HYBRID ROUTING IN HIERARCHICAL AIRBORNE NETWORK WITH
MULTI/UNI/OMNI-DIRECTIONAL ANTENNAS

by

LEI HU

FEI HU, COMMITTEE CHAIR
SUNIL KUMAR, COMMITTEE CO-CHAIR
XIAOYAN HONG
AIJUN SONG
SHUHUI LI

A DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2018
ABSTRACT

During the previous decades, the advanced technologies such as unman aerial vehicle (UAV) and directional antennas in airborne networks has developed rapidly. However there is limited research on the networking protocols for airborne network considering the hybrid and complicated network topology and situations. During my PhD study, my research mainly targets the different routing protocols for hierarchical airborne networks. Also in order to test the performance of our proposed networking protocols, a directional wireless mesh network (WMN) testbed is designed and set up to study the protocols in routing layer and data link layer with the support of directional antennas.

In Chapter 1, I will briefly introduce the basic concepts and background which may be related to my research. Especially, the key point is the hybrid and hierarchical network topology and the novel technologies for advanced antennas for airborne networks.

In Chapter 2, I work on designing a hybrid two-layer routing protocol for a hierarchical airborne network. This hybrid routing protocol is proposed to achieve better performance such as throughput and end-to-to delay for the complicated network topology and situation in mobile hierarchical airborne networks.

In Chapter 3, we design and develop a WMN testbed equipped with directional antennas to test the performance for real-time multi-media communication. Different network topology with directional antennas are implemented and hardware demos are conducted in this testbed to study the performance and functions of the previous MAC and routing protocols in mesh network.

In Chapter 4, we target an innovative routing protocol with a new concept of low probability of detection (LPD) for covert communication. In our research, a novel routing protocol is proposed for the network equipped with directional antennas under the threaten of a potential adversary.
In Chapter 5, we make the conclusion about the research on routing protocols and WMN testbed. We summarized the research during the PhD study and the future work I will continue.
DEDICATION

This dissertation is dedicated to my family, my friends and these who helped and supported me.
ACKNOWLEDGMENTS

For this doctorate dissertation, I sincerely acknowledge my advisor, Professor Fei Hu, and the members of my committee, Professor Sunil Kumar, Professor Xiaoyan Hong, Professor Shuhui Li and Professor Aijun Song for their guidance and help during my PhD study at the University of Alabama.

Especially, I’d like to thank my advisor Dr. Fei Hu with my most gratitude. With his great support and guidance, I am able to work on my research and finish this dissertation. And during my PhD here, he is not only an advisor, but also a friend to me. I really appreciate his advice for my research and life during my PhD study.

I also want to thank my committee co-chair, Dr. Sunil Kumar, for his patient guidance and enthusiastic support on my research. Dr Kumar has given a lot of beneficial advice and insightful comments for my research. Also he has helped me a lot with my research ideas and academic papers.

And I also want to show my gratitude to Dr. Xiaoyan Hong for the help and guidance in the courses in computer science and other committee members Shuihui Li and Aijunsong. I’d like to thank you for your advice and comments on my dissertation proposal and final defense.

Furthermore, I’d like to thank my labmates for the time we’ve spent together on the projects, discussion and conference about our research. I am really grateful to the following labmates (names listed in no particular order) Dr. Yeqing Wu, Dr. Qingquan Sun, Dr. Rui Ma, Dr. Jiang Lu, Dr. Ting Zhang, Dr. Ji Qi, Dr. Xin Li, Dr. Qian Mao, Dr. Koushik A.M, Dr. Xiaoyang Liu, Dr. Shaowei Wang, Bailing Chang, Yu Lu, Zhijing Ye, Niloofar Toorchi, Immanuel David Rajan, Iftikhar Rasheed for all of your company and support during my PhD study.

Also a lot of thanks to my friends Lin Zhang, Zhiyong Xia, Yuan Cao, Man Lu, Dr. Wangxin Huang, Dr. Xiaoqing Wang for the good memory we share together in Tuscaloosa.
Finally, I’d like to express my most gratitude to all my family. I want to say thank you to my parents for the great efforts and sacrifice in these years. Thank all other family members for your support in my life.
CONTENTS

ABSTRACT ................................................................. ii
DEDICATION .............................................................. iv
ACKNOWLEDGMENTS ..................................................... v
LIST OF TABLES ........................................................ x
LIST OF FIGURES ......................................................... xi
CHAPTER 1 INTRODUCTION .............................................. 1
  1.1 Background and motivation ........................................ 1
  1.2 Research Area and Objective ....................................... 2
  1.3 Outline of Dissertation ............................................ 3
CHAPTER 2 HYBRID ROUTING IN HIERARCHICAL NETWORKS .......... 5
  2.1 Introduction .......................................................... 5
    2.1.1 Hierarchical hybrid wireless network ($H^2WN$) ............. 5
    2.1.2 Problem Statement of $H^2WN$ Routing Issues ................. 6
    2.1.3 Our Contributions .............................................. 9
  2.2 Related Work ....................................................... 10
  2.3 System Assumptions ............................................... 11
  2.4 High-Level Network: Fence-Structured Multi-Beam Routing ....... 12
    2.4.1 Problem Statement ............................................ 12
    2.4.2 Fence-Structured Routing Methodology ....................... 14
    2.4.3 Fence Routing Design ........................................ 16
  2.5 Moth-Inspired Low-Level Network Routing ........................ 28
    2.5.1 Problem Statement ............................................ 28
2.5.2 Moth’s Target Approaching Behavior .......................... 29
2.5.3 Moth-Inspired Event-to-Sink Data Delivery .................... 30
2.5.4 Handle Hybrid Uni-/Omni- Directional Antennas ................. 32
2.5.5 Moth-Inspired Routing Algorithm .............................. 34
2.6 Ant-Inspired Cross-Level Routing ................................. 35
2.6.1 Problem Statement .............................................. 35
2.6.2 Ant-Inspired Cross-Level Routing .............................. 36
2.7 Performance Evaluations ........................................... 39
2.7.1 Fence Routing with MBSA ....................................... 39
2.7.2 Moth Routing in the lower level ................................ 42
2.7.3 Ant-inspired Cross-Level Routing .............................. 46
2.7.4 Overall Network Performance .................................. 49
CHAPTER 3 DIRECTIONAL WMN TESTBED WITH QOS APPLICATION .... 56
3.1 Introduction ......................................................... 56
3.1.1 Directional mesh network (DMN) ............................... 56
3.1.2 Problem Statement of DMN Testbed ............................ 58
3.1.3 Our Contribution ............................................... 61
3.2 Related Works ..................................................... 62
3.3 System Architecture ............................................... 63
3.4 Hardware Preparation and Deployment ............................. 65
3.5 Directional Routing ................................................ 70
3.5.1 Routing Protocols .............................................. 70
3.5.2 Directional Antenna and Qos Application ....................... 78
3.6 Mac Layer Experiment ............................................. 79
3.7 Results and Evaluation ............................................ 81
CHAPTER 4 NOVEL OLSR ROUTING FOR LOW PROBABILITY OF DETECTION 91
4.1 Introduction ................................................................. 91
4.1.1 Background and Basics of Low Probability of Detection (LPD) .......... 91
4.1.2 Significance of Neighbor Discovery on Routing Schemes ................. 93
4.1.3 Basic of OLSR ......................................................... 95
4.1.4 Social Network Analysis Methods .................................... 96
4.1.5 Our Contributions .................................................... 98
4.2 Related Work ............................................................. 99
4.3 Narrow Beam Scan-based Neighbor Discovery Protocol ..................... 100
4.4 Link State Update in OLSR with Multi-Beam Directional Antennas ........ 102
4.5 Multi-path OLSR in LPD ................................................ 107
4.5.1 Limits of Reliable Communication with LPD .......................... 107
4.5.2 Multi-path OLSR with LPD .......................................... 112
4.5.3 Routing Shortcut Mechanism ....................................... 116
4.6 Performance Evaluations ................................................ 117
4.7 CONCLUSION ............................................................ 120
REFERENCES .................................................................... 122
LIST OF TABLES

2.1 Routing Table ............................................................... 20
4.1 Simplified Routing Table with LPD .................................. 112
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Illustration of nodes in a hierarchical hybrid wireless network</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>Features of multi-channel MBSAs.</td>
<td>12</td>
</tr>
<tr>
<td>2.3</td>
<td>Impact of high mobility.</td>
<td>13</td>
</tr>
<tr>
<td>2.4</td>
<td>Conventional routing schemes cannot exploit the MBSA's benefits.</td>
<td>14</td>
</tr>
<tr>
<td>2.5</td>
<td>Neural network features</td>
<td>15</td>
</tr>
<tr>
<td>2.6</td>
<td>Enhanced main path search via two-time launching.</td>
<td>18</td>
</tr>
<tr>
<td>2.7</td>
<td>Establishment of fence routing.</td>
<td>19</td>
</tr>
<tr>
<td>2.8</td>
<td>Prioritized fountain codes.</td>
<td>21</td>
</tr>
<tr>
<td>2.9</td>
<td>Multi-channel helps achieve full duplex.</td>
<td>23</td>
</tr>
<tr>
<td>2.10</td>
<td>Cross-layer MAC and routing schedule control</td>
<td>24</td>
</tr>
<tr>
<td>2.11</td>
<td>Gradient routing.</td>
<td>29</td>
</tr>
<tr>
<td>2.12</td>
<td>Moth's innate capability for an uncertain target.</td>
<td>30</td>
</tr>
<tr>
<td>2.13</td>
<td>LFC routing path establish scheme.</td>
<td>30</td>
</tr>
<tr>
<td>2.14</td>
<td>Hybrid antennas in Lower-level Network.</td>
<td>33</td>
</tr>
<tr>
<td>2.15</td>
<td>Prediction-based local repair</td>
<td>33</td>
</tr>
<tr>
<td>2.16</td>
<td>Cross-level communication in the network</td>
<td>36</td>
</tr>
<tr>
<td>2.17</td>
<td>The illustration of ant-inspired cross-level routing.</td>
<td>37</td>
</tr>
<tr>
<td>2.18</td>
<td>The illustration of ant-inspired cross-level routing.</td>
<td>38</td>
</tr>
<tr>
<td>2.19</td>
<td>Average throughput achieved by fence routing (width =3).</td>
<td>40</td>
</tr>
<tr>
<td>2.20</td>
<td>Average delay achieved by fence routing (width =3).</td>
<td>40</td>
</tr>
<tr>
<td>2.21</td>
<td>Average throughput achieved by fence routing (width = 2).</td>
<td>41</td>
</tr>
<tr>
<td>2.22</td>
<td>Average delay achieved by fence routing (width = 2).</td>
<td>41</td>
</tr>
</tbody>
</table>
2.23 Average throughput achieved by Two-level network (width = 3). .......................... 42
2.24 Average delay achieved by Two-level network (width = 3). ............................... 43
2.25 Average throughput of different routing protocols in the low-level network. ............ 44
2.26 Performance of moth routing. ................................................................. 45
2.27 Throughput for different updating periods in moth routing. ............................... 45
2.28 Moth routing performance for different number of hops in line section. ............... 46
2.29 Moth routing performance for different number of hops in circular area. ............... 47
2.30 Hop number in the routing path. ................................................................. 48
2.31 Packet loss rate for different cross-level routing protocols. ............................... 49
2.32 Delay for different cross-level routing protocols. ........................................... 49
2.33 Throughput for the overall network. ......................................................... 51
2.34 Delay for the overall network. ................................................................. 51
2.35 Throughput for new topology. ................................................................. 52
2.36 Delay for new topology. ............................................................... 53
2.37 Throughput for different protocols for new topology. ................................... 54

3.1 WMN Router .................................................. 66
3.2 Comparison of NAND partition .................................................. 67
3.3 Patch for NAND repartition ................................................ 68
3.4 Ramdisk Image .................................................. 68
3.5 Target Image .................................................. 69
3.6 Temporary System .................................................. 69
3.7 OLSR overview in Luci .................................................. 71
3.8 Interface configuration .................................................. 72
3.9 Forward section .................................................. 72
3.10 Neighbors in OLSR Mesh .................................................. 72
3.11 Topology in OLSR Mesh .................................................. 73
3.12 HNA in OLSR Mesh .................................................. 73

xii
CHAPTER 1
INTRODUCTION

1.1 Background and motivation

Rapid development in Antennas Technologies will lead to improve and change the way mobile and wireless communication systems are implemented. In recent years, a new antenna model named multi-beam smart antenna(MBSA) has more and more frequently applied because of their exception performance in high speed network. Nowadays, normally omni-directional antennas are applied in wireless mesh networks(WMNs). However, because of its global directivity, it may cause interference in the networks of high density. And also because of the exposed nodes, the performance in communication will be compromised for the network equipped with omni-directional antennas. Another popular antenna model is called directional antennas which tends to limit the signals into a narrow sector. Thus, the interference can be effectively avoided in communication with directional antennas. Another benefit for this directional antennas is that the transmission range can be significantly increased based on the centralized radiation energy. In order to further improve the benefits of directional antennas, MBSA is proposed to implement multi-path communication for better performance, which means multiple communication can be achieved at the same time. Simultaneous communication in different beams of MBSA could be a very effective way to implement space division multiple access(SDMA).

