographic forwarding as a function of number of hops in connected networks.

The reactive-greedy-reactive protocol is based on the notion that the protocol uses two mechanisms for data forwarding, reactive (i.e. AODV) and greedy geographic forwarding. At the beginning, the protocol sends data (and also location information) on a reactive route towards the destination. If the reactive route breaks, RGR will continue by switching to the greedy geographic part of the protocol.

The simple architecture of RGR mechanism is shown to have a better delivery ratio performance compared to AODV and greedy geographic. The packet delivery ratio improvement can be up to 5% for searching missions. The packet delivery ratio improvement happens while the delay significantly decreases for the searching scenarios (up to 1 sec). Therefore, for delay-critical applications, RGR has a significant improvement in terms of important delay metric with a slightly better delivery ratio. In the meantime, the overhead does not have a significant increment for RGR compared to AODV. The interesting point about RGR architecture is that reactive part does not require local repair and the greedy geographic part is implemented without the use of an independent location service mechanism. In other words, the reactive part uses greedy geographic to be needless of local repairs and the greedy geographic forwarding part uses the RREQ/RREP architecture of AODV as its location service mechanism. In such a design, the reactive and the greedy geogaraphic parts perform as each other's complements to improve the end-to-end delay and delivery ratio of the network.

6.2 Future Work

Our simulations illustrate that the proposed RGR mechanism has a better delay compared to AODV and greedy geographic forwarding in most scenarios. In terms of packet delivery ratio, RGR performs better than AODV only in searching scenarios. Since there are several additional suggestions to improve the mechanism, future works can be done to modify RGR mechanism to achieve better performance. In the rest of this section, we briefly review some of the modifications required for improving RGR.

The first idea to improve RGR is to include a time stamp into the control messages in order to have a measure of freshness of the location information. At the moment, the location information is obtained from the routing table in intermediate nodes in case a switch to the geographic forwarding occurs. Currently, sequence numbers are used to modify routing tables and the freshness of location information is also evaluated by the the sequence numbers already available in the protocol. In order to have a better measure of freshness of location information, we can include the time stamp. The time stamp is specifically useful in case we need to predict the future location of a node. An improved idea is to include a time stamp with all location information and to select the freshest information (i.e. the most recent location information). The lack of accurate location information will be more serious specially for unidirectional routes where the destination does not send any response to the source containing its up-to-date location information. Although our simulation results show an improvement compared to AODV even without such a time stamp, implementing the time stamp may intuitively provide a larger improvement.

The other interesting idea is to use trajectory information in order to improve the routing process. The fact is that each UAV is aware of its velocity and direction instantaneously. Also, the UAVs can maintain their past trajectories for a longer period of time (it is feasible to assume that we have enough processing speed and memory in the UAV to perform such an operation). One idea is to ask different end-point UAVs to include not only location and time but also their instantaneous trajectory information (such as velocity and direction of heading) into their messages. In this case, other nodes can access to trajectory information of the destination. This information can be used to improve the estimate of an intermediate node about the destination location if a switch to the greedy geographic mechanism is required. If we know the current location, velocity and direction of heading of a UAV, a relatively reliable estimation of the future location of the UAV can be achieved. The fresher the velocity information is, the more reliable an estimate of the future location can be achieved.

As explained in the previous paragraph, the trajectory information of the end-points can be helpful for the cases that a switch to the greedy geographic forwarding occurs. One more item of future work is to use trajectory information of the nodes to improve the reactive source-destination route as well. Right now, the reactive route is based upon hop count, which is originally available in AODV. In the current design, if two routes to the destination are found by the source, the one with the smallest hop count is used. We can propose a measure of path stability based on the instantaneous trajectory information of the UAVs. One good measure is the inner product of the velocity vectors of the UAVs. Rather than considering the minimum hop count, we can propose a mechanism to select the most stable routes (i.e. routes that consist of hops that are predicted to exist for long time, based on the current relative velocity of the nodes). Such a design can also be used to provide QoS routing to higher priority applications.

Another area of future research is to address the RERR handling in RGR. Right now, as soon as a source learns about a link breakage, it invalidates the route and, presuming it has more data, sends out a RREQ to globally fix the route. Instead, we could do one or both of the following: 1) While we do not have a new route, we continue using the old route, relying on the greedy geographic forwarding to deliver the packets. 2) Instead of aggressively fixing the route, we could keep using it and fix it only after some timeout and/or a fixed number of RERR messages have been received. Again, we would rely on the greedy geographic forwarding. Such a future research may lead to smaller average overhead.

In an UAANET, sometimes several clusters of UAVs may be available in some situation (e.g. in a search mission). The proposed RGR routing does not have an alternative communication architecture for such cases. One useful approach in sparsely connected networks is to combine store-carry-forward (SCF) mechanism with RGR. SCF has already been proposed as a routing/forwarding mechanism in partially connected networks [57]. Combining SCF with geographic routing can be a potential suggestion not only to prevent a blind mechanism such as face routing but also to use all the available potentials of the networking characteristics including high mobility and availability of geographic locations [45].

Another direction of future work on the materials presented in Chapter 3 is to analytically derive upper and lower performance bounds for greedy geographic forwarding and compare them with the Monte Carlo simulation results. One other idea can be to evaluate beacon-less geographic routing protocols. Beacon-less mechanisms avoid the need to periodically exchange hello messages, but limit the greedy forwarding range further. Similar to our study, it would be interesting to explore what network densities would be required for beacon-less geographic forwarding to be successful. Also, one may think of generalizing face routing or proposing other alternatives for three dimensional environments. Finally, one step for future work is to implement RGR in a realistic network environment to test its performance and to observe the functionality in a realistic implementation and compare it with OPNET results.

List of References

[1] E. Perkins, C. Belding-Royer and S. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing RFC3561.” http://www.ietf.org/rfc/rfc3561.txt, [Accessed: Febuary 2011].

[2] R. Shirani, M. St-Hilaire, T. Kunz, Y. Zhou, J. Li, and L. Lamont, “The performance of greedy geographic forwarding in unmanned aeronautical ad-hoc networks,” in *9th Annual Conference on Communication Networks and Services Research Conference (CNSR 2011)*, pp. 161 –166, May 2011.

[3] B. Karp and H. T. Kung, “GPSR: greedy perimeter stateless routing for wireless networks,” in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 243–254, ACM, 2000.

[4] M. Mauve, A. Widmer, and H. Hartenstein, “A survey on position-based routing in mobile ad hoc networks,” *Network, IEEE*, vol. 15, pp. 30 –39, Nov/Dec 2001.

[5] X. Hong, K. Xu, and M. Gerla, “Scalable routing protocols for mobile ad hoc networks,” *Network, IEEE*, vol. 16, pp. 11 –21, jul/aug 2002.

[6] L. Qin and T. Kunz, “Survey on mobile ad hoc network routing protocols and cross-layer design,” tech. rep., Carleton University, August 2004.

[7] “Unmanned aerial vehicles classification.” http://www.vectorsite.net/twuav.html, [Accessed: March 2010].

[8] “Tier classification of uavs.” http://www.airpower.maxwell.af.mil/airchronicles/cc/uav.html, [Accessed: March 2010].

[9] T. Pham, G. Cirincione, D. Verma, and G. Pearson, “Intelligence, surveillance, and reconnaissance fusion for coalition operations,” in *Information Fusion, 2008 11th International Conference on*, pp. 1–8, July 2008.

[10] J. Veverka and M. Campbell, “Operator decision modeling for intelligence, surveillance and reconnaissance type missions,” in *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, vol. 1, pp. 754–759 Vol. 1, Oct. 2005.

[11] “Search and rescue missions.” http://www.justaeroworks.com/pages/JAW.08SRM.html, [Accessed: September 2009].

[12] E. Frew and T. Brown, "Airborne communication networks for small unmanned aircraft systems," *Proceedings of the IEEE*, vol. 96, pp. 2008–2027, Dec. 2008.

[13] D. Lyon, "A military perspective on small unmanned aerial vehicles," *Instrumentation Measurement Magazine, IEEE*, vol. 7, pp. 27–31, Sept. 2004.

[14] E. Sakhaee, A. Jamalipour, and N. Kato, "Aeronautical ad hoc networks," in *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, vol. 1, pp. 246–251, April 2006.

[15] E. Sakhaee and A. Jamalipour, "Stable clustering and communications in pseudolinear highly mobile ad hoc networks," *Vehicular Technology, IEEE Transactions on*, vol. 57, pp. 3769–3777, Nov. 2008.

[16] E. Sakhaee and A. Jamalipour, "The global in-flight internet," *Selected Areas in Communications, IEEE Journal on*, vol. 24, pp. 1748–1757, Sept. 2006.