The range of MBSA applications has been growing in the last few years. MBSA could be deployed for potential applications such as scientific survey, long distance search and radar system. One important area for application with MBSA is in the military area. There are already a wide range of MBSAs currently being used or developed by the military in a hierarchical airborne
network with complex network topology. Such a network has the following 2 main features: (1) Two-level architecture: The high level is a sparse wireless network with long-distance links and high-rate communications. It also has a commander node that collects data from the entire network and sends control commands. The lower level is a dense network with short-distance low-rate links. (2) Hybrid directional antennas: The network has mixed antennas (multi/uni/omni-directional antennas). Normally the high-level nodes are equipped with more expensive, multi-directional (also called multi-beam) antennas, while low-level nodes have inexpensive omni-directional or uni-directional (single-beam) antennas.

In my PhD research, I mainly focus on the routing protocols in a hierarchical airborne networks which may be comprised of multiple heterogeneous levels. And each level has its own properties, topology and structures such as antennas, node density and link rate. For example, in our hierarchical airborne network, on the lower level, there is a network of high density filled with unmanned aerial vehicles (UAVs). And the UAVs are equipped with omni-directional antennas. However, on the high level, there are large aircraft which are equipped with MBSAs which can achieve long range and multi-path communication.

1.2 Research Area and Objective

In our research, our goal is to build an innovative routing that can (1) fully explore the multi-beam multi-channel links to achieve a high-throughput, GPS-free transmission between long-distance high-level nodes, and (2) in the low-level network, allow any event node to report data to a sink node that could suddenly move away. Our proposed routing also allows the event node to trace the movement trajectory of the high-level commander node in order to efficiently deliver data to it. We propose to use bio-inspired algorithms, called MAN (Moth, Ant, Neuron), to achieve the above routing. In addition, we use human brain neuron grid architecture to construct a weighted fence routing among multi-beam high-level nodes in order to achieve a high-throughput routing. Also we optimize the proactive link state routing protocol (OLSR) in ad hoc networks with multi-beam directional antennas (MBDA), by considering the fundamental limits of reliable LPD communication which is an import aspect in military communication. In
order to study and test the networking protocols, we also design and implement a directional wireless mesh network testbed. Our testbed is based on OpenWrt, a Linux distribution for embedded system. Also this paper describes the detailed implementation of the WMN test bed with directional antenna. We conduct experiments on this testbed for multimedia video transmission and apply Qos mechanisms in the wireless mesh network.

1.3 Outline of Dissertation

The outline of this dissertation is structured as follows.

Chapter 2 shows the design and implement of an innovative hybrid multi-path routing protocols for a hierarchical airborne network equipped with MBSAs and omni-directional antennas. On the higher layer, a fence-structured multi-path routing protocol is proposed and designed to exploit the benefits of MBSAs. And on the lower layer, for the problem of singular mobility, a hybrid routing protocol is designed which is composed of three sections including line section, AODV section and gradient routing section. Also the simulation results validate the better performance of this new hybrid routing protocol for this airborne network.

In Chapter 3, we introduce a directional WMN testbed for the study of networking protocols in networks equipped with directional antennas. We choose an open-source embedded system named openWRT as our operating system in the mesh routers. And we work on designing and modifying the MAC protocols and routing protocols with the source code for the Linux drivers. And in order to test the performance of our proposed protocols and policies, we design and implement several experiments for different network topology with directional antennas.

In Chapter 4, a new concept named low probability of detection(LPD) is introduced and studied. Besides the previously proposed routing protocols in an airborne networks, a new model for potential adversary is introduced in the network, which may detect and jeopardize the communication of the nodes. Thus, in order to deal with the adversary, covert communication should be assured in this unsafe network with potential detectors from the enemies. Therefore some limitations for LPD is applied in this network and a new routing protocol is proposed for covert communication with directional antennas.

3
Chapter 5 makes a conclusion for the dissertation and our future work.
CHAPTER 2
HYBRID ROUTING IN HIERARCHICAL NETWORKS

2.1 Introduction

2.1.1 Hierarchical hybrid wireless network ($H^2WN$)

We target the routing issues in a hierarchical hybrid wireless network ($H^2WN$). It has a typical two-level topology and hybrid antenna systems (omni-/multi-directional antennas). In the high-level network, it has a relatively sparse topology consisting of powerful nodes that can communicate with each other with long-distance links. These nodes are equipped with multi-beam antennas to achieve high-rate transmissions. In the low-level network, it has much higher node density and lower mobility than the high-level network. The nodes in low-level network are equipped with omni-directional antennas.

The hierarchical network can be deployed in many applications such as airborne network and sensor and actuator network [18]. Such a hierarchical architecture provides the separation of different types of wireless nodes, easy network management, and good balance between deployment cost and network throughput. For instance, we can use the expensive high-level nodes to deliver the high-throughput data that is generated from many low-level cheap nodes. Typically there is a ’sink’ node in the low-level network that can collect the event data from other nodes. The high-level network often has a node that serves as the role of ‘commander’ that can control the whole network.

Today’s antenna systems have been improved drastically these years due to the rapid development of advanced circuits and materials. The inexpensive popular antenna is omni-directional antenna. It simply radiates the same signal to all directions even though it may target only a specific node. To better focus the energy on a specific direction, we can use
directional antenna with one beam (also called unidirectional antenna). Since all signal energy is concentrated in a single angle, it avoids the interference with other directions, and also is able to send data for a longer distance than omnidirectional antenna. Nowadays people start to look into another cost-efficient antenna, called multi-beam smart antenna (MBSA). The MBSA simply extends the single-beam antenna to multi-beam structure, and allows independent transceivers to be used in each beam. Each beam can send out different data according to its own queue management policy. Thus a MBSA significantly improves the throughput through simultaneous data transmissions in multiple directions. Since it does not rely on the receiver’s feedback, some directions may have poor performance. But those beams can ‘help’ each other by using detour beam transmissions. In this paper we assume that the high-level nodes are equipped with MBSAs. Thus those powerful nodes can achieve long-distance, high-rate transmissions. But the low-level network still uses cheap omnidirectional antennas.

2.1.2 Problem Statement of $H^2WN$ Routing Issues

This research targets the routing protocol design in the above described $H^2WN$. Particularly we will target 3 routing tasks: low-level routing, high-level routing, and cross-level routing (see Fig.2.1). Note that those 3 routing tasks can form a complete end-to-end communication task across the entire $H^2WN$: An unmanned aerial vehicle (UAV) can use ‘low-level routing’ to report its event data to a mobile sink, which uses ‘cross-level routing’ to reach a high-level aircraft, which can then use ‘high-level routing’ to reach another aircraft.

Problem of low-level routing: We will handle the UAV-to-UAV routing issue in the low-level network. Particularly, we assume that there is a UAV that serves as the role of sink. If any UAV detects an abnormal event, it immediately forwards the data to the sink via multi-hop routing scheme. Although general event-to-sink routing has been studied [20, 31], they can not handle the sink’s singular mobility problem. Unlike the high-level nodes that have ‘even mobility’, that is, each node has similar mobility (for aircraft network, it is generally 80 – 120m/s), the low-level network has ‘singular mobility’, that is, most nodes may have little or low mobility, while the sink could have high mobility due to its task of global data collection.
Note that the above ‘singular mobility’ issue is popular in many mobile data collection applications. The sink needs to move around to collect the data from the nearby nodes. If the sink is static, the nodes that are far from the sink will take a long time to reach the sink. And it is also unfair to the nodes near the sink since they need to help to relay the packets from all other nodes. By using singular mobility of the sink, we can make all nodes have similar probability of relaying the data. Besides UAV network, in many other networks (such as wireless sensor networks), the sink (an actuator or a robot) is often mobile to detect events in the whole network.

To handle the sink’s singular mobility, even an on-demand (reactive) routing such as AODV protocol could not fast track the sink’s sudden leaving, not mentioning the proactive routing (such as DSR scheme) that only works well for semi-static network topology. As a matter of fact, since most of other low-level nodes do not have such sudden leaving behavior, the global path search via blind RREQ broadcasting would cause much routing overhead in a high-density, low-rate ad hoc network (such as our targeted UAV network here). Our work will solve such an event-to-sink routing search under singular mobility. Our routing scheme will be able to fast track the sink’s movement area. It first uses a multi-hop relay path to quickly reach the approximate ‘sink area’, and then uses gradient routing to deliver the data to the exact location of the sink.

Figure 2.1: Illustration of nodes in a hierarchical hybrid wireless network.
**Problem of high-level routing:** We will also address the aircraft-to-aircraft routing issue in the high-level network. The biggest challenge is how to maximize the benefits of multi-beam antennas in the high-level routing process.

**Problem of cross-level routing:** Between the two levels, a critical routing issue is how an event node in the low level can quickly localize the *commander* node in the high level, and then efficiently forwards the data to a low-level node that is closest to the commander node. Note that the high-level network is much sparser than the low-level network. Thus it is difficult for an event node to find the closest high-level node. In order to deal with these issues, we develop a GPS-free commander node tracking scheme in order to efficiently find a low-level node that is closest to the commander node.

Currently there are some existing schemes to deal with the issues of cross-level routing. For example the wireless sensor and actuator networks [1, 27] has the cross-level scheme to deal with the interconnection between the actuators in the high level and the sensors which collects data in the low level, however this routing scheme simply assumes the actuators in the high level are static and the position for the commander is known in advance. Thus the sensors can upload the collected data to the fixed commander just using conventional routing protocols such as link state routing. Therefore this protocol cannot deal with the high mobility for the nodes in airborne networks. Another typical wireless network with cross-level routing schemes is wireless sensor and robot network [10, 37, 52]. This network developed cross-level algorithms for data delivery between the sensors and robot. Although this scheme can deal with the scenarios with mobility for the nodes in the high level, however this scheme doesn’t consider the multi-path routing with multi-beam antennas in the high level as well as the singular mobility in the low level. In this robot networks, it just applies a conventional routing protocol for the low level. Therefore if it is introduced in our airborne network, without cooperation with the Moth-Inspired routing in the low level, the efficiency of this protocol may be significantly compromised. Also multi-path routing is also not introduced in wireless robot network, so this is another factor that may lead to the poor performance of this cross-level scheme in our hierarchical networks. In order to further
study the performance of different cross-level protocols, we also do simulation for our proposed cross-level routing and the existing schemes in section VII of this chapter.

2.1.3 Our Contributions

In this research, we will propose a series of innovative routing solutions to overcome the above challenging routing issues based on bio-inspired principles

(1) Fence-like high-level multi-beam routing: We propose to build a fence-like routing scheme in the high-level network with a routing topology similar to the weighted neural network, to concurrently dispatch the data to multiple beams. Such a fence routing structure can explore the high-capacity of MBSAs very well.

(2) Moth-inspired low-level network routing: Male moth has a peculiar trajectory when searching for a light source. It follows a straight line first, then follows zigzag curve, and finally uses circular trajectory to lock the source. Such a pattern helps to quickly localize the uncertain light source. Inspired by moth movement, we propose a line-fan-ring (LFR) routing search scheme to handle the sink’s singular mobility issue in the low-level network.

(3) Ant-inspired cross-level routing: Ants use striking chemical scents to record the trajectory such that their partners can quickly find the food source. Such a trail has time-decaying feature. Inspired by ant chemical trail, we use high-density low-level nodes to record the trajectory of the commander node that is in the high-level network, and create a time-decaying routing path to reach the commander node from any place in the low-level network.

Roadmap: The rest of this paper is organized as follows: In section II, we will briefly summarize other related work. Then section III gives our system assumptions. Section IV has the detailed discussion of high-level network routing scheme with neuron grid architecture. We then move to the low-level routing based on moth-inspired algorithm in section V. The cross-level routing is given in section VI. The simulation results are provided in section VII, followed by the conclusions in Section VIII.
2.2 Related Work

Most conventional ad hoc routing schemes assume that the networks are equipped with omni-directional antennas [35, 36]. In order to improve the throughput performance, directional routing has been proposed in a few works [47, 53]. However, most of them just consider the directional antenna with one single beam [4, 22]. Some reactive routing schemes such as AODV, search and maintain the routing paths only if these routes are required for a specific session. This helps to efficiently reduce the route maintenance overhead. However, for the networks with MBSAs, these conventional routing protocols have some dominant drawbacks such as traffic flooding when searching for the new paths in the networks with high-density of nodes. Those conventional routing schemes often assume the use of omnidirectional antennas, and do not have concurrent, well-scheduled data delivery features when MBSAs are used in the high-level network.

In [34], the author proposed a multi-path data delivery scheme based on OFDMA and MIMO. However, MIMO has entirely different features from MBSAs (see Fig.1). We previously proposed a diamond-based routing protocol for the airborne network [5]. However, it cannot explore all the beams due to the use of a single relay node in some hops. In this work, we propose a fence-like routing protocol in which each hop has multiple nodes that can explore the multiple beams for concurrent transmissions.