[17] D. Henkel, C. Dixon, J. Elston, and T. X. Brown, "A reliable sensor data collection network using unmanned aircraft," in *REALMAN '06: Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, pp. 125–127, ACM, 2006.

[18] A. Erman, L. Hoesel, P. Havinga, and J. Wu, "Enabling mobility in heterogeneous wireless sensor networks cooperating with UAVs for mission-critical management," *Wireless Communications, IEEE*, vol. 15, pp. 38–46, December 2008.

[19] N. Li, H. Liu, E. Earon, C. D. Fulford, R. Huq, and C. A. Rabbath, "Multiple UAVs autonomous mission implementation on COTS autopilots and experimental results," in *AIAA Guidance, Navigation and Control Conference*, Aug 2009.

[20] J. S. Jang and D. Liccardo, "Small UAV automation using MEMS," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 22, pp. 30–34, May 2007.

[21] J. How, C. Fraser, K. Kulling, L. Bertuccelli, O. Toupet, L. Brunet, A. Bachrach, and N. Roy, "Increasing autonomy of UAVs," *Robotics & Automation Magazine, IEEE*, vol. 16, pp. 43–51, June 2009.

[22] J. How, B. Bethke, A. Frank, D. Dale, and J. Vian, "Real-time indoor autonomous vehicle test environment," *Control Systems Magazine, IEEE*, vol. 28, pp. 51–64, April 2008.

[23] X. C. Ding, M. Powers, M. Egerstedt, S. yih Young, and T. Balch, "Executive decision support," *Robotics & Automation Magazine, IEEE*, vol. 16, pp. 73–81, June 2009.

[24] Y. Ben-Asher, S. Feldman, P. Gurfil, and M. Feldman, "Distributed decision and control for cooperative UAVs using ad hoc communication," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 3, pp. 511 – 16, 2008.

[25] B. Bethke, M. Valenti, and J. How, "UAV task assignment," *Robotics & Automation Magazine, IEEE*, vol. 15, pp. 39–44, March 2008.

[26] J. Tisdale, Z. Kim, and J. Hedrick, "Autonomous UAV path planning and estimation," *Robotics & Automation Magazine, IEEE*, vol. 16, pp. 35–42, June 2009.

[27] D. Cole, A. Goktogan, P. Thompson, and S. Sukkarieh, "Mapping and tracking," *Robotics & Automation Magazine, IEEE*, vol. 16, pp. 22–34, June 2009.

[28] M. Likhachev and A. Stentz, "Path clearance," *Robotics & Automation Magazine, IEEE*, vol. 16, pp. 62–72, June 2009.

[29] Y. Jin, Y. Liao, A. Minai, and M. Polycarpou, "Balancing search and target response in cooperative unmanned aerial vehicle (uav) teams," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 36, pp. 571–587, June 2005.

[30] D. Pack, P. DeLima, G. Toussaint, and G. York, "Cooperative control of UAVs for localization of intermittently emitting mobile targets," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 39, pp. 959–970, Aug. 2009.

[31] C.-M. Cheng, P.-H. Hsiao, H. Kung, and D. Vlah, "Performance measurement of 802.11a wireless links from UAV to ground nodes with various antenna orientations," in *Computer Communications and Networks, 2006. ICCCN 2006. Proceedings.15th International Conference on*, pp. 303–308, Oct. 2006.

[32] T. Coelho, R. Macedo, P. Carvalhal, J. Afonso, L. Silva, H. Almeida, M. Ferreiral, and C. Santos, "A fly-by-wireless UAV platform based on a flexible and distributed system architecture," in *Industrial Technology, 2006. ICIT 2006. IEEE International Conference on*, pp. 2359–2364, Dec. 2006.

[33] T. X. Brown, B. Argrow, C. Dixon, and S. Doshi, "Ad hoc UAV ground network (AUGNet)," in *AIAA Guidance, Navigation and Control Conference*, September 2004.

[34] "Global security, rq-4 global hawk (tier ii+hae UAV)." http://www.globalsecurity.org/intell/systems/global_hawk.html, [Accessed: September 2009].

[35] D. Medina, F. Hoffmann, S. Ayaz, and C.-H. Rokitansky, "Feasibility of an aeronautical mobile ad hoc network over the north atlantic corridor," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on*, pp. 109–116, June 2008.

[36] M. Jensen, "Alternative antenna polarization schemes for satellite-handset links including operator tissue," in *Antennas and Propagation Society International Symposium, 1997. IEEE., 1997 Digest*, vol. 2, pp. 1336–1339 vol.2, Jul 1997.

[37] G. Codispoti, M. Lisi, and V. Santachiara, "X-band SAR active antenna design for small satellite applications," vol. 1, pp. 666 – 9, 1995.

[38] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad hoc network research,” *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483–502, 2002.

[39] G. Lin, N. G, and R. Rajaraman, “Mobility models for ad hoc network simulation,” in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 1, pp. 454–463, March 2004.

[40] T. X. Brown, S. Doshi, S. Jadhav, and J. Himmelstein, “Test bed for a wireless network on small UAVs,” in *AIAA 3rd “Unmanned Unlimited” Technical Conference*, September 2004.

[41] J. Park and S. Sahni, “Power assignment for symmetric communication in wireless sensor networks,” in *Computers and Communications, 2006. ISCC ’06. Proceedings. 11th IEEE Symposium on*, pp. 591–596, June 2006.

[42] Y. Johnson, D. Hu and D. Maltz, “ The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4,” 2011. http://tools.ietf.org/html/rfc4728, [Accessed: April 2011].

[43] N. Moghim, F. Hendessi, and N. Movehhedinia, “An improvement on ad-hoc wireless network routing based on AODV,” in *ICCS ’02: Proceedings of the The 8th International Conference on Communication Systems*, pp. 1068–1070, IEEE Computer Society, 2002.

[44] C. Mbarushimana and A. Shahrabi, “Comparative study of reactive and proactive routing protocols performance in mobile ad hoc networks,” in *AINAW ’07: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*, pp. 679–684, IEEE Computer Society, 2007.

[45] E. Kuiper and S. Nadjm-Tehrani, “Geographical routing in intermittently connected ad hoc networks,” in *AINAW ’08: Proceedings of the 22nd International Conference on Advanced Information Networking and Applications - Workshops*, pp. 1690–1695, IEEE Computer Society, 2008.

[46] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, “On the pitfalls of geographic face routing,” in *DIALM-POMC ’05: Proceedings of the 2005 joint workshop on Foundations of mobile computing*, pp. 34–43, ACM, 2005.

[47] X. Guan, *Face Routing in Wireless Ad-Hoc Networks.* PhD thesis, University of Toronto, 2009. http://www.cs.toronto.edu/guan/phdthesis.pdf, [Accessed: August 2010].

[48] R. Flury and R. Wattenhofer, “Randomized 3d geographic routing,” in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pp. 834 –842, april 2008.

[49] A. Nayak and I. Stojmenovic, *Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication.* Wiley, 2010.

[50] D. Son, A. Helmy, and B. Krishnamachari, “The effect of mobility-induced location errors on geographic routing in mobile ad hoc sensor networks: analysis and improvement using mobility prediction,” *Mobile Computing, IEEE Transactions on*, vol. 3, pp. 233 – 245, july-aug. 2004.

[51] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, “Geographic routing without location information,” in *MobiCom ’03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pp. 96–108, ACM, 2003.

[52] S. Shah and K. Nahrstedt, “Predictive location-based QoS routing in mobile ad hoc networks,” in *Communications, 2002. ICC 2002. IEEE International Conference on*, vol. 2, pp. 1022 – 1027 vol.2, 2002.

[53] S. Mousavi, H. Rabiee, M. Moshref, and A. Dabirmoghaddam, “Model based adaptive mobility prediction in mobile ad-hoc networks,” in *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, pp. 1713 –1716, 21-25 2007.

[54] D. Gu, G. Pei, H. Ly, M. Gerla, B. Zhang, and X. Hong, “Uav aided intelligent routing for ad-hoc wireless network in single-area theater,” in *Wireless Communications and Networking Conference, 2000. WCNC. 2000 IEEE*, vol. 3, pp. 1220 –1225 vol.3, 2000.

[55] K. Xu, X. Hong, M. Gerla, H. Ly, and D. Gu, “Landmark routing in large wireless battlefield networks using uavs,” in *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE*, vol. 1, pp. 230 – 234 vol.1, 2001.

[56] J. Kurhinen and J. Janatuinen, “Geographical routing for delay tolerant encounter networks,” in *Computers and Communications, 2007. ISCC 2007. 12th IEEE Symposium on*, pp. 463–467, July 2007.