In some of recent works [32] [29] about hierarchical network routing, they assume a cluster-based network topology. But they do not consider the singular mobility of the destination, which may become a critical factor affecting the overall performance of the hierarchical network. In [40], a moth-inspired routing protocol is proposed to improve the communication performance in mobile networks. Under certain conditions this protocol may improve the throughput and reduce the packet loss rate in the network. However, it can easily select the non-optimal routing path, and the performance such as network delay and communication overhead may be sacrificed. In [50], a bio-inspired node localization mechanism for hierarchical networks is proposed. In that work, the current location of the mobile sink is projected into the 2D hull to maintain its location.
information. This mechanism can help to build a cross-layer routing path in the hierarchical networks. The geographic routing is used there and the performance of this protocol can be further improved by applying other mobile protocols such as DSR or OLSR.

However, no work has been conducted on the three challenging H$^2$WN routing issues including the sink’s singular mobility, multi-beam concurrent transmissions, and cross-level routing design.

2.3 System Assumptions

Here we target a H$^2$WN with hybrid antennas. Since omnidirectional antennas have been well studied before, here we explain the MBSA features:

*Tx/Rx consistency*: All beams can send (Tx) or receive (Rx) data. And each beam could operate in different channels. However, for those beams using the same channel, they should obey “beam consistency” principle, that is, those beams should be either ALL-Tx or ALL-Rx mode (see Fig.2.2 Left). The reason of this constraint is due to the energy leaking from one beam’s main lobe into another beam’s side lobe if they are in different Tx/Rx modes. Of course, if they use different channels, there will be no energy leaking since there is no radio interference between different channels (see Fig.2.2 Right).

*Beam locking*: If a beam finishes its data transmission earlier than other beams, it can NOT change its Tx/Rx mode if there exist other beams that are using the same channel to send/receive data. For this case, this beam can switch the channel in order to change its Rx/Tx mode.

Due to the above principles, it is critical to schedule the data transmission in the hop-to-hop routing path carefully among the high-level nodes, in order to fully explore the high-capacity of the MBSAs, meanwhile not violating the above communication constraints in different directions.

*On the impact of high mobility in the high-level network*: In airborne networks, the high-level nodes are aircrafts, which could have high mobility. However, such a high mobility does not mean that we cannot build a relatively stable routing topology. As shown in Fig.2.3, assume a beam has 60 degree of coverage. The aircraft-to-aircraft distance could be 100km long.
An aircraft with initial position in the center of the scope and the mobility of $120m/s$ will take average 7 minutes to fly out of a beam’s scope ($50km$ long). For such a long time, any routing protocol could easily finish a typical communication session. Even though the light propagation delay is $0.3ms$ for $100km$ of link distance, a common DSR-based routing protocol just needs around $5ms$ of time to finish the propagation of RREP and receive the RREQ. Therefore, for airborne network, high mobility is not a main concern in terms of routing protocol stability.

This work will use airborne network as a $H^2WN$ example. The proposed routing schemes can be easily extended to other $H^2WN$ applications. In the low-level airborne network, the UAVs have limited antenna gain and short radio propagation distance. We assume that an UAV can only reach the aircraft right above it. If the commander node moves to another place, a UAV needs to search the routing path to reach a UAV that is closest to the commander. This is the motivation of recording the trajectory of the commander by using the high-density UAV network.

### 2.4 High-Level Network: Fence-Structured Multi-Beam Routing

#### 2.4.1 Problem Statement

The high-level network consists of powerful nodes with MBSAs. As an example, the aircrafts can use high-gain MBSAs to send data to all directions within a $>50km$ of radius. The high-level routing should fully explore the benefits of multi-beam, concurrent Tx/Rx capabilities. However, none of existing ad hoc routing schemes explores the MBSAs’ benefits very well.

As shown in Fig.2.4 (a), general DSR or AODV-like routing protocols only use a
Figure 2.3: Impact of high mobility.

single-path to deliver data. They use at most 2 beams of the MBSA. One may argue that other source/destination pairs may help to utilize the multi-beam capability of a node. However, those pairs' paths may have very few intersection nodes (note that only those intersection nodes can explore their multi-beam capability). Handling multiple pairs in one single routing scheme (Fig.2.4 (b)) would involve complex schedule coordination among those pairs, if we want to ensure that those intersection nodes follow multi-beam concurrent communication constraints. Furthermore, those pairs may have very different starting/ending times and QoS(Quality of Service) requirements. Thus it is difficult to coordinate the hop-to-hop relay schedules among all nodes, especially in those intersection nodes. In this work, we target the throughput maximization problem for a particular communication pair by using as many beams as possible to deliver data. Thus multiple pairs will automatically benefit from each individual, throughput-maximized communication pair.

One may think of multi-path routing could possibly explore the MBSA capability. It is true that by using multiple, intersected paths we can at least explore the multi-beam capability of some nodes (such as nodes $R_1$ and $R_2$ in Fig.2.4 (c)). However, those paths may be loosely coupled, and most relay nodes may still use at most 2 beams during their communications. A more coupled routing structure called Diamond Routing [5] is shown in Fig.2.4 (d). Here we use
periodical traffic convergence/divergence nodes (such as $R_1$ and $R_2$) to guarantee that there always be a main path consisting of fusion nodes that can explore their multi-beam capabilities. Now the question is: how do we explore the multiple beams of the rest of nodes in such a fusion routing topology, such as A, B, C, D, etc., which may still use only 2 beams for Tx/Rx?

### 2.4.2 Fence-Structured Routing Methodology

Neuro scientists have attempted to understand how the human brain uses the largest intelligent network in this planet - neural network with billions of neurons, to memorize things. With only a few micro-watts of energy, so many neurons transfer biological pulses to each other to perform complex logical deduction activities. Recently, Google AlphGo machine uses the Deep Leaning mechanism (a neural network with over 160 layers of neurons) to beat the top players of the chess called Go, a popular game in major Asian countries. When we look at the findings of the neuroscience [38, 48], it is surprising to see that there exist essential commonalities between neural networks and MBSA-based routing(Fig.2.5):

1. **MBSA-like tentacles help a neuron to quickly deliver pulses to neighboring neurons:**

   Using special cell microscopy, people have found that a neuron could have numerous tentacles in its ending location. Those tentacles look like a MBSA’s beams. A neuron could fast pump different amplitudes of bio-electrical signals into hundreds of tentacles simultaneously. Those
tentacles send or accept bio-pulses concurrently. This matches with MBSA’s Tx/Rx consistency and beam synchronization principles.

(2) The neurons use tentacles with different biological amplitudes (‘weights’) to form the large neural network: A neuron’s tentacles do not use the same pulse control level. As a matter of fact, each tentacle has very different bio-pulse amplitude compared to others. Using those ‘weighted links’, billions of neurons form a perfectly coupled neural network across left and right brains.

(3) Human brain quickly recalls something by building a main neuron path: Scientists have found that the brain is able to broadcast a bio-electrical query message to certain area and retrieves the results by enhancing a ‘main path’ that consists of most relevant neurons. Such a main path is important since it helps to retrieve more and more detailed history information from nearby neurons. It does this by extending the main path to a wider neuron path. This explains why someone remembers something’s big picture first, and then retrieves more and more details.
(4) Artificial Intelligence via a fence-like, weighted neural network for deep learning:

Based on the weighted neural network architecture, people have successfully achieved intelligent pattern learning through pre-training and self-learning of those weights before testing a new data set. As shown in Fig.8 (4), the entire neural network could have multiple ‘hidden layers’ with different weights in each neuron link. Such a fence-like neural network can achieve artificial intelligence.

2.4.3 Fence Routing Design

Search for Main Path

Inspired by the above neural network concept, we will build a fence routing for multi-beam transmissions among high-level network nodes. To determine the main direction and range of such a ‘fence’ area, we will need to search for a main path, just like the above main neuron-chain idea. Then later on we can add 1 or 2-hop nodes around the main path to form a grid-like fence routing.

Although we can simply use DSR-like RREQ broadcasting to establish a main path, such a blind global broadcasting may cause high routing protocol overhead, especially for large-scale networks. If more than 4 beams are available, a node can further distinguish more directions. We do not assume the existence of GPS-based location information since in many poor weathers or complex terrains the GPS satellite signals may not be available or poor. Without knowing the locations, the geographical routing algorithms may not perform correctly.

To simplify the routing protocol, we only use the multiple beams of a node to send out RREQ messages. Due to the cheap, easy-to-install compass, each node can easily determine 4 directions (east, south, west, and north). If more than 4 beams are available, a node can further distinguish more directions. A simple routing scheme, called ORRP (orthogonal rendezvous routing protocol), only uses two orthogonal directions (i.e., west-east and south-north) to search a routing path. We could adopt it to limit the RREQ transmissions only in 4 beams/directions.

A simple routing scheme, called ORRP (orthogonal rendezvous routing protocol) [13], only uses two pairs of orthogonal directions (i.e., west-east and south-north) to search for a
routing path. ORRP consists of two parts: (1) **proactive** maintenance of rendezvous node: a node maintains multiple rendezvous nodes which may be a few hops of away from the node. A node can easily appoint some nodes as rendezvous nodes by issuing R-RREQ messages to 4 directions, and then periodically check the existence of those nodes. If they move too far away, they will be replaced with closer ones. (2) **reactive** establishment of an orthogonal path to one of the rendezvous nodes: If a source wants to search a path to its destination, it sends out RREQ in 4 directions. If in one of its directions it reaches a rendezvous node, it will use that path as the routing path.

ORRP has a few important aspects: (1) in the reactive rendezvous node search process, when any relay node receivers such a RREQ message, it will use the beam with the reverse direction to forward the message. (2) although the sender prefers to search a node in a straight line of the beam direction, it may encounter a ‘void area’ where there are no forwarding nodes. For this case, an ‘angle correction’ strategy is proposed to go around the void area by adjusting the beam angles.

Since we only limit RREQ in 4 directions, it is possible that the sender may never find a rendezvous node. However, in [11] it has proved that the failure probability is less than 4%. And such case only occurs when the network is too sparse. As a matter of fact, any two pairs of orthogonal lines have over 95% of chance to intersect with each other. If the network is too sparse and a rendezvous node cannot be found, conventional DSR-based blind RREQ broadcasting can be used to search for the main path. Although the path found by ORRP may not be the shortest one, it is again proved that the hop number difference between ORRP and DSR is minor. In a nutshell, ORRP does not use blind global network search, and does not assume the GPS locations are known either. It uses the 4 beams of a MBSA to search for a main path. It is scalable to large-scale network due to the limited RREQ transmissions.

**ORRP Enhancement via Two-time Launching:** The original ORRP could find a path that is much longer than DSR result. In this work we further improve ORRP by using two-time launching concept (Fig.2.6). In the first-time launching, we use the above described method to
find two paths (i.e. two pairs of orthogonal lines could have at least two intersection nodes). Thus we know that a shorter main path must be in such a near-rectangular area. Then the sender rotates about 45 degrees toward the 'inside' direction (Fig.2.6), and performs the second-time launching, i.e., sending out RREQ message in the new direction until reaching a rendezvous node that is pre-maintained by the destination node.

Note that in a real MBSA it has at least 4 beams (to cover 4 directions). If the MBSA has 1 more beam between any two directions (thus it has total 8 beams), we do not need to rotate the sender’s MBSA since the beam between the two directions could be used to launch the RREQ message.

By using the above two-time launching, we could find a main path in the diagonal direction of the rectangular area. Such a main path is very close to the DSR result.

![Figure 2.6: Enhanced main path search via two-time launching.](image)

The search process for the main path based on the enhanced ORRP with 4-beam antennas is shown in Algorithm 1.

**Establishment of Fence Routing**

Based on the neural network principle, one can recover all the details of a past event by activating the nearby neurons around the main path. The links between the neurons have different weights, to reflect the fact that some life details could be remembered better than others. Those weighted neurons form a fence-like 'information pipe' to recover all the relevant memory signals.
We thus propose to build a fence routing based on the neural network principle. As shown in Fig.2.7, once a main path is found (here it is S-C-F-H-D), each node in the main path (except the source and destination) searches the 1-hop away neighbors. Note that here we ask the node to search the 1-hop neighbors in the direction that is orthogonal to its ORRP RREQ forwarding direction. For example, F uses its neighbor E as the fence node, C uses B and D, and H uses G. This is important in order to keep similar distances between neighboring fence nodes (such as B - E and E - G). Since the RSS (received signal strength) is proportional to the link distance in free space, the main path node always selects the node that gives itself a good RSS. It is possible that a main path node does not have any 1-hop neighbor in the orthogonal direction. Or, it may find only one neighbor in one side.

One may argue that a node could select 2-hop-away neighbors to serve as fence routing in order to establish more links among them. However, it has some drawbacks: (1) in some high-level network such as airborne network, the link distance could be over 100 km. If we select a longer distance node (such as 2-hop-away nodes), it may cause ultra-long links. This makes the radio propagation delay (at light speed) longer, and complicates the goal of synchronizing the multi-beam Tx/Rx transmissions due to the large link delay differences. (2) A node may only has 4 beams. If a beam needs to communicate with more than 2 nodes, it needs complex schedule control to avoid packet transmission collisions. (3) Too many nodes in the fence routing make the routing topology maintenance difficult. Therefore, here we only use 1-hop fence nodes.

![Figure 2.7: Establishment of fence routing.](image_url)
Additionally, in the fence routing, each node should maintain a routing table as shown in Table 2.1 for source nodes. Here, QoS (Quality of Service) is an advanced feature that prioritizes internet traffic for applications such as online gaming and live streaming to minimize the impact of busy bandwidth.

**Table 2.1: Routing Table**

<table>
<thead>
<tr>
<th>Dest.</th>
<th>Beam</th>
<th>Next Hop</th>
<th>Link W</th>
<th>QoS</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>2</td>
<td>C</td>
<td>0.50</td>
<td>1</td>
<td>Video Conference</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>B</td>
<td>0.30</td>
<td>3</td>
<td>Live Stream</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>C</td>
<td>0.50</td>
<td>2</td>
<td>Video on Demand</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>D</td>
<td>0.20</td>
<td>4</td>
<td>File Download</td>
</tr>
</tbody>
</table>

Note: every node keeps local sense of directionality. QoS can be introduced in the routing protocol.

**Link weight determination based on cross-layer design:** Since the link weight will be used to allocate the traffic amount in each link (more traffic will be assigned to the beam with better link conditions), we will need to define the link weight accurately. Here we use the following three parameters to define the weight:

\[
W = W_1 \ast RS + W_2 \ast \left( \frac{1}{PER} \right) + W_3 \ast \left( \frac{1}{ETX} \right),
\]

where

\[
W_1 + W_2 + W_3 = 1.
\]

Here \(RS\) is proportional to \(1/(d^2)\). PER represents the packet error rate and ETX (expected transmission count) is a router layer parameter, and has been defined in the literature such as [25]. The above weight definition reflects the cross-layer design nature of our routing design.

**Integrating Prioritized Fountain codes with Fence Routing**

The biggest advantage of a MBSA is its multi-link data delivery. By using our above proposed fence routing, we can significantly improve the throughput of the wireless network. Although throughput improvement is our first goal of the routing design, our second goal is to try to use MBSA-oriented routing to enhance the transmission reliability, i.e., reducing the packet
error rate (note that all packets with damaged bits will be dropped by the receiver). Although TCP-based Transport Layer could be used to achieve reliable transmission, it is built on frequent retransmissions, which may eventually sacrifice the throughput. Our solution here is to use fence routing’s diversified links to improve reliability and thus relieves the burden of the Transport Layer protocols. Particularly, we propose to use our previously invented prioritized Fountain codes (PFC) to enhance both throughput and reliability.

The basic PFC principle is as shown in Fig.2.8. It is a special rateless codes, and thus can automatically adjust the traffic rate based on the link quality. It is better than general rateless codes due to the control of redundancy in outer coding process: a higher priority flow has more redundancy to achieve a faster and easier packet decoding in the receiver. PFC is suitable to QoS-oriented applications.

![Prioritized Fountain Codes Diagram](image)

**Figure 2.8: Prioritized fountain codes.**

**Single-channel VS Multi-channel**

Today’s MBSA could operate in single-channel (i.e., all beams operate in the same carrier frequency), or in multi-channel status. In our fence routing design, we have considered those two
cases. Single-channel: If there is a single channel available in each link, all the beams of the same node need to strictly follow the MBSA consistency principle, i.e., all beams should be all-Tx or all-Rx status. If a beam finishes its Rx, it must wait for other beams’ traffic being finished. This makes the whole node act like conventional half-duplex mode: a node must first finish Rx before it switches to Tx. Since the whole column in the fence routing must have a coarse synchronization, this means all of those nodes in the same column must switch Tx/Rx almost at the same time. Multi-channel: Fortunately, today many networks have multiple channels available for communications. For example, an airborne network may use Ku-band (15GHz) to achieve long-distance communications. It divides the Ku-band into multiple channels (each around 200MHz). Thus a node can use part of beams for Tx while others for Rx as long as they use different channels. Of course, all beams that use the same channel still need to be in the same Tx/Rx mode. Obviously multi-channel makes a node act like full-duplex status, i.e., any beam can switch the mode (Tx/Rx) as long as it uses different channel. However, some rules still need to be followed in order to avoid transmission errors: (1) Angle overlap issue: If there exists radio interference across 2 hops due to the beam coverage overlap, different channels should be used if full-duplex needs to be achieved. As shown in Fig.2.9, node 1 uses a beam to talk with node 2. However, such a beam can cause interference to node 23 communication since they have coverage overlap. Unless node 2 still uses half-duplex state, that is, it does not send data to 5 if it is receiving data from 1. Otherwise, it must choose different channel (here we use Ch2) from 12 (it uses Ch1). This way it can simultaneously receive data from 1 and send data to 2. (2) Channel matching rule: This means that any beam-to-beam communications must use the same channel. This is an obvious rule to obey in any communication system.

**Cross-layer MAC and Routing Design**

To fully utilize the multi-beam concurrent transmissions of MBSAs, the above fence routing scheme should be integrated with the MAC layer schedule control. We recommend that the MAC layer use a TDMA-like schedule control between different hops. If the MAC is CSMA-based scheme, a coarse TDMA-based sublayer could be added above the CSMA. Since
the TDMA-like schedule uses much longer slot duration (could handle over 10 packets) than general TDMA time slots, adding such a higher sublayer is feasible. The cross-layer design is important for the following reasons:

a) Achieve a pipelined transmission: As shown in Fig.2.10, suppose Col-1 nodes are receiving data from S, Col-2 nodes can send data to Col-3, and Col-4 can send to D. The nodes belonging to the same column will have similar transmission delays after we allocate the traffic to each link based on the link weights. For example, if S-to-2 link’s weight is double of S-to-3 link, we can allocate double traffic amount in S-to-2. Because we use rateless codes that can automatically adjust the sending rate, the S-to-2 link would have approximately double sending rate than S-to-3, thus the transmission delay is approximately the same in those two links. Therefore, we could make all nodes in the same column achieve a coarse Rx/Tx synchronization. If Col-(i) is receiving data, it cannot send data to Col-(i+1). However, Col-(i+1) can send data to Col-(i+2) at the same time. Thus a pipelined column-by-column transmission is achieved.

b) Handle long-distance links: Due to the long-distance link, the propagation delay could not be ignored. For example, if a link is 100\(km\), it is 0.33\(ms\) of delay. For such a ‘long’ time, it is difficult to accurately detect transmission collisions if CSMA-based MAC scheme is used. To
avoid the waste of link bandwidth, some packets could be sent during such a long time. Therefore, some researchers [42] suggest to use TDMA-like scheme to send some packets each time in a Tx phase, and then switch to Rx phase to receive some packets. Since the TDMA slot time is much longer than conventional slot definitions, it cannot be called real TDMA scheme (instead, it is called TDMA-like scheme). Additionally, the time boundaries between Tx/Rx do not need to be accurately determined. In other words, there may exist certain Tx/Rx starting time deviations among different nodes in the same column. We call such a situation as coarse synchronization.

A higher MAC sublayer can be added above the CSMA to achieve a coarse TDMA-like Tx/Rx switching control. For example, in [42] it has proposed the overlay network concept to use long time interval in the higher sublayer.

c) Use the multi-beam feature for reverse data transmission: In Fig.2.10, when Col-1 nodes are receiving data from S by using left half of its beams, the right half of its beams could be in idle (I) status; Or, they could receive data sent in reverse direction (either ACK packets or reverse data sent from the receiver). We use status ‘B’ (back) to mean reverse direction data.

**Bandwidth Allocation Policy**

The bandwidth allocation policy is defined as a linear programming(LP) problem with certain network and interference constraints [57]. We model the fence topology as a directed graph $G(V, E)$. For simplicity, we assume the largest width in the fence routing topology is 3,
which means that in each column there exist a main path node and 2 side path nodes (one in each side of the main path). Thus for each node \( v \in V \), there are at most three beams in active Tx or Rx mode. We denote a beam \( i \) of node \( v \) as \( v_i \). For any link \((u,v) \in E\), the flow rate or average throughput is denoted as \( f(u,v) \), and \( f(u_i,::) \) denotes the outgoing flow rate from beam \( i \) of node \( u \); \( f(::,v_j) \) denotes the incoming flow rate to beam \( j \) of node \( v \). If node \( v \) is within the range of beam \( i \) in node \( u \), i.e. \( v \in R(u_i) \), the flow rate \( f(u,v) \) can also be represented as \( f(u_i,v) \). Assume there is a lower bound \( f_0 \) for the flow rate at each beam of the nodes in the fence routing topology.

Additionally, we need to consider the radio interference. Our objective for bandwidth allocation is to maximize the overall throughput while minimizing the interference in the network with multi-beam antennas. We can now formulate the LP problem as follows:

\[
\text{max} \sum_{S_i \in S} f(S_i,:) - IF \\
\text{Subject to} \\
\sum_{S_i \in S} f(S_i,:) = \sum_{D_j \in D} f(::,D_j). \\
f(u_i,:) = f(u,v) \quad \text{for} \quad v \in R(u_i). \\
f(::,v_j) = f(u,v) \quad \text{for} \quad u \in R(v_j). \\
\sum_{A_i \in A} f(A_i,:) = \sum_{A_j \in A} f(::,A_j) \quad \text{for} \quad A \in M. \\
f_0 <= f(u,v) <= C_e \quad \text{for} \quad (u,v) \in E. \\
\text{Size}(B) <= 3 \quad \text{for} \quad B \in V. \\
f(u,v) >= f(u,w) \quad \text{for} \quad (u,v) \in ME, (u,w) \in ME. 
\]  

Here, \( IF \) represents total interference within network, \( C_e \) denotes the link capacity, \( M \) represents the nodes in the middle of the fence except for source \( S \) and destination \( D \). And \( ME \) is a link of the main path, which is supposed to have more bandwidth than the links of the side path. \( \text{Size}(B) \) represents the number of beams in use for node \( B \) when \( B \) is transmitting or receiving.
Overall high-level network routing protocol

We provide the main operations of the routing protocol among the high-level nodes as follows:

**INPUT:** Node IDs, beam IDs, Channel IDs, Traffic QoS parameters (delay deadlines), link rate and network capacity in each beam, source traffic amount and rate, antenna beam angle, node mobility speed;

**OUTPUT:** Fence routing topology, Tx/Rx schedule

**Protocol Part 1 - Main/side paths establishment**

1. Using the enhanced ORRP algorithm described before to establish a GPS-free main path.

2. Every main path node $h_i$ in stage $i$ widens the path to construct side paths by searching neighbors in both sides of the main path via neighbor discovery scheme. Also a upper bound $V_{max}$ and lower bound $f_{min}$ are used to rule out the neighbor nodes with low flow rate and high mobility.

3. A side node in current column (also called stage), $i$, can be included in the neighbor list of the main path node $h_i$, as long as it is in one-hop range of the main path node. Suppose the list size is $N$. For the desired stage width $W$ (it means that there could be the maximum $W$ nodes in one stage), if $N > (W - 1)$, we could select $(W - 1)$ side nodes with relatively higher sending rate and lower mobility among all $N$ nodes. If $N < (W - 1)$, then the width for this stage needs to be reduced. For example, assume that the desired stage width is 4, however in stage $i$, there are only 2 side nodes in the list. Then in stage $i$ the width has to be reduced to 3. Therefore there may be different widths for every stage.

4. Building all the stages until reaching the destination node $D$. Every node in stage $(i + 1)$ should be within the communication range of all the nodes in stage $i$. This is also the reason that we cannot use a too high value of $W$. Otherwise, a node in stage $i$ may not be able to read a stage $(i + 1)$ node in the diagonal direction.
Protocol Part 2 - Beam orientation adjustment

5. Based on the steps above, the stages are constructed to form a fence structure. Then for a MBSA of any node, the beam direction should be adjusted properly to establish the links between the nodes in two consecutive stages. A node $A$ in stage $i$ keeps sending probing messages through its beams to the nodes in stage $(i+1)$, if a node $B$ in stage $(i+1)$ receives the probing message, it sends back an ACK message. Thus the link between $A$ and $B$ is established, and the beam of $A$ stays towards $B$.

Protocol Part 3 - Beam table maintenance

6. After the beams’ directions are determined in the above step, the beam table will record the node ID within each beam’s range. In addition, in the probing ACK message, each node of stage $(i+1)$ feedbacks its information such as node ID, reception beam ID, signal strength, queue size, etc.

7. Each node in stage $i$ also puts its own information into the beam table including the beam ID, desired destination ID, beam range, queue length, traffic QoS, traffic type, traffic priority, etc.

Protocol Part 4 - Traffic Control

10. For any node, all the beams should have coarse time synchronization due to the concurrent packet transmission (CPT) and concurrent packet reception (CPR) requirement for all the beams of a MBSA. Between the stages the transmission (Tx) /reception (Rx) modes alternate, i.e. if stage $i$ is in Tx mode, then stage $(i+1)$ should be in Rx mode, and stage $(i+2)$ should be in Tx again.

11. To be compatible with TDMA-like protocol in MAC layer, the time interval for each Tx/Rx mode should be determined based on the traffic amount in each beam and the link speed. The goal is to ensure that the entire stage has coarse time synchronization.

Protocol Part 5 - QoS Control

12. For each node, the links in its beams are assigned different priorities based on the link quality (measured by weights). The links with higher priority are assigned with more traffic
amount.