[57] E. Kuiper and S. Nadjm-Tehrani, “Geographical routing with location service in intermittently connected manets,” *Vehicular Technology, IEEE Transactions on*, vol. 60, pp. 592 –604, feb. 2011.

[58] H. Li, B. Yang, C. Chen, and X. Guan, “Connectivity of aeronautical ad hoc networks,” in *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, pp. 1788 – 1792, dec. 2010.

[59] X. Ma, M.-T. Sun, G. Zhao, and X. Liu, “An efficient path pruning algorithm for geographical routing in wireless networks,” *Vehicular Technology, IEEE Transactions on*, vol. 57, pp. 2474–2488, July 2008.

[60] S. Ruhrup, H. Kalosha, A. Nayak, and I. Stojmenovic, “Message-efficient beaconless georouting with guaranteed delivery in wireless sensor, ad hoc, and actuator networks,” *Networking, IEEE/ACM Transactions on*, vol. 18, pp. 95–108, Feb. 2010.

[61] F. Theoleyre, E. Schiller, and A. Duda, “Efficient greedy geographical non-planar routing with reactive deflection,” in *Communications, 2009. ICC ’09. IEEE International Conference on*, pp. 1 –5, june 2009.

[62] E. W. Frew and T. X. Brown, “Networking issues for small unmanned aircraft systems,” *J. Intell. Robotics Syst.*, vol. 54, no. 1-3, pp. 21–37, 2009.

[63] T. Melodia, D. Pompili, and I. Akyildiz, “On the interdependence of distributed topology control and geographical routing in ad hoc and sensor networks,” *Selected Areas in Communications, IEEE Journal on*, vol. 23, pp. 520–532, March 2005.

[64] X. Ma, M.-T. Sun, X. Liu, and G. Zhao, “Improving geographical routing for wireless networks with an efficient path pruning algorithm,” in *Sensor and Ad Hoc Communications and Networks, 2006. SECON ’06. 2006 3rd Annual IEEE Communications Society on*, vol. 1, pp. 246–255, Sept. 2006.

[65] M. Zorzi and R. Rao, “Multihop performance of geographic random forwarding for ad hoc and sensor networks,” in *Global Telecommunications Conference, 2003. GLOBECOM ’03. IEEE*, vol. 7, pp. 3948–3952, Dec. 2003.

[66] M. Zorzi and R. Rao, “Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance,” *Mobile Computing, IEEE Transactions on*, vol. 2, pp. 349–365, Oct.-Dec. 2003.

[67] E. Schiller, P. Starzetz, F. Theoleyre, and A. Duda, “Properties of greedy geographical routing in spontaneous wireless mesh networks,” in *Global Telecommunications Conference, 2007. GLOBECOM ’07. IEEE*, pp. 4941–4945, Nov. 2007.

[68] “Matlab curve fitting toolbox.” http://www.mathworks.com/help/toolbox/curvefit/, [Accessed: August 2011].

[69] D. Chen and P. Varshney, “A survey of void handling techniques for geographic routing in wireless networks,” *Communications Surveys Tutorials, IEEE*, vol. 9, pp. 50 –67, quarter 2007.

[70] “OPNET, Application and Network Performance.” http://www.opnet.com, [Accessed: August 2011].

[71] E. Kuiper and S. Nadjm-Tehrani, “Mobility models for UAV group reconnaissance applications,” in *Wireless and Mobile Communications, 2006. ICWMC ’06. International Conference on*, pp. 33–33, July 2006.

[72] M. Zayene, N. Tabbane, and R. Elidoudi, “Performance evaluation of greedy perimeter stateless routing protocol in ad hoc networks,” in *Computer Sciences and Convergence Information Technology, 2009. ICCIT ’09. Fourth International Conference on*, pp. 907 –912, Nov. 2009.

[73] I. Abualhaol and M. Matalgah, “Performance analysis of multi-carrier relay-based uav network over fading channels,” in *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, pp. 1811 –1815, Dec. 2010.

[74] A. Alshabatat and L. Dong, "Performance analysis of mobile ad hoc unmanned aerial vehicle communication networks with directional antennas," in *International Journal of Aerospace Engineering*, Hindawi, 2010.

[75] I. Chatzigiannakis, E. Kaltsa, and S. Nikoletseas, "On the effect of user mobility and density on the performance of ad-hoc mobile networks," in *Networks, 2004. (ICON 2004). Proceedings. 12th IEEE International Conference on*, vol. 1, pp. 336 – 341 vol.1, nov. 2004.

Appendix A

OPNET Implementation

In the following section, we review the OPNET implementation of RGR. Due to the fact that RGR is based on AODV, we do not discuss the parts that are in common and already documented by OPNET. The goal is to discuss the necessary changes that were made to have RGR. For further information on OPNET implementation of AODV, please refer to [70].

A.1 Adding Location Information in Header Files

In order to define the required structure for AODV, the following changes should be made in the header files. In **AodvT_Route_Entry**, **AodvT_Forward_Request_Entry** two new double variables for (x,y) are defined. For each of the above structures, the following two lines are added.

double x_value;
double y_value;

After defining the required route entries in rgr.h, we also need to add entries in the RREQ and RREP structures. In order to do that in **aodv_pkt_support.h**, the following changes are applied.

For RREQ/RREP option, **AodvT_Rreq** and **AodvT_ Rreq**, two double variables for (x,y) are added. Then, in the function related to RREQ/RREP in the function block, we have the necessary data structure for sending location information in RREQ/RREP architecture.

Also, in **aodv_ptypes**, the prototypes of AODV have been defined. We need to add location information in the functions (in process model) that requires that data. These functions are:

aodv_route_table_entry_create,
aodv_request_table_forward_rreq_insert,
aodv_pkt_support_rreq_option_create,
aodv_pkt_support_rrep_option_create.

As an example, **aodv_route_table_entry_create** will be changed as follows. This will be the same for other functions as well. The last two double variables are the ones related to location information (x,y).

aodv_route_table_entry_create (AodvT_Route_Table*, InetT_Address, InetT_Subnet_Mask, InetT_Address, IpT_Port_Info, int, int, double, double /* RGR */, double /*RGR*/)

A.2 Switching to Greedy Geographic Forwarding

The process of switching to greedy geographic forwarding happens in application packet arrival function of AODV. The idea is that a switch to greedy geographic forwarding happens when an application packet arrives and no reactive route is available to destination. The function for handling arrival of an application packet is **aodv_rte_app_pkt_arrival_handle**. Thus, the jump function to greedy geographic forwarding (**aodv_to_greedy_jump_towards_dest**) is called by the application packet arrival function, after finding the reactive route is not available. In such a case, the statistics of the number of switches are also updated as explained in Appendix A.5.

In **aodv_to_greedy_jump_towards_dest** function, firstly the destination location is extracted. Also, the application packets are queued while the closest neighbour to destination is found. Then, the neighbour table is achieved by a pointer. The number of neighbours and a list of available neighbours is obtained through two different functions. By accessing neighbour's location information in neighbour table, the closest neighbour to destination is found. The data packets then are sent to MAC to be forwarded to that closest neighbour.

A.3 Lifetime

The **Lifetime** field is firstly initialized by **ACTIVE_ROUTE_TIMEOUT** and then it is modified by different control packets. When a route is used, the **Lifetime** field of all the nodes on the route is modified to be no less than the current time plus **ACTIVE_ROUTE_TIMEOUT**.

After reception of a RREQ, the **Lifetime** of the reverse route to the source is set to

Lifetime=aodv_rte_max_find(Existing_Lifetime, Min_Lifetime)

Where the **Existing_Lifetime** is:

Existing_lifetime = route_entry_ptr-route_expiry_time - op_sim_time ();

And **Min_Lifetime** is defined as:

Min_Lifetime = op_sim_time ()+(2.0 * net_traversal_time) -
(2.0 * rreq_option_ptr-hop_count * node_traversal_time);

The **op_sim_time** provides the current time in OPNET simulator. After a route request has been sent out, a route should be received within the **net_traversal_time**. Also, **node_traversal_time** is a conservative estimate of the average one hop traversal time for packets and should include queuing delays, interrupt processing times and transfer times.

An active (valid) route is a route that can be used for data forwarding, and is recognized by a valid mark in its routing table entry. An expired route, denoted by a state of invalid in the routing table entry is called an invalid route. Note that **Lifetime** for an active route is the expiry time, and for an invalid route is the deletion time. For an invalid route, the **Lifetime** is defined as current time plus delete period.

A.4 Create Functions and Arrival Functions in RGR

Aodv_request_table_create and **aodv_route_table_create** functions are used to create request tables and route tables. These two functions are available in external C files and we do not change them for implementing RGR. The functions for creating a packet are defined in the function block of the process model. These functions include:

1) **aodv_route_table_entry_create**: Create a new route entry

2) **aodv_pkt_support_rreq_option_create**: Create a route request option

3) **aodv_pkt_support_pkt_create**: To set different options in AODV packets (such as RREQ, RREP, RERR)

4) **aodv_pkt_support_rrep_option_create**: Create a route reply option

In the **create** functions, only the node needs to add location information for greedy geographic mode as it is explained in Appendix A.1.

Aodv_rte_pkt_arrival_handle is used to handle all packet arrivals in OPNET implementation of AODV. The **aodv_rte_pkt_arrival_handle** function then is based on the type of the packet calls:

1) **aodv_rte_app_pkt_arrival_handle**: To handle an application packet received from a higher layer.

2) **aodv_rte_rreq_pkt_arrival_handle**: To handle the reception of a RREQ.

3) **aodv_rte_rrep_pkt_arrival_handle**: To handle the reception of a RREP.

4) **aodv_rte_rrep_hello_pkt_arrival_handle**: To handle the arrival of a hello packet. If there is a valid a route to this neighbour, the route's sequence number is updated by using hello's sequence number.

A.5 Statistics of the Number of Switches

When **aodv_rte_app_pkt_arrival_handle** function is called, the process of switching to greedy geographic forwarding takes place. Thus, the statistics of such a switch are saved. The time of such an event is also automatically saved in OPNET. The following statistics are written:

op_stat_write (local_stat_handle_ptr->total_switches_shandle, 1);

op_stat_write (global_stat_handle_ptr->total_switches_global_shandle, 1);

In order to be able to write the statistics, the desired variable for global statistics should already be registered in **aodv_support.c**. The following line registers the global handle in a function called **aodv_support_global_stat_handles_obtain** of **aodv_support.c**.

For the global statistics handle, we have:

stat_handle_ptr ->total_switches_global_shandle = op_stat_reg ("AODV.Total Switches", OPC_STAT_INDEX_NONE, OPC_STAT_GLOBAL);

The local statistics handle will be:

local_stat_handle_ptr->total_switches_shandle = op_stat_reg ("AODV.Total Switches", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);

Also because statistics are defined as a C structure, in the header file **rgr.h**, the required variable for statistics is defined in **AodvT_Local_Stathandles** structure as follows:

```
typedef struct
{
Stathandle total_switches_shandle;
} AodvT_Local_Stathandles;
```

The desired statistics should also be defined in the node model. The wireless node model should be enhanced to include the statistics of number of switches. The reason is that the statistics are collected in each node, which requires the node to know about the existence of the newly defined statistics. In **mante_station_adv** in Interfaces/Node Statistics, the statistics of number of switches are added.

Appendix B

10 Searching UAVs in a 25 km^2 Area

In this Appendix, the figures for scenarios including 10 searching UAVs in a 25 km^2 environment are depicted. The delivery ratio of RGR and both versions of AODV are almost equal (the curves matches each other), which confirms our intuition about the increment of the probability of an isolated network. In such a case with a partitioned network, all protocols will fail and RGR cannot show its superiority.

Another interesting fact about these scenarios is the overhead of RGR which is higher at the beginning of the simulation and will tend to AODV as time passes. Although the difference is not significant, the number of control packets in RGR and AODV with local repair are close to each other and higher than AODV without local repair.

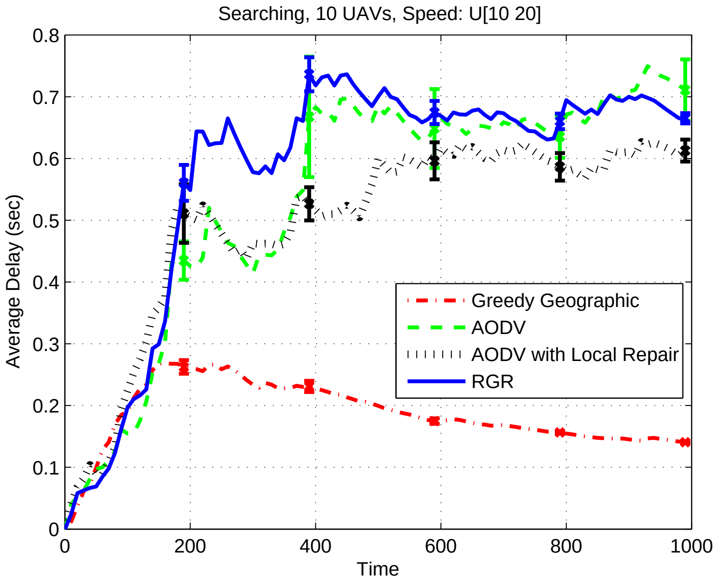

Figure B.1: Average delay in a searching scenario of size 25 km^2 including 10 low speed UAVs

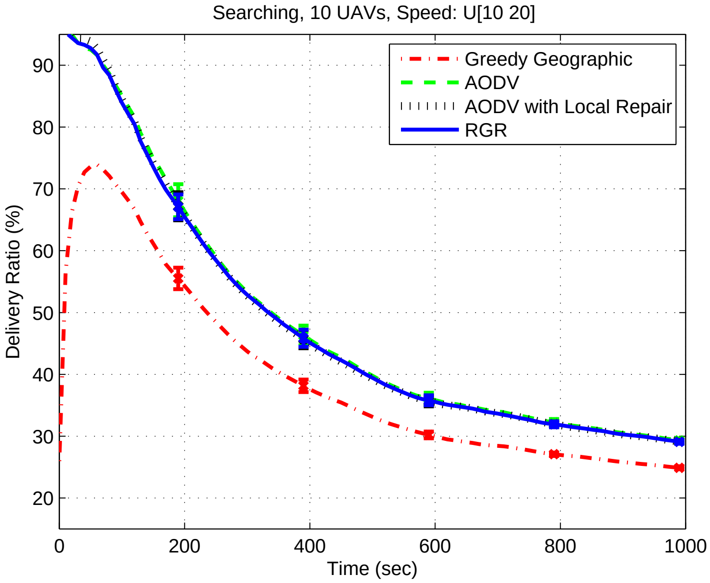

Figure B.2: Average delivery ratio in a searching scenario of size 25 km^2 including 10 low speed UAVs

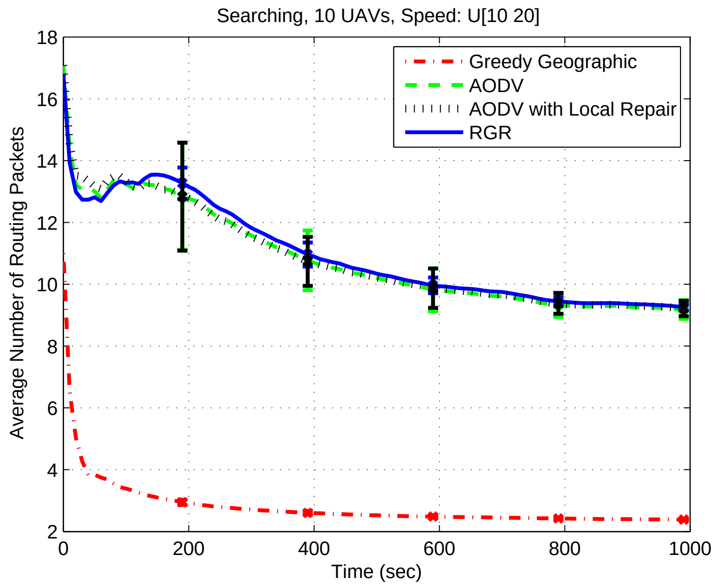

Figure B.3: Average routing traffic in a searching scenario of size 25 km^2 including 10 low speed UAVs

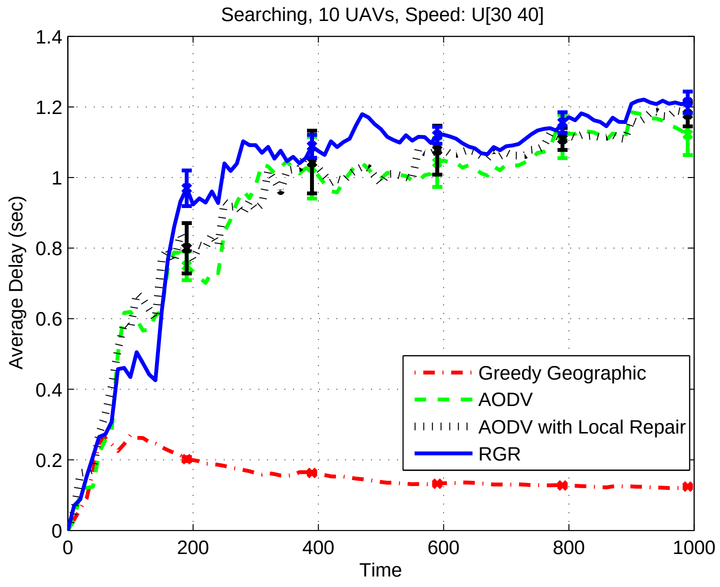

Figure B.4: Average delay in a searching scenario of size 25 km^2 including 10 medium speed UAVs