13. To avoid traffic congestion, the nodes in stage \((i + 1)\) periodically inform the nodes in stage \(i\) about the queue size.

### 2.5 Moth-Inspired Low-Level Network Routing

#### 2.5.1 Problem Statement

While the high-level network has even mobility (i.e. the nodes have similar moving speeds), the low-level network has **singular mobility**, i.e., the sink node typically has much higher mobility than other nodes. This is mainly due to its global network data collection requirement.

The low-level nodes have much less mobility than high-level nodes in the H2WN. For example, in an typical airborne network, UAVs move much more slowly than aircrafts. And many of them have almost stationary nodes. For example, in a wireless sensor and actuator network, the sensors do not move much. Let’s first consider the case that the sink does not have much mobility. A simple *gradient routing* could well support the node-to-sink communications. As shown in Fig.2.11, the sink can simply broadcast an announcement message to the global network in order to know how many hops away each node is. Since the sink does not move much, it only broadcasts such a message occasionally. Then any node can simply forward data to an inner circle neighbor, and eventually reaches the sink.

However, if the sink moves much more quickly than other nodes, such a gradient-based routing does not work well since an established gradient map quickly becomes invalid when the sink moves to a new place within a short time.

Conventional DSR-like routing can not efficiently handle *singular mobility*. Since DSR uses blind, global RREQ broadcasting, a source node needs to frequently broadcast such RREQ messages in order to keep track of the fast moving sink.

Here we assume a GPS-free network. Thus the geographical routing based schemes will not work here either.
2.5.2 Moth’s Target Approaching Behavior

Moth has a special way to pursue the light source. It first detects the approximate direction of the light source (it has an acute thermal detection capability to find such a direction). Then it uses 3 steps to locate the light (Fig.2.12): (1) Fast, straight approaching: It first uses a diving speed to quickly approach to the target. (2) Zigzag trajectory: When the moth feels that the target is not far away, it slows down its speed and carefully flies toward the target with the zigzag style. (3) Circular fly to lock the target: When the moth feels that the light source is close enough, it uses a circular trajectory to get closer and closer to the target until finally it locks the exact location [40].

Scientists believe that the moth’s innate capability is the best way to pursue an uncertain target. Since the moth never thinks that the target has fixed location, it adopts the above finer and finer search process. This behavior is very much like the spaceship’s flying style: it first uses the highest speed to fast approach to a planet. Then it slows down and flies circularly in the orbit of the planet until landing. In our daily lives, we drive so fast in the highway, and slow down quite a bit until parking in the destination.
2.5.3 Moth-Inspired Event-to-Sink Data Delivery

Inspired by the above moth’s target searching behavior, we propose a line-fan-circular (LFC) routing path establishment scheme (Fig. 2.13). The entire routing path consists of 3 parts: (1) *Line segment (LS)*: In this section of path, the event node (E) uses a single path to relay the data in hop-by-hop. (2) *Fan segment (FS)*: in this phase, the last node of the line section uses DSR-like RREQ broadcasting to search for the neighborhood of the sink. (3) *Circular area (CA)*: This is the neighborhood of the sink. When a FS node intersects with any of the CA nodes, it stops FS, and the intersection node uses gradient path to directly reach the sink.

Due to singular mobility, the sink has a quickly changing neighborhood. It is not beneficial to maintain a gradient routing map for the entire network since a node may never reach the sink if following an outdated gradient path. Therefore, we only ask the sink to maintain a very small gradient map (maybe just 2 hops away), see the circular area of Fig. 2.13. Suppose a sink moves at
speed of 100 meters per second. Assume that it broadcasts a message to its neighborhood every 10 seconds. Since the UAV communication distance can be up to 10km, the sink only leaves its original location for 1km within 10 seconds. The CA established last time is thus still valid. Therefore, as long as a packet reaches one of the CA nodes, it can be delivered to the sink.

Whenever the sink wants to establish a new CA, it will broadcast a CA\_Fresh message to its neighborhood. There is a field in the header called CA\_TTL (time-to-live). Suppose it is set to 3. Each time a node receives the CA\_Fresh, it subtracts 1 from CA\_TTL. If it is zero, the node knows it is in the outer circle of the CA and stops further broadcasting.

Any node can at least use a part of the DSR-like path to reach the sink. Here we assume that the sink broadcasts the RREQ to the global network at a reasonable rate. After a complete DSR process through the network, all nodes will get to know the approximate gradients to the sink. Especially they will know the shortest path to reach the sink (i.e. along the gradients). Even if the sink quickly moves away, at least the first part of the path will bring the data to the correct direction to the sink.

This fact motivates us to use LS (Fig.2.13) to fast approach to the CA. The length of the LS depends on the empirical values. Its value is a trade-off between the following factors: the network scale, how many hops between the source and sink, the sink mobility, and the maximum 1-hop communication distance.

The FS is needed to reach the CA. In the moth’s trajectory, there is a zigzag section between LS and CA. Here we make a light change: we use a short section of fan-like message broadcasting in FS section to search the CA. As long as the message reaches a node of the CA, the node immediately sends back a CA\_FOUND message to the source.

Such a FS is very short compared to the LS. If LS is updated due to a new round of global DSR process launched by the sink, the corresponding FS path should be re-built. If the intersection node (with the CA) is not available anymore due to the sink’s new CA announcement, the FS should also be updated to find a new intersection node.
2.5.4 Handle Hybrid Uni-/Omni-Directional Antennas

Our targeted H2WM assumes a hybrid antenna system. While the high-level nodes could be equipped relatively more expensive MBSAs (i.e., multi-directional antennas), the lower level nodes could certainly have the cheap omnidirectional antennas, as well as the inexpensive uni-directional antennas (often just called directional antennas). Today, a common Wi-Fi device could be equipped a omnidirectionalantenna and a directional antenna. Their main differences are shown in Fig.2.14. As we can see, the directional antenna controls the transmission direction in a small angle (thus has higher antenna gain in that direction). By rotating the beam, it can change the direction. It has longer communication distance than the omnidirectional antenna. As in other directional network studies, here we also assume that the directional antenna can both send (Tx) and receive (Rx) data. However, the omnidirectional antenna is used for data reception (Rx) only. The reason is simple: the omnidirectional antenna does not have high antenna gain and its signal propagation distance is shorter than directional antenna. Thus it is not suitable for Tx if directional antenna is already installed. However, it can operate in Rx due to the antenna’s high sensitivity (it can receive signals with only -70dB). If the sink wants to run DSR protocol, it needs to quickly rotate its beam (called sweeping) to send out RREQ in all directions (see Fig.2.14(c)). Once a node determines its next-hop node, it can rotate its beam to next-node direction in order to efficiently send out data. While other relay nodes may just use omnidirectional antenna to receive the data, the sink can use directional antenna to receive the data from a 1-hop node in its CA. In directional routing scheme, an issue must be solved called “beam overlap” (see Fig.2.14 (d)). The beam of node 1 may cause strong radio interference with 35 link since all nodes are almost in the same line. To overcome such an issue, we require that each node report its received interference signals periodically in the entire event-to-sink path. If the interference source comes from previous hops, the node that causes the interference (here it is node 1) must launch a ‘local repair’ process to find alternate relay node that does not cause beam overlap issue. Here nodes 4 and 6 are found to replace nodes 2 and 3.

Mobility-Prediction based beam adjustment: Although the nodes in low-level network
Figure 2.14: Hybrid antennas in Lower-level Network.

have much smaller mobility than the sink, they may have certain movement. For example, each UAV may fly at a speed of >10m/s. If the LFC path is broken due to the node mobility, the path has to be locally repaired. However, many packets may get lost if we wait for the detection of link breakage and path repair. Since the nodes have small mobility and the low-level nodes may perform a mission-oriented mobility (for example, an UAV targets a small surveillance area), we propose to use a prediction-based scheme to prepare a backup node for seamless path repair.

Figure 2.15: Prediction-based local repair

Mobility prediction is difficult in GPS-free network. However, even without accurate node
location information, we can still make a simple prediction: whether or not a node will leave the beam coverage of the upstream node. This is because any node knows the RSS (received signal strength) based on its power level measurement in the transceiver. Based on free-space path loss equation, such a RSS is reversely proportional to the distance between two nodes (Fig.2.15(a)). In directional antenna case, the RSS value quickly approaches to zero once the node leaves a beam coverage. However, before it moves out, the RSS still follows the path loss equation, and thus has a smooth decaying curve. When it gets closer to the beam boundary, due to the stronger energy leaking signals from a neighboring beam, RSS also decays, but with a faster speed than distance-caused fading. Therefore, we propose to use a good time series model, ARMA, to predict when the RSS will be in the next time unit. Fig.2.15(b) shows the basic principle of ARMA-based prediction. It consists of AR and MA components to reflect the predicted new value as well as prediction errors. It gives a confidence range of the predicted values depending on the error tolerance levels (called prediction limits, see Fig.2.15(c)). Once the RSS decays into the red zone, it means the node is not reachable anymore. In our RSS-prediction based scheme, once it get closer to a threshold (still outside the red zone), the node quickly sends back a LEAVING message to its upstream node (see node 1 in Fig.2.15(d)). Then the upstream node will trigger the local path repair process, similar to the previous ‘beam overlap’ case.

2.5.5 Moth-Inspired Routing Algorithm

Here we provide the big picture of the moth-inspired routing algorithm.

**INPUT:** node IDs, average node mobility $V_s$, sink mobility $V_d$, sink announcement round ID $N_L$, sink announcement period $T_d$, link capacity $L_C$ and rate $L_R$, source traffic rate $R_S$, radio reaching range $R$, LS hop number $M_1$.

**OUTPUT:** Moth-inspired routing topology

**Periodic sink existence announcement:** The sink periodically broadcasts its information such as the current location and node ID to the network, by using DSR protocol (or other reactive ad hoc routing schemes). The nodes receiving the sink’s message update their neighbor lists and routing tables to build the gradient routing paths. Those nodes use gradient routes to connect with
the sink in the CA.

**Route partition with mobility:** The whole route is partitioned into three sections (LS, FS, and CA). In LS, the hop number $\#M_1$ depends on node mobility rates $V_s, V_d$. The nodes within the first $M_1$ hops starting from source node are selected from the DSR-based path to form the LS. The end node $A$ in LS is also regarded as the starting node of the FS. Node $A$ uses blind message broadcasting to search for the target nodes $B_1, B_2, \cdots, B_n$ located in CA. One of the target nodes $B_i$ is chosen as the intersection node between FS and CA, if the path from $A$ to $B_i$ is the shortest among all the paths originated from $A$. By using gradient routing, node $B_i$ can reach the sink with singular mobility.

**Routing path update:** Every $T_1$ seconds, the sink broadcasts a *LocationUpdate* message to nearby nodes (within $M$ hops), to build a new CA. If the target node $B_i$ is not in the new CA anymore, a new search of FS needs to begin to find a new target node $B_j$. Every $T_2$ seconds ($T_2 > T_1$), the source triggers a new round of DSR to establish a new LS.

### 2.6 Ant-Inspired Cross-Level Routing

#### 2.6.1 Problem Statement

Although the sink could serve as the data aggregation point to collect event data from any low-level node, in many $H^2WNs$, one of the high-level nodes (called commander node), plays a more critical role than the sink since it can send commands to any low-level node (using downlink communications). It also collects emergency reports from any low-level node (using uplink communications).

Although downlink communication (commander-to-UAV) is relatively easier due to the powerful long-distance communication capability of the aircrafts, the uplink communication (UAV-to-commander) needs to adopt the principle of “the closer, the better” to save the transmitter’s power. This is because an UAV is designed to communicate within a short-distance link ($< 10km$). It needs to use a high antenna power to reach a high-level node. Therefore, if it needs to talk with the commander, it must first forward the data to a UAV that is closest to the
commander, and then asks that UAV to relay the data to the commander.

Suppose a low-level node $A$ wants to send data to a commander. Assume in the beginning it can send signals to the commander that is right above it ($10km$ higher). Then the commander moves away for $100km$. Now $A$ needs to propagate the signal for a much longer distance. Obviously, this is not realistic to a UAV with limited antenna power. A better approach is to use a node that is closest to the commander, to help to relay the data to the commander.

![Figure 2.16: Cross-level communication in the network.](image)

Therefore, always finding a closest node plays a critical role in UAV-to-commander uplink communications. Although some existing cross-level protocols are introduced to deal with the coordination issues between the high level and low level, however these cross-level schemes are not designed for hierarchical airborne networks and they cannot deal with the high and singular mobility in the network. Also even though some of them take the mobility into consideration, they cannot cooperate with the hybrid moth-inspired protocols in the low level and also they cannot exploit the benefits of multi-beam smart antennas in the high level. In this research, we propose a cross-level routing scheme that can not only ensure the shortest uplink distance, but also record the commander’s latest locations (thus we can always keep track of the commander’s moving trajectory).

### 2.6.2 Ant-Inspired Cross-Level Routing

Ants have the innate capability to collaborate to find food. As shown in Fig.2.17 when an ant finds a food source, it gets back to the nest and leaves a special chemical material called
pheromone. Other ants will follow such a trail to reach the food source. When more and more ants follow the same trail, each of them leaves certain amount of pheromone trail, and thus such a trail is enhanced. If the food is already stored in the nest, the pheromone evaporates and eventually disappears. Then no ant will follow such a trail any more.