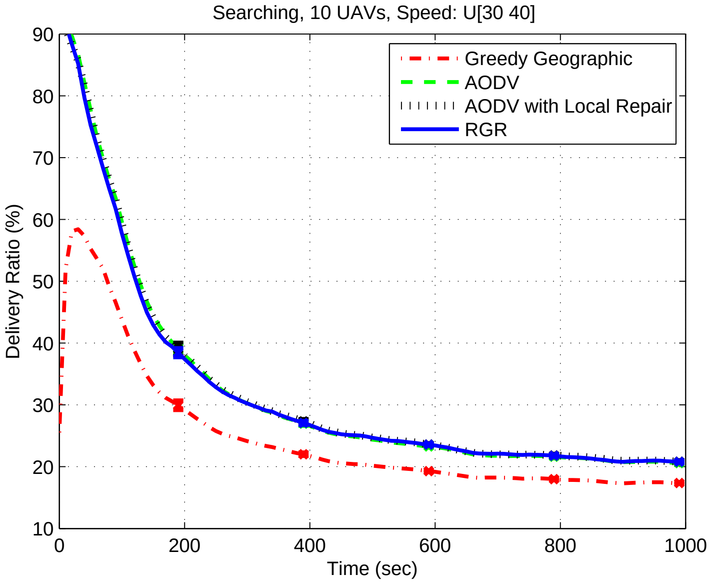

Figure B.5: Average delivery ratio in a searching scenario of size 25 km^2 including 10 medium speed UAVs

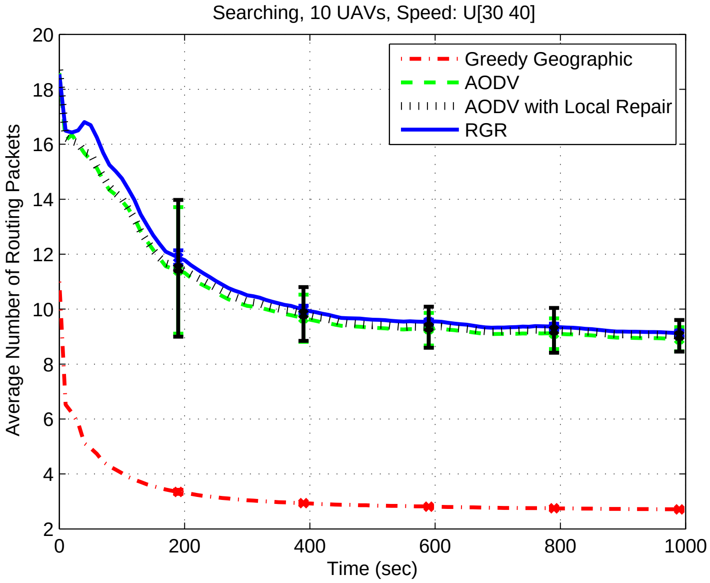

Figure B.6: Average routing traffic in a searching scenario of size 25 km^2 including 10 medium speed UAVs

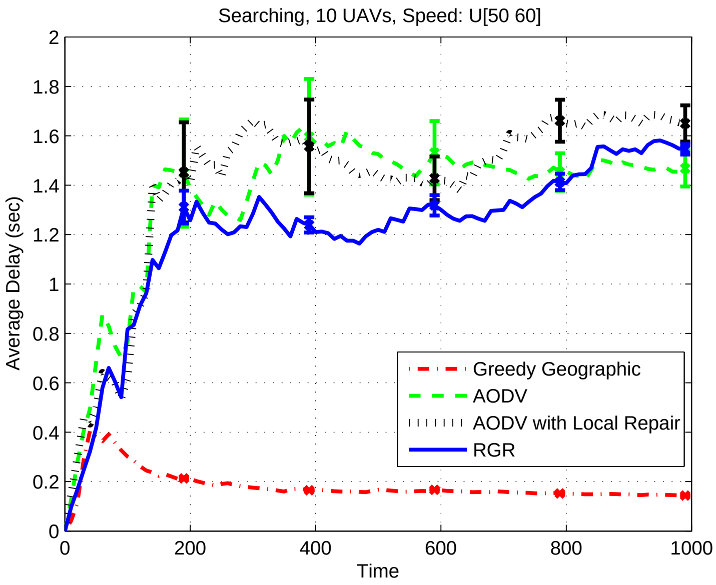

Figure B.7: Average delay in a searching scenario of size 25 km^2 including 10 high speed UAVs

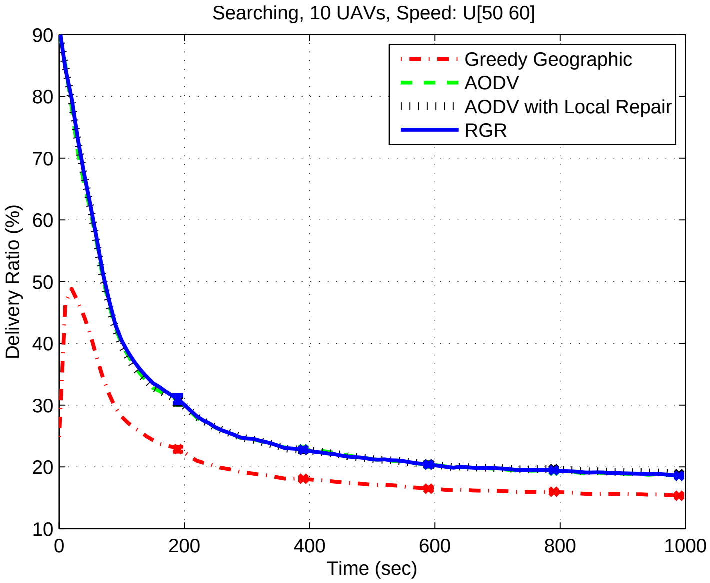

Figure B.8: Average delivery ratio in a searching scenario of size 25 km^2 including 10 high speed UAVs

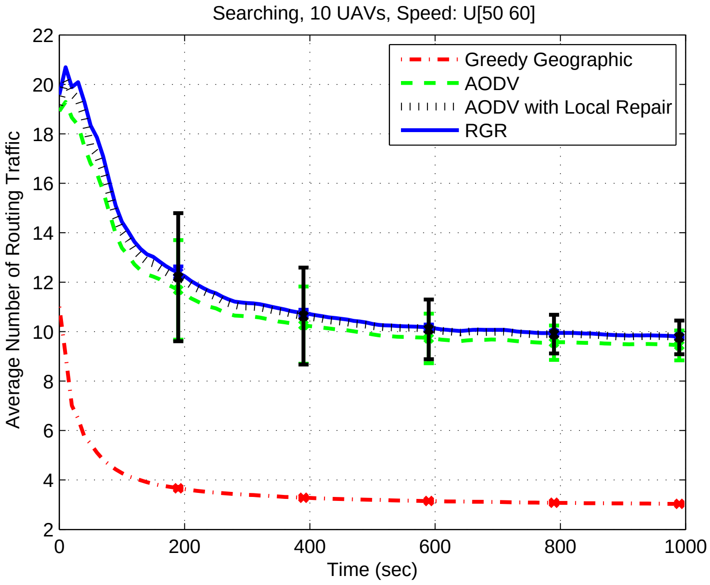

Figure B.9: Average routing traffic in a searching scenario of size 25 km^2 including 10 high speed UAVs

Appendix C

Searching Mission with 30 Nodes

In this Appendix, the figures for a searching mission with 30 UAVs in a 25 km^2 area are illustrated.