![Ant Nest](image)

**Figure 2.17:** The illustration of ant-inspired cross-level routing.

Inspired by the pheromone trail phenomenon [50], we propose to use the low-level nodes to record the trail of the commander node of the high-level network. Our idea is motivated by the following facts: (1) the low-level network is much denser than the high-level network. So many swarming UAVs can well record the trajectory of an aircraft. (2) The high-level node can easily reach the low-level node through its powerful antenna (a MBSA). Each time the commander moves, it can broadcast a TRAIL message to the low-level network. Any node receiving such a message will keep a record in its commander tracking table. Figure 2.18 illustrates the basic idea. Each trail node maintains a table containing beam direction, decaying factor, trail timestamp, next-node, etc.
Figure 2.18: The illustration of ant-inspired cross-level routing.

**Ant-inspired cross-level routing protocol:** Here we provide the big picture of ant-inspired UAV-to-commander routing scheme.

**Input:** node IDs, link rate $R_U$ on the upper layer, link rate on the lower layer $R_L$, antenna signal range $R$, the shortest distance between the low- and high-level network $h$, commander announcement period $T_P$;

**Output:** trail list in the lower level, cross-level routing path.

**Step 1:** The commander node $N_c$ periodically sends the TrailNotification message with its ID and timestamp to the nodes in the lower layer. If a node $L_i$ in the lower layer receives the message from the commander node, then it creates a trail table to store the information including timestamp, last trail node ID, etc. Also node $L_i$ periodically broadcasts the trail table to its neighbors in the lower layer. Any node that hears the broadcasted trail table and is in the trail will add a “shortcut pointer” to its trail table to indicate that it can directly reach the ending node in the trail.

**Step 2:** If a node in the trail wants to send data to the commander node, it will first search “shortcut pointer”. If the pointer exists, it will immediately forward data to the ending node in the trail. Otherwise, it initiates a route discovery to find a closest trail node. It does this by following general DSR process: it broadcasts a RREQ message to nearby nodes. If a node that received the RREQ and is in the trail, it feedbacks the FoundTrail message to the source.

**Step 3:** Decaying of the trail: if a node finds itself has not received the new
TrailNotification message for a duration that is longer than a preset threshold $T_{trail}$, it sends out a LeavingTrail message to nearby trail nodes, and leaves the trail.

2.7 Performance Evaluations

2.7.1 Fence Routing with MBSA

The case of airborne network

We first simulate a two-level airborne network, where the higher level is a sparse aircraft network with at most 5 hops between any source and destination. All the high-level nodes are randomly located in a area of $300km \times 300km$. The average link distance is $50km$, and the radio transmission range is less than $90km$ for a MBSA. The average transmission speed is 10 Mbps for the main path links, and 5 Mbps for the side path links. The average packet size is 1500 bytes. Every node has a buffer that can hold at most 200 packets. Each MBSA has 4 beams. We compare the throughput and delay of fence routing with diamond routing [13] and single path routing (DSR).

The throughput and end-to-end delay are shown in Fig.2.19 and Fig.2.20, respectively. The packet generation rate ranges from 0 to 1000 packets/s, so does the corresponding throughput. The throughput of fence routing is higher than diamond routing and conventional multiple path routing. The throughput of fence routing can almost reach the average packet generating rate, which indicates a negligible packet loss rate. The throughput becomes steady at around the generation rate of 1000 packets/s. But for diamond routing and single path routing, the steady throughput is only about 400 packet/s and 200 packet/s, respectively. Regarding the end-to-end delay, fence routing has much less delay than other two schemes.

We then study fence routing with the maximum width = 2 in each column. This means that in each column/stage, besides the main path node, only one side node can be added to the fence topology. Fig.2.21 and Fig.2.22 show the throughput and delay. Again, fence routing can improve the network performance compared with single path routing and diamond routing.
Figure 2.19: Average throughput achieved by fence routing (width =3).

Figure 2.20: Average delay achieved by fence routing (width =3).
Figure 2.21: Average throughput achieved by fence routing (width = 2).

Figure 2.22: Average delay achieved by fence routing (width = 2).
**General two-level wireless network**

Here we simulate a general network that can have up to 10 hops between the source and destination in the high-level network with MBSAs. The link state is 1Mbps for the main path. Also the average link distance is 1km and the antenna transmission range is about 1.8km in each beam.

Fig.2.23 and Fig.2.24 show the throughput and delay with width = 3 (i.e. there are 3 nodes in each column of the fence routing). Again, the performance of fence routing is much better than other two schemes.

![Figure 2.23: Average throughput achieved by Two-level network (width = 3).](image)

**2.7.2 Moth Routing in the lower level**

In the lower level of the hierarchical network, we compare the performance of three routing protocols, including conventional static routing (DSR without path updating scheme), dynamic routing without considering singular mobility of the sink node, and our proposed moth routing. For the conventional mobile routing scheme(DSR), it searches for the new path again
every 40 seconds. In the moth routing protocol, the sink broadcasts its message to its 2-hop neighbors every 10 seconds. And the packet generating rate at the source is 25 packets/s. Assume there are 5 hops in the LS of moth routing. The link capacity is 0.5 Mbps and the average packet size is 1500 bytes/s.

Fig.2.25 shows the end-to-end throughput for different destination/sink speeds. At a low speed such as 0.1 km/s, the conventional mobile routing seems to achieve better performance than the other two protocols. This is because that moth routing may construct a non-optimal path compared with conventional DSR-like protocol. However, with the increase of the sink speed, the throughput of conventional mobile routing decreases significantly, while our proposed moth routing can still keep the throughput above 18 packets/s.

Next we study the impact of the route updating period on the performance. Assume a sink speed of 0.3 km/s. From Fig.2.26 we can see a smaller updating period brings a higher throughput. We can see that 10 s is the best route updating period for moth routing with a sink
speed of $0.3 m/s$, and the throughput can reach $15$ packets/s. For the same sink speed, moth routing can achieve higher throughput than conventional mobile routing. Sometimes the curve fluctuates sharply, this may be because of the randomness of the mobility such as speed and direction in our simulation.

**Different Update Periods for Moth Routing:** As we can see from Fig.2.27, in low sink mobility such as $0.1 km/s$, updating period has small impact on the performance of moth routing. When the sink speed increases, updating period becomes more influential. From Fig.2.27, we can see that $10s$ is a good choice of updating period for moth routing and achieves a throughput of $18$ packets/s.

**The impact of different LS lengths:** In moth routing, the first section is a straight line (LS) which helps to quickly approach to the sink’s direction. From Fig.2.28 we can see that there exists a threshold for the number of hops in LS. In this particular example, if more than 6 hops are used in LS, the throughput will decrease significantly, also the delay will increase sharply. This is
Figure 2.26: Performance of moth routing.

Figure 2.27: Throughput for different updating periods in moth routing.
because we assume at most 13 hops are needed based on our network topology, if too many hops are used in LS, there is a high chance that the end of the LS route deviates from our destination node or even misses it.

Figure 2.28: Moth routing performance for different number of hops in line section.

The impacts of different CA lengths: To study the proper size of the CA in moth routing, we evaluate the performance of the overall network for different hops in this area. Fig.2.29 shows the average throughput for 3-hop CA, 2-hop, and 1-hop, respectively. We can see that the CA with 2 hops can obtain the best throughput. Thus in our routing protocols, the 2-hop neighbors are maintained in the CA for the gradient routing.

2.7.3 Ant-inspired Cross-Level Routing

In this section we will validate the ant-inspired cross-level routing. We assume that there is a commander node \( C \) in the upper level and a source node \( S \) in the lower level. And originally the source node \( S \) is right below the commander node \( C \). This means that node \( C \) is in the direct communication range of node \( S \). But after some time, node \( C \) may fly out of the range. We have
proposed an ant-inspired, cross-level routing scheme for the communications between $S$ and $C$. In the simulations, we consider a movement area of $20km \times 20km$ for both levels of networks. There are 10 nodes in the upper level and 100 nodes in the lower level. We assume that at first the commander node flies in the direction that is 45 degrees deviating from X axis. Then the commander adjusts its direction randomly with at most $\pi/16$ deviation from its original direction. We keep a trail list for the commander node by resorting to the nodes on the lower level. Fig.2.30 shows the number of hops in the routing path. We can see that as the commander node’s speed increases, the hop number also increases (assume no shortcut scheme is applied). The hop number reaches 15 when the commander speed is $0.3km/s$. By using shortcut path, we can see that the hop number can be decreased to only 3 or 4 hops. Therefore with shortcut scheme, the complexity of our cross-level routing can be significantly decreased. We can also see that the commander’s speed has little impact on the hop number with shortcut scheme.

**Compared with other cross-level routing schemes:** As discussed in Section II, there are
already some existing cross-level routing protocols besides our proposed ant-inspired cross-level routing. In our simulation, we use the same upper level and lower level routing as mentioned above. However different cross-level routing protocols are introduced for the airborne network to compare our proposed cross-level routing with the protocols in wireless sensor and actuator networks and wireless robot networks which both have their own cross-level schemes [50]. Fig.2.31 shows the packet loss rate for different protocols. We can find that in the airborne networks, our proposed moth routing can ensure the packet loss rate below 0.2 no matter how high the sink mobility is. However for the other two cross-level schemes, we can see that the packet loss rate can be pretty high while the sink mobility increases. Fig.2.32 presents the event-to-commander delay in the airborne network with different cross-level protocols. We find that the delay for the network wit our cross-level routing is much lower than the delay for the other two networks. This means that our proposed ant-inspired cross-level scheme can ensure the high performance and exploit the benefits of the multi-beam smart antennas for the hierarchical
airborne networks.

![Figure 2.31: Packet loss rate for different cross-level routing protocols.](image)

![Figure 2.32: Delay for different cross-level routing protocols.](image)

### 2.7.4 Overall Network Performance

In order to evaluate the end-to-end routing performance from a low-level source node to a high-level command node, we have integrated all the three proposed routing algorithms together, which include (1) the moth-inspired low-level UAV-to-UAV routing, (2) ant-inspired UAV-to-aircraft (low-to-high level communications), and (3) neuro-inspired fence routing for high-level aircraft-to-aircraft routing. And we have put significant effort to conduct new simulations for the end-to-end throughput and delay in the hierarchical network. Also, we
compare our end-to-end routing algorithms with the existing hierarchical routing protocols such as mobile routing protocols (AODV) and static routing protocols. Because the high-level aircraft network has much longer link distance (over 50km) than low-level UAV networks (less than 10km), we assumed 3 hops of the fence routing in the high level with the link speed of 10 Mbps. Because the link speed in the low level UAV network is just 0.5 Mbps, the throughput of the low level network becomes the bottleneck for the overall network.

Fig.2.33 and Fig.2.34 show the simulation results for three different protocols for the hierarchical networks (our proposed scheme and other two schemes): (1) Our Routing: This is our proposed end-to-end routing scheme that consists of three algorithms in the overall network including moth routing in the lower level, ant-inspired cross-level routing with shortcut, and fence routing in the higher level. (2) Mobile routing: This is a scheme that we will compare with. In this scheme, we apply mobile routing protocols in the whole network including conventional AODV-based mobile routing in the low level, multi-path routing in the high level and also our proposed cross-level routing. (3) Static routing: This is another scheme that we will compare with. In this scheme, we apply static routing protocols in the network which consists of static routing in the low level, multi-path routing in the high level and our proposed cross-level routing.

In Figs.2.33 and 2.34, on the X-axis, the gateway node means the low-level node that can directly connect to the closest node on the high-level network. This gateway node helps to send the data from the low-level UAV network and the corresponding commander node in the aircraft network. Thus the mobility of the gateway node is critical since it determines the ant-inspired trail dynamics in the low-level network. We can see that our proposed routing has higher throughput and lower delay than the conventional protocols, especially when the gateway node is in high mobility speed. Our scheme can keep the throughput more than 10 packets/s even when the speed of the gateway node reaches as high as 0.3 km/s, however the throughput for the conventional protocols are significantly compromised in such a mobility speed.

**Scalability of the routing scheme:** In order to study how scalable the proposed algorithms are, we have conducted new simulations for the hierarchical networks with different
Figure 2.33: Throughput for the overall network.

Figure 2.34: Delay for the overall network.
size to see if our proposed protocol can still take effect in large-scale networks.

We first increase the number of the aircraft nodes (from 25 to 50) in the high-level network, and enlarge the area from 300km x 300 km to 500km x 500km. Then we conduct the simulations on the new network topology for our proposed fence routing, diamond routing and conventional multi-path routing. The figures below show the performance of the new network topology. We can see from Fig.2.35 and Fig.2.36 that for the new large-scale network, our proposed fence routing can still get better throughput/delay performance than the conventional diamond and multi-path routing schemes.