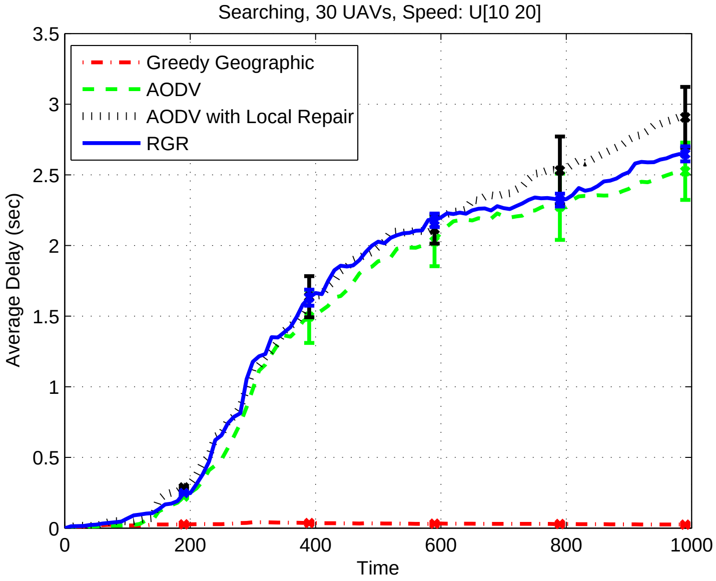

Figure C.1: Average delay in a searching scenario of size 25 km^2 including 30 low speed UAVs

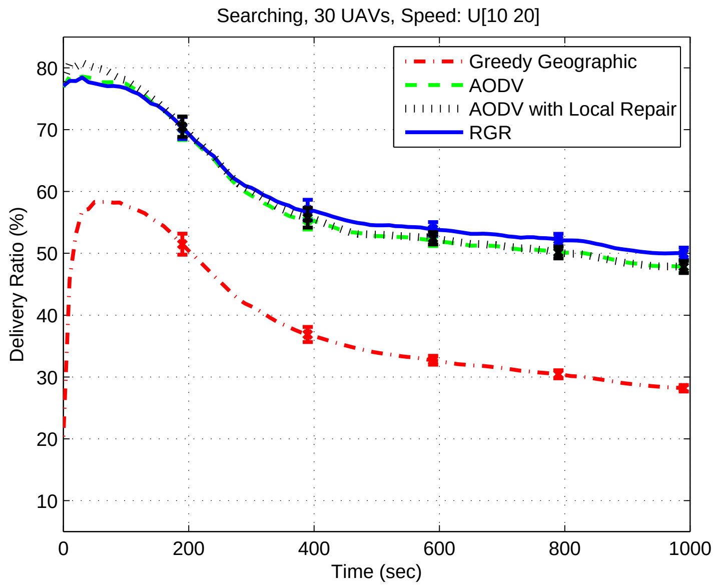

Figure C.2: Average delivery ratio in a searching scenario of size 25 km^2 including 30 low speed UAVs

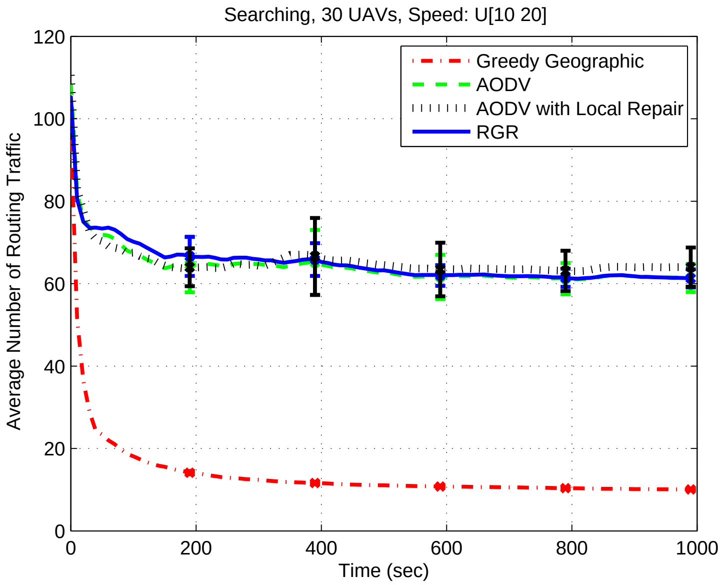

Figure C.3: Average routing traffic in a searching scenario of size 25 km^2 including 30 low speed UAVs

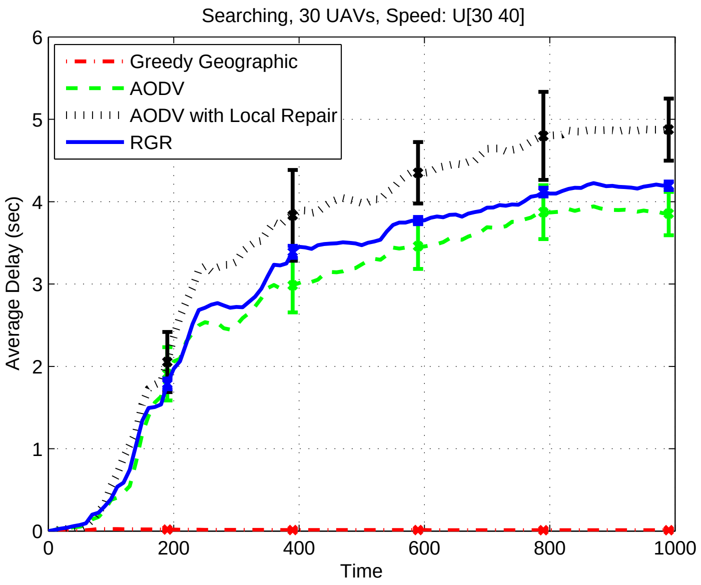

Figure C.4: Average delay in a searching scenario of size 25 km^2 including 30 medium speed UAVs

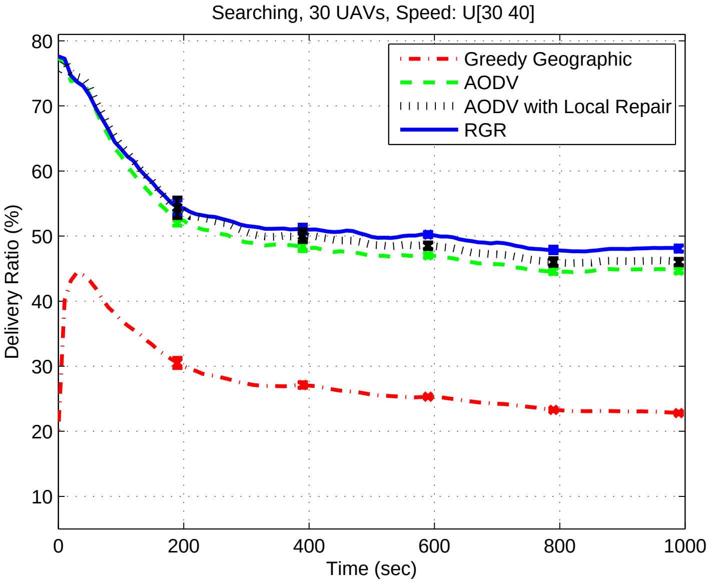

Figure C.5: Average delivery ratio in a searching scenario of size 25 km^2 including 30 medium speed UAVs

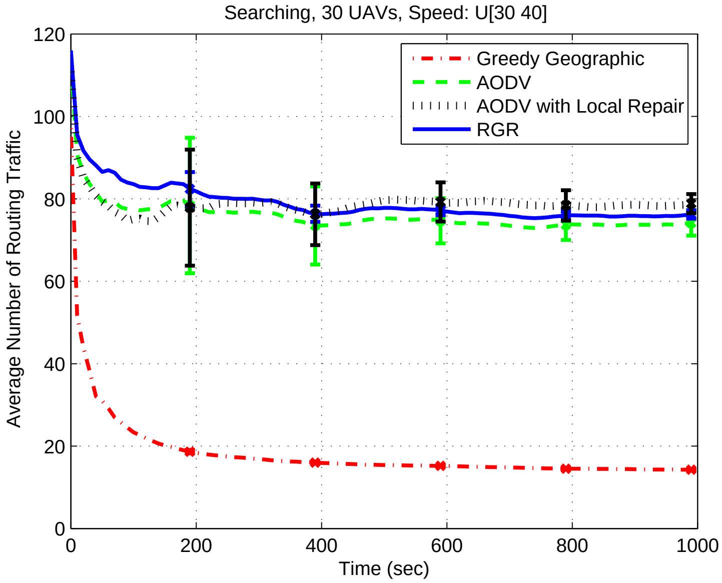

Figure C.6: Average routing traffic in a searching scenario of size 25 km^2 including 30 medium speed UAVs