![Figure 2.35: Throughput for new topology.](image)

Then we increase the number of nodes of low-level network from 100 to 225 and also enlarge the size of the network topology. Again we conduct the simulations for this new topology for our proposed moth routing, conventional mobile routing and static routing respectively. Fig.2.37 shows the simulation results for this scaled network topology. We found that our proposed moth routing still achieves better performance than the other two protocols in
Figure 2.36: Delay for new topology.

large-scale UAV network.
Figure 2.37: Throughput for different protocols for new topology.
Algorithm 1 Main path search based on enhanced ORRP scheme

**Input:** Source Node ID $S$, Destination Node ID $D$, two pairs of orthogonal Beam IDs

**Output:** The main path $P(S, D)$ of the fence routing

**Part One:**

1. $D$ periodically broadcasts ORRP announcement messages along the orthogonal directions from the 2 pairs of orthogonal beams.
2. Each neighboring node that received the messages stores the Destination ID $D$, the neighbor ID in the previous hop, the hop count and its Beam ID $BID_1$ from which the packets came in.
3. Each relay node forwards the announcement packets using the beam in the opposite direction.
4. Proactively, these relay nodes and destination node maintain the rendezvous-to-destination paths.

**Part Two:**

1. $S$ sends out the route request packet (RREQ) in its 4 orthogonal beams.
2. The neighboring node that receives the RREQ packet stores the Source ID $S$, its previous hop, the hop count and beam ID ($BID_2$) from which it received this RREQ, then forwards it to the next hop using the beam in the opposite direction.
3. The subsequent nodes keep forwarding the RREQ along the orthogonal directions in the same way until a rendezvous node $R$ specified in Part One receives the RREQ. This intersection node $R$ stops forwarding and sends out a RREP containing the route to destination node $D$ back to node $S$ in the reverse direction.
4. After $S$ receives the RREP packet, it establishes the source-rendezvous-destination path.

**Part Three:**

1. Based on Part Two, $D$ rotates the beam facing the intersection node $R$, for about 45 degrees and then repeats the steps in Part One.
2. $S$ also rotates the corresponding beam for about 45 degrees in the internal direction and repeats the steps in Part Two.
3. There exists a new rendezvous node $R'$, and $S$ can achieve a shorter path which can be taken as the main path $P(S, D)$ in the fence routing.
CHAPTER 3
DIRECTIONAL WMN TESTBED WITH QOS APPLICATION

3.1 Introduction

3.1.1 Directional mesh network (DMN)

During the latest decades, wireless mesh and ad-hoc network have become more and more popular not only within the academia but also in our everyday life. Nowadays the majority of the existing networks need a access point acting as a center to operate in the infrastructure mode. And this means that the wireless nodes within the different endpoints are not allowed to communicate with each other directly. Now increasing users are engaged in the WIFI network, which makes the processing center really occupied and the efficiency may be drastically compromised. Therefore, the wireless mesh and ad-hoc network is a potential solution to the explosive increase of wireless network occupancy because it enables the nodes within a local network to interact with each other without any central coordinator. Therefore it helps to expand the coverage of network as much as possible. This can further improve the quality of wireless communication within busy public areas. Another advantage of mesh networking is that it can help to broaden the range of wireless network by using multi-hop relaying routes. On the other hand mesh network can work as an alternative network in the case of natural disasters such as tornado and earthquake because WIFI network is quite vulnerable to these extreme circumstances. However, with automatic and dynamic routing, mesh network may get rid of the influence of changes in network topology such as node damage or network extension.

Normally the idea of mesh network is originated from multipoint-to-multipoint communication, which has better performance than the simple point-to-point and point-to-multipoint communication. In order to exploit the benefits of mesh network, there comes
the idea of directional mesh network which cooperate with directional antennas. Directional mesh network has a lot of advantages, though, it’s quite difficult to design a robust and scalable mesh network with high performance and low overhead. The main challenge is about how to design a steady mesh routing protocol with directional antennas under the circumstances of volatile links, varying topologies and limited bandwidth.

As we all know, there are basically two kinds of antennas, namely, omni-directional antennas and directional antennas in terms of directivity. For an omni-directional antenna, it functions equally in any direction and its radiation pattern looks like a doughnut. For directional antenna, it functions more efficiently in some direction than others. So the directional antenna enhances the energy in some direction and increases the scope of radiation in this specific direction. Therefore when applied in the field of communication, the directional antenna can greatly improve the efficiency of data transmission, extend the range of communication and prevent the interference from other sources. Also now the antenna technologies have achieved a lot of progress. There comes all kinds of directional antennas. There is a new type of directional antenna called multi beam smart antennas which means extending general directional antenna from one direction and multiple. It enables concurrent multi-beam transmissions using the same channel which can significantly improve the throughput of wireless mesh network compared with conventional directional antenna which can only transfer in one direction for a specific node at one time.

In our work, a mesh router integrating with a directional antenna constitute the ground and kernel for our directional wireless mesh network. Therefore, in order to study and experiment on the network protocols and properties, the concept of DMN testbed is proposed to help the investigation and design of wireless mesh network with directional antennas. This DMN testbed can not only support the utility for hardware assembly and debug, but also provide the platform for software development and performance test for network design and implementation. There are also some existing DMN testbeds such as SoftMAC and OpenMesh which have some common drawbacks. For example, they are not generic platforms, for example OpenMesh takes a very
complex process to modify part of the MAC protocols. And this testbed is implemented on PCs and cannot be applied in the embedded systems such as routers which are more portable and generic. Also these existing testbeds are normally very expensive. For example, some testbeds use USRP boards, which are open for network development, however, this USRP board is too much expensive, which makes it impossible for the routing protocol test in large scale networks. Also these testbeds are normally fixed and cannot be moved anywhere else, there is no portable and cheap testbed for directional mesh networks now. In order to overcome these common drawbacks, our DWN is built based on cheap and portable routers and generic open-source Linux platform for software development. And we have do experiments on the MAC protocols, routing protocols and application utilities in this DWN testbed.

In this chapter, we designed a wireless mesh network testbed for multimedia communication. Our testbed is based on OpenWrt, a Linux distribution for embedded system. Also this paper describes the detailed implementation of the WMN test bed with directional antenna. We conduct experiments on this testbed for multimedia video transmission and apply Qos mechanisms in the wireless mesh network. Furthermore, we applied two different routing protocols OLSR and Batman-adv for mesh network in OpenWrt. Also we compare the network performance of these two routing protocols for mesh network in our testbed. Also this paper studies the influence of directional antennas in video communication in our testbed. Finally, the paper discusses how to modify the MAC layer protocols in OpenWrt and adjusts some parameters in MAC layer to study the MAC protocol for long range communication.

3.1.2 Problem Statement of DMN Testbed

In order to design a wireless testbed, we need to consider not only the hardware devices but also system platforms, software protocols and application packages. Thus there come some major problems to solve for the testbed. First, a low-cost and generic development platform is a critical element in this design. Nowadays, simply laptops are often used as a development platform to form a wireless network. Using the default network settings in Laptops, we can install some application utilities or packages such as Motion and Wireshark to do research on the
wireless network. However, based on this simple platform, further study on network stacks is impractical because the protocols are just embedded in the kernel modules of the operating system, which is basically inaccessible and unchangeable for the users in laptops. Also, there is a wireless board named USRP which is pretty useful and popular for the research on wireless network. This board allows the users to develop their own protocols, however there are also some drawbacks. Firstly, this board has no operating system or protocol stacks embedded within it, thus we have to build a wireless network from the beginning, and develop our own protocols for every network layer from physical layer to application layer, this takes a lot of efforts and time, which is inadvisable for our research. Also, this USRP board is very expensive, which costs about 3000 dollars. Therefore USRP is also not a good choice for our testbed. In summary, the existing development platforms for wireless network are either too expensive or not convenient for advanced protocols development especially in MAC layer and Routing layer.

Moreover, hardware deployment and compatibility is always an issue for wireless network development. An appropriate router for our wireless testbed is really difficult to select. We need a router which can act as a mesh router in our DMN testbed, which means that the system of this router is compatible with mesh protocols such as routing protocols and we can also install and modify the protocol stacks in its network card. Even if the selected router can support new mesh protocols, the initialization of the system may also become a problem. In the process of initialization, we have to remove or erase the old kernel modules, software packages, protocols even the whole system, and then the new modules or packages are installed in this router to fulfill our requirements in the testbed. This means for our testbed, we need to apply a open-source embedded system for our selected router. Also this system should have basic network stacks and drivers for our Atheros chipset. After successfully installing the embedded system, based on the existing network stacks within, we can study and develop our own specific protocols to conduct research on the network performance for directional antennas. However, there may be some connection issues between the routers and laptop in order to install the system. A typical connection issue in the wireless router is that DHCP server in the laptop cannot resolve the MAC
address of the router and assign the proper IP address for the router. Thus this problem makes the installation of the open-source platform on the router difficult and even impossible. Therefore, even we have selected out the router and system platform for our test bed, there may be still some connection or deployment problem for the hardware devices.

Sometimes, because of the features of the hardware devices, we may also need to manage the storage system of our routers. In order to install the embedded system on the selected router, cross-compilation is normally employed, which means the system and packages are compiled first to generate the required image files in Linux system on the laptop. So we need to use the tools such as DHCP and TFTP to connect our laptop with the routers which require the images files to boot and install the embedded systems. Thus there comes a challenge for our testbed, because the router we select has very limited storage space for kernel files. Probably the image files are too large for the router to keep. Therefore, even we have setting the DHCP and TFTP server correctly, the images files cannot be accepted completely be our routers because of limited storage space. And it’s really a challenge to deal with the storage problem for hardware devices like a wireless router which is not intended for drivers and modules development.

Another challenge difficult to resolve is that how to modify the protocols and drivers in a embedded system. In order to conduct research on the network protocols such as mesh routing protocols and MAC protocols, normally a Linux embedded system is employed on the wireless router which is open-source and support network utilities and tools. Because the properties of the embedded platform, cross-compilation needs to be used to modify the kernel modules and protocols in this system. Although some application packages such as gedit and luci can be downloaded and installed from the Internet directly, however the network protocols are embedded in the Internet Stacks and drivers as part of the kernel modules in this Linux system. Thus in order to successfully develop the network protocols, the system architecture and drivers functionalities should be carefully investigated. Because of the high complexity of wireless drives, there are some many source code files to deal with in the Linux system. So it can take a lot of time and efforts to conduce research on the system architecture and the source codes of the wireless drivers
in this system before we understand the mechanisms and functions of the drivers which are associated with the network layers and protocol stacks. In summary, there is potentially a big challenge in dealing with the system architecture and the network drivers in the source code package of our selected embedded system.

The testbed also needs to install a display tool to show up the performance of the network. This display tool is important, because we have to evaluate the network after we have designed some new protocols in this testbed for our research on wireless mesh network. Therefore, a display system should be designed and implemented in our testbed. Finally, quality of service (Qos) is also another issue under consideration. In our testbed, we should also introduce a mechanism for Qos applications especially for networks with directional antennas. In summary, we have to resolve these issues for the implementation of our wireless testbed.

3.1.3 Our Contribution

In this research, we analyze and select out hardware and software platform for this testbed. Our work mainly focuses on the following points. First, wireless routers are applied to constitute a local area network based on our given topology. In order to form a mesh network, these routers should be transformed into mesh routers in our wireless mesh network. Then the mesh clients such as PC and Laptop can have access to the mesh network within the range of a specific mesh router. Thus, in our testbed, multiple integrated routers form the wireless mesh network in a specific topology. And every routers act as a mesh router in the mesh network. So in order to fulfill the requirements of our DMN testbed, we choose Mikrotik routerboard as our basic element which can interconnect with the directional antenna. Although this routerboard has their own system, some open-source embedded system such as OpenWrt and dd-Wrt can also be installed and employed based on the hardware configuration of this routerboard. Also the routerboard is pretty cheap, each costs about 200 dollars and it supports two radios 2.4GHz and 5GHz.

Also with an embedded Linux system named OpenWRT installed in our integrated router, a lot of applications can be implemented in our directional WMN testbed such as real-time multimedia transmission and Qos mechanisms. Also, we can equip the mesh routers with
directional antennas. Within the Linux platform of our mesh routers, we can customize some applications and utilities such as luci and telnet. Moreover, in order to investigate the network protocols in the mesh network, we can design and implement our own protocols for the mesh routers. For example, in our Linux development environment, we can apply and install some ad hoc routing demons such as OLSR and Batman-adv. For further research on the MAC layer, we can also have access to the ath9k device driver and mac80211 package in our platform to develop our required MAC protocols in our mesh network. We also introduce some application packages such as Motion to our system to detect and evaluate the performance of our wireless network. Therefore, we design and implement a wireless testbed which can be generally used in many scenarios for the research on protocols in wireless network.

This paper is organized as follows. First in Section II, the related work about mesh network is presented. Then in Section III, we introduce the hardware and software tools used in our testbed. In section IV, the methodology and procedures are put forward for implementing our testbed. Then in section V, we present some results of our experiments and some evaluation of performance on our testbed. Finally, there comes the conclusion of our paper in Section VI.