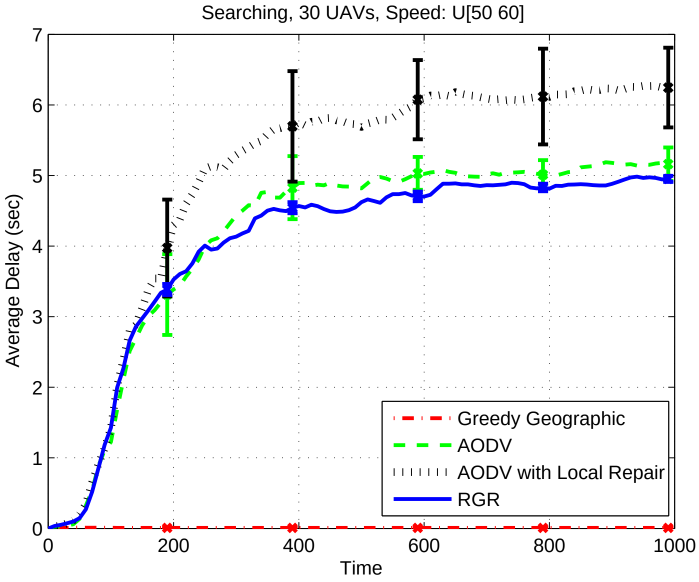

Figure C.7: Average delay in a searching scenario of size 25 km^2 including 30 high speed UAVs

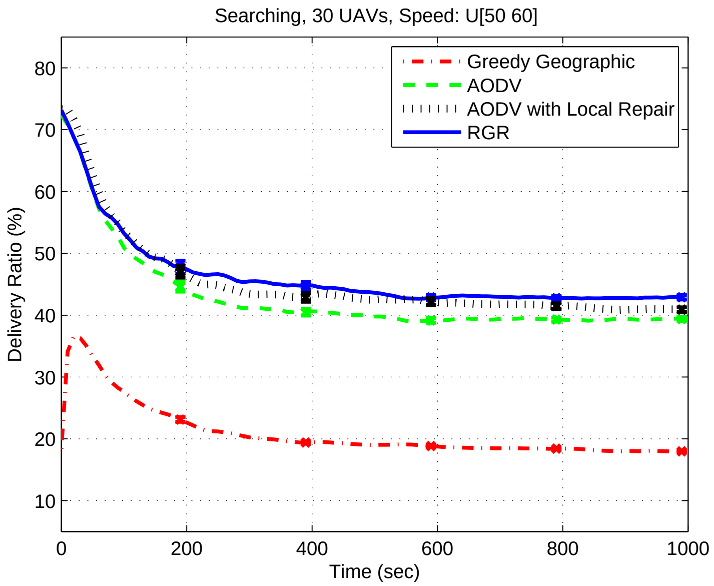

Figure C.8: Average delivery ratio in a searching scenario of size 25 km^2 including 30 high speed UAVs

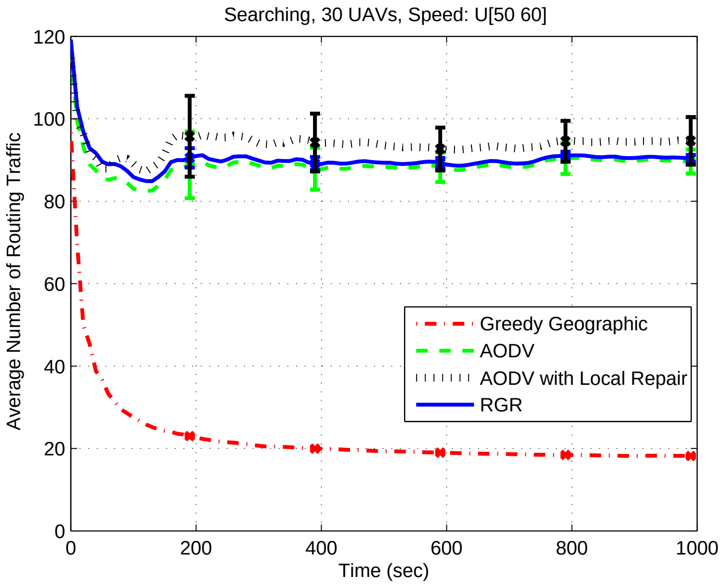

Figure C.9: Average routing traffic in a searching scenario of size 25 km^2 including 30 high speed UAVs

Appendix D

Tracking Mission with 20 Nodes

In this Appendix, the figures of a tracking mission including 20 UAVs are illustrated.

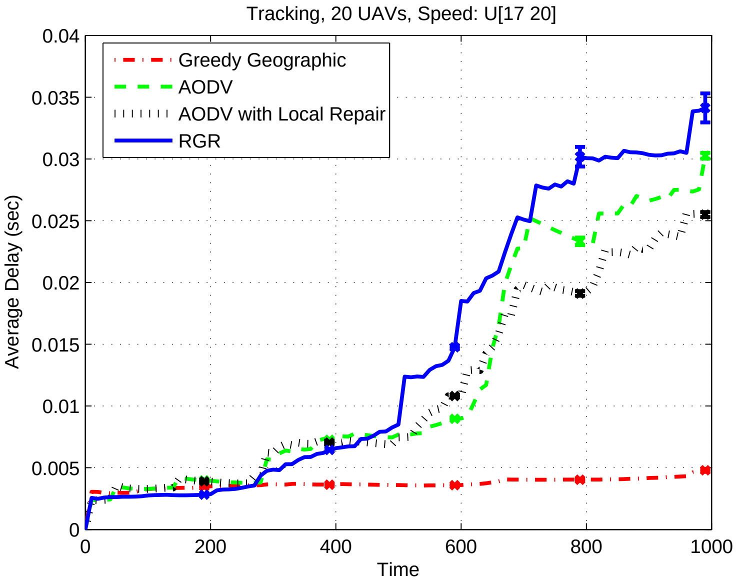

Figure D.1: Average delay in a tracking scenario including 20 low speed UAVs

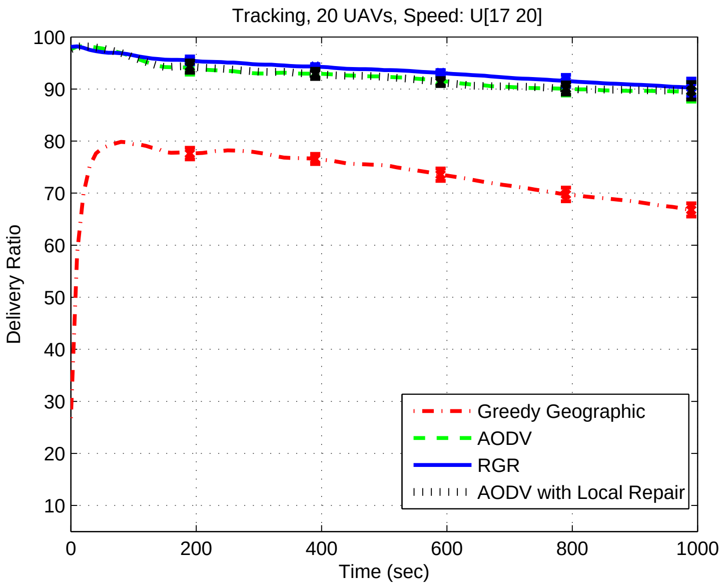

Figure D.2: Average packet delivery ratio in a tracking scenario including 20 low speed UAVs

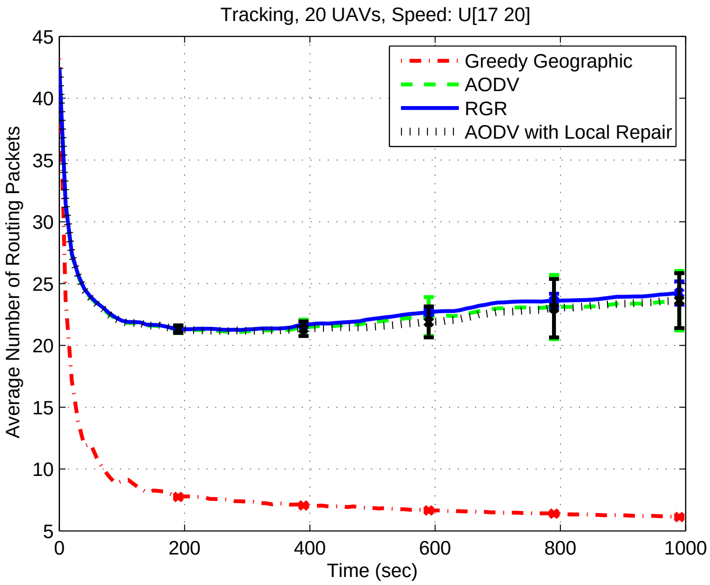

Figure D.3: Average routing traffic in a tracking scenario including 20 low speed UAVs

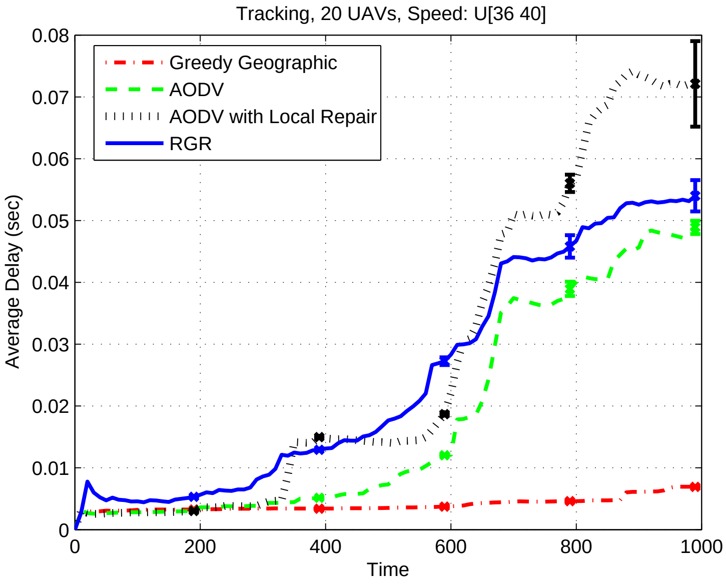

Figure D.4: Average delay in a tracking scenario including 20 medium speed UAVs

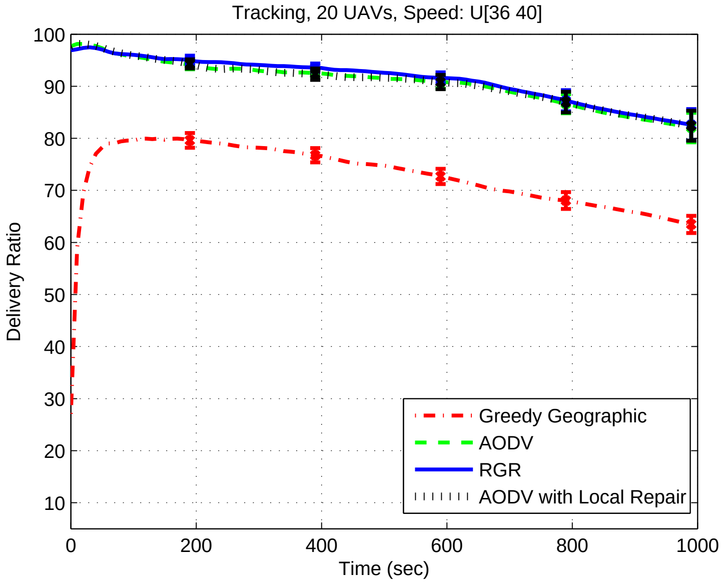

Figure D.5: Average packet delivery ratio in a tracking scenario including 20 medium speed UAVs

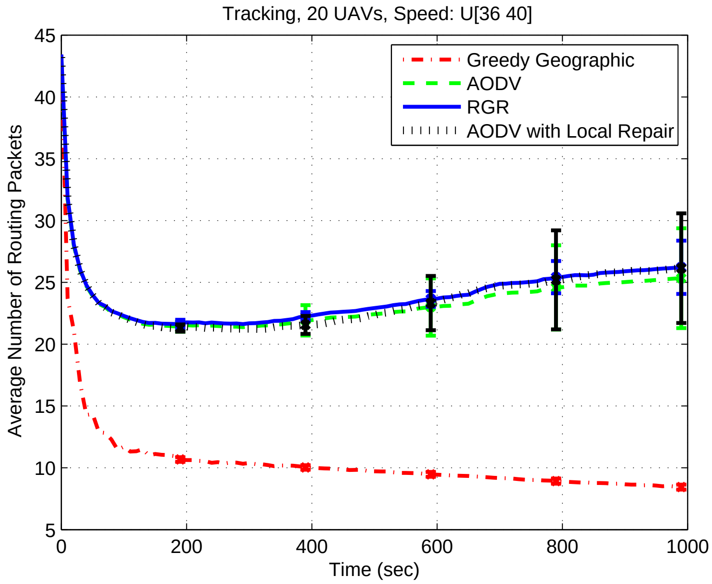

Figure D.6: Average routing traffic in a tracking scenario including 20 medium speed UAVs

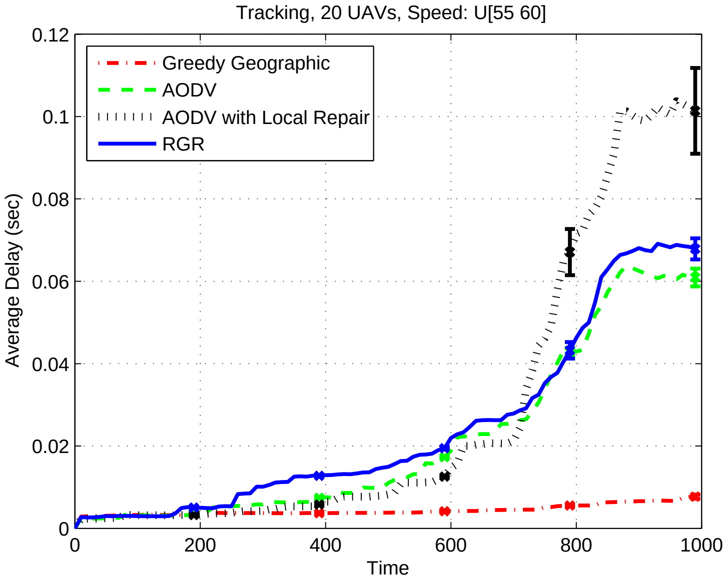

Figure D.7: Average delay in a tracking scenario including 20 high speed UAVs

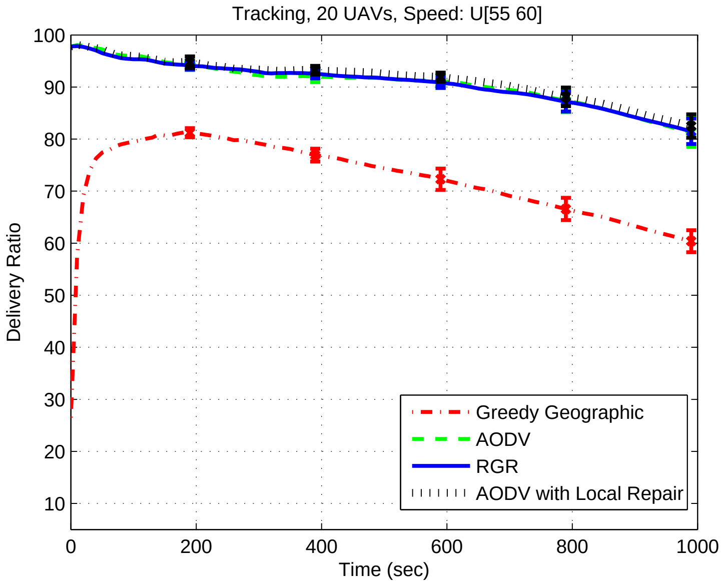

Figure D.8: Average packet delivery ratio in a tracking scenario including 20 high speed UAVs

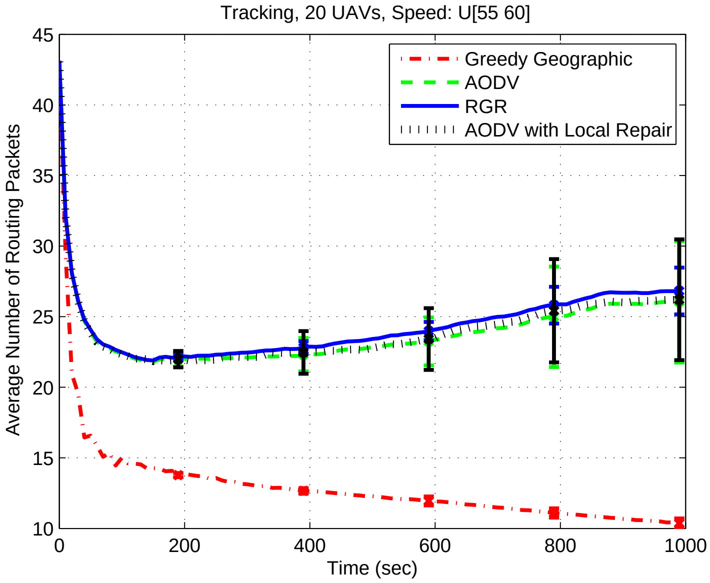

Figure D.9: Average routing traffic in a tracking scenario including 20 high speed UAVs