3.2 Related Works

There are a lot of research work on wireless mesh network testbed. We know there are 8 principles that multi-beam directional antennas have to follow for simultaneous transmissions in the wireless networks based on [23]. Also [3, 15] discuss that wireless mesh network integrated with multi-beam directional antenna can satisfy the requirements of QoS. A novel MAC protocol for directional antenna is discussed in [6, 24]. A wireless token-passing protocol for QoS is discussed in [14]. [33, 43] studies the influence of the distance on the behavior of IEEE 802.11 DCF. But this paper only makes minor changes to the MAC protocol, which makes it improper for MAC protocols development as a testbed. Also this paper only conduct research on 802.11 DCF, there’s no proof that their model can be used to help modify and design new MAC protocols. [45] describes FreeMAC which is a reconfigurable MAC protocol development framework for the design and implementation of multi-channel MAC protocols. However this
testbed didn’t consider using the directional antennas. And they are designed to develop the MAC protocols, thus there is no environment for routing protocol or application layer development. And also the platform is mainly for multi-channel MAC protocols. Therefor this FreeMAC is not a generic testbed for network protocols design and development. [26, 39] proposes a softMAC which can act as a flexible wireless research platform. In this platform a software system is developed to allow researchers to use proper network cards to conduct experiments on the MAC protocols. Also their testbed didn’t provide any platform for the development of other layer protocols. [46] introduces a testbed to develop and test multi-channel MAC protocols in a mesh/adhoc scenario. However, this testbed turns out to be pretty hard to implement and the cost can be very high. On the other hand, our testbed is pretty easy to implement. The software platform is also open-source, thus it’s convenient to modify the protocols on MAC layer and Routing layer and set up the applications based on this platform. And the routers used are pretty cheap, just 200 dollars for each one. Therefore based on these inexpensive hardware and software, we can design and build an open-source and easily-operated wireless testbed which can be generally used for research on network protocols. And most importantly, because of our cheap routerboards, our testbed has advantages on the investigation and development of large scale networks. For example, in order to test the performance of the directional routing protocol, we need to deploy as many nodes as possible to form the required topology for the mesh network. Therefore, scalability becomes the best advantage for our DMN testbed especially when we use it to study and test the MAC and routing protocols based on directional antennas.

3.3 System Architecture

For system design of our DMN testbed, we emphasize on its reconfigurability and scalability because we want to conduct research on network protocols and multimedia communication based on directional antennas on this testbed. Moreover, portability is also another requirement for the future application of wireless mobile network. So besides the selection of hardware devices and software systems, a concrete and convenient system architecture should be considered first. Because of the small-scale of our routerboard, our system
can be easily enlarged which means the extra mesh nodes can be directly added into the existing mesh network based on the open-source Linux platform embedded into the routerboard. We can configure the routerboard as mesh routers easily in the embedded system by setting and modifying the configuration files in the platform. Also for our DMN testbed, directional antennas are connected to the wireless board of the router. Another point in consideration is that a surveillance system is required to test and evaluate the performance of this testbed. Also Qos mechanism is applied in this DMN testbed for multimedia communication.

In order to build a flexible and robust system for our testbed, we should select out appropriate hardware and operating platform carefully. As stated before, the hardware should be cheap and support the platform in which network protocols are accessible to study and modify. Also this hardware device is required to work together with the directional antenna. This means the selected router should have interfaces for different antennas such as omnidirectional antennas and directional antennas. Also this testbed also has a lot of requirements for the operating platform we are working on. At first, this platform should be open-source, which means we can investigate and modify the source codes for this system including network protocols. Also, there should be enough existing application packages for this platform to use especially in network protocols. Thus there comes the concept of our WMN router shown in Fig.3.1.

In our work, for the hardware implementation, first as the basis of our testbed, we choose MikroTik Routerboard 493G as our main embedded platform. The most advantage is that it has nine Gigabit ethernet ports, three miniPCI slots, and two switch chips, so the ethernet ports can be connected together in two switch groups for wire speed throughput. Also the RB493G has a USB 2.0 port and a microSD card slot for adding more storage or a 3G USB modem for backup connectivity. The heart of this device is a high performance Atheros AR7161 CPU and it features 256MB of built in RAM. Therefore R52Hn Daughterboard which is also from MikroTik naturally becomes our preferred choice for our wireless chip because these two boards match pretty well. In our testbed, another highlight is directional antenna. After thorough search and analysis, we choose HG5158PEV as our lightweight directional antenna in the directional WMN testbed. This
HG5158PEV model characterize a directional patch antenna with good directivity operating in the frequency band from 5.1GHz to 5.8GHz. The wideband 5 GHz design of this antenna eliminates the need to purchase multiple antennas for every frequency channel. This simplifies installations since the same antenna can be used for a wide array of in-building wireless applications where wide converge is desired. This antenna model has pretty good directionality which is suitable for our directional WMN. The horizontal and vertical beam width are both 12.5° while the gain can reach as high as 18dBi. So in our WMN testbed, every mesh router works with such a directional antenna to implement directional communication.

For the software design of our embedded platform, a embedded operating system named OpenWrt is generally preferred. OpenWrt is described as a Linux distribution for embedded devices such as routers. Rather than providing a static, feature-rich router firmware, OpenWrt tends to offer a full Linux system, namely, a accessible file-system composed of packages for various functions. This mechanism allows the user to customize the system by using the suitable packages to implement required applications without having to develop a new firmware for it. Thus with OpenWrt a variety of software can be applied to run on the routers which may have never been taken into consideration before. The OpenWrt build environment is made up of scripts, Makefiles, patches and packages. By cross-compilation, the required firmware images are generated in the bin directory (bin/) and the user-space tools are put into the building_dir directory (building_dir/).

### 3.4 Hardware Preparation and Deployment

For OpenWRT, in order to install this system in our Routerboard, we should use cross-compilation and set up it correctly in our computer first to get these image files. And then we can transfer the image files to the specific mesh routers to install them as the operating system and kernel modules. In order to build and install the images on our Routerboard 493G, there are mainly six steps including NAND repartition, building the OpenWrt Ramdisk image, building the OpenWrt target image, configuration of network, DHCP and TFTP servers, booting the Ramdisk image and flashing the target image. For Routerboard 493G, the space of the NAND is divided
Figure 3.1: WMN Router
into 6 segments including "kernel" blocks and "rootfs" blocks. We found that in default the space of "kernel" blocks is about 4MB, however the space of "rootfs" blocks reaches about 127MB. Obviously the space of "kernel" is not large enough for our elf image. Thus as the first step, repartition is required to cut down the space of "rootfs" partition for extending the partition of "kernel". Based on the structure and mechanism of OpenWrt, the MTD driver holds functions and files to read and write the flash memory, and the location is "target /linux/ar71xx/files/drivers/mtd/nand/". In this directory, we are interested in rb4xx_nand.c based on the hardware RB 493G. Thus we patch this file to increase the "kernel" partition to 8MB as shown in Fig.3.2 and Fig.3.3. For the next two steps, they are basically the same, except for generating different images such as elf image and tar.gz image shown in Fig.3.4 and Fig.3.5 according to our configuration in the OpenWrt GUI. The fourth step is pretty important for OpenWrt installation. Because DHCP server is used to assign the router an IP address. And TFTP server helps to automatically send the images from a laptop with static address to our mesh router.

In the next step, we boot the ramdisk image shown in Fig.3.6 which can provide a temporary version of OpenWrt on our routerboard. So in this temporary system, we erase the content of the "kernel" and "rootfs" and create new yaffs2 partitions. In order to flash the target image, we transfer the target images into the routerboard using security file copy (scp). After having the target image in our system, flashing the image becomes pretty simple, just taking the target image as the new "kernel" and extracting the tar.gz image as the new "rootfs". So if we reboot the routerboard from the NAND, we will install the OpenWrt on the routerboard successfully.

```
root@OpenWrt:-# cat /proc/mtd
dev: size erasest size name
    mtd0: 00000000 00000000 "rootboot"
    mtd1: 00000000 00000000 "hard_config"
    mtd2: 00000000 00000000 "bios"
    mtd3: 00000000 00000000 "soft_config"
    mtd4: 00000000 00000000 "booter"
    mtd5: 00000000 00000000 "kernel"
    mtd6: 00000000 00000000 "rootfs"
```

Figure 3.2: Comparison of NAND partition
Figure 3.3: Patch for NAND repartition

Figure 3.4: Ramdisk Image
Figure 3.5: Target Image

Figure 3.6: Temporary System
After successfully building and installing OpenWrt on Routerboard 493G, we finally obtain a basic WMN router for the testbed which consists of a number of mesh routers in a specific topology. Thus the concept of WMN router in our testbed is related to both hardware devices and software applications. As for hardware, the router can work together with both directional antenna and omni-directional antenna. Also the wireless board can provide two frequency bands 2.4GHz and 5GHz. In terms of software, with OpenWrt as an embedded Linux system, we can customize the applications, utilities, even kernel modules such as routing protocol and MAC protocol by modifying the corresponding wireless drivers.

3.5 Directional Routing

3.5.1 Routing Protocols

In the routing layer, there are mainly two protocols which are commonly used for wireless mesh network including OLSR and BATMAN-adv. OLSR is the abbreviation for Optimized Link State Routing which is a proactive link-state routing protocol for wireless ad hoc networks. The most significant forwarding policy and mechanism in OLSR is multi-point relays which means only a subset of neighbors which are chosen for link state updates and are entitled for the responsibility of forwarding packets. By using the MPR nodes, the number of links for periodic updates can be dramatically decreased especially in highly dense networks. And also cooperated with directional antennas, the long range communication can be achieved by the

This protocol selects out the shortest route based on the network topology. Bascially it uses two types of control messages hello and TC. Also HNA messages are used to allow hosts to announce themselves as gateways. There is a OLSR Daemon based on OpenWRT which can be easily installed and set up by "opkg install" command. Before installing OLSR in our platform, we can also introduce another utility named LuCI. LuCI is an application that supports a maintainable interface for embedded devices. And this tool can be easily downloaded and installed in OpenWRT. After installing LuCI using "opkg instll" command, we can see the
overview of the network clearly in Fig.3.7. And before installing OLSR Daemon, we also need to configure the Firewall setting for forwarding rules. Because the default setting for forwarding is reject. So in order to activate the forwarding function for our routerboards. First we need to a new zone section to cover our WIFI interface in Fig.3.8. After setting up the interface, a forward section is required to allow different interfaces to connect to each other in Fig.3.9. Thus after we download and install the OLSR Daemon in our routerboard, we can use a computer to connect to the routerboard with a network cable. By using the IP address of LAN in our routerboard, we can open the homepage of Luci for our network. Fig.3.7 shows the homepage of Luci for a wireless network with OLSR. Fig.3.10 shows the connection of neighbors in OLSR mesh network. From Fig.3.11, we can figure out the topology for active OLSR nodes in OLSR Mesh. We can see that these active nodes and their last hop nodes have good communication status. Fig.3.12 shows the active host net announcements. These HNA messages allow hosts to announce themselves as gateways. Also we can see the firewall setting in Fig.3.14. And Fig.3.13 shows us the visualization of 3 points communication.

<table>
<thead>
<tr>
<th>OLSR Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interfaces</td>
</tr>
<tr>
<td>Neighbors</td>
</tr>
<tr>
<td>192.168.2.10</td>
</tr>
<tr>
<td>192.168.2.5</td>
</tr>
<tr>
<td>Nodes</td>
</tr>
<tr>
<td>HNA</td>
</tr>
<tr>
<td>Links total</td>
</tr>
<tr>
<td>Links per node (average)</td>
</tr>
</tbody>
</table>

Figure 3.7: OLSR overview in Luci

For our testbed, first we download and install OLSR protocol, considering OLSR is a proactive protocol, then we implement and set up another routing protocol named
Figure 3.8: Interface configuration

Figure 3.9: Forward section

Figure 3.10: Neighbors in OLSR Mesh
Figure 3.11: Topology in OLSR Mesh

Figure 3.12: HNA in OLSR Mesh
Figure 3.13: Viz in OLSR Mesh

Figure 3.14: Firewall in OLSR Mesh
BATMAN-ADV in order to test and compare the performance of different routing protocols. There are default packages for OLSR and BATMAN-ADV on the official website of OpenWrt. Thus if we want to modify the routing protocol, we can modify the source code and then recompile these files with C language in the packages according to our specific requirements such as changing the control messages or using multi-cast instead of broadcast with directional antenna. As we all know, most of the classical routing protocols are typically not well suitable for wireless ad-hoc networks, mainly because they are normally unstructured, sometimes dynamically change their topology, and are based on an inherently unreliable medium. OLSR, the currently most employed protocol for such scenarios, has introduced some major changes to deal with the challenges raised by wireless mesh networks. However, due to the constant growth of existing mesh networks and because of the inherent requirement of a link-state algorithm to recalculate the whole topology-graph, the limits of this algorithm have become a challenge. Recalculating the whole topology graph once in an actual mesh network may take up to several seconds in an embedded system.

In order to deal with the challenges in mesh networks, another routing protocol named BATMAN-adv is proposed compared with OLSR. Compared with OLSR, Batman-adv has some very remarkable advantages. The approach of BATMAN-adv is to divide the information of the optimum end-to-end routes in the mesh to all functioning nodes. In this protocol, every node maintains only the information about the best next hop node to the other nodes in its routing table. Therefore there is no need for a global knowledge for the change of local topology. This protocol is especially beneficial when there are some unreliable links in the networks. Therefore B.A.T.M.A.N is explained in this way. At first every node broadcast the control packets named OGMs to remind the neighbors of its current information. The neighbors keep forwarding the OGMs to inform their own neighbors about the existence of the source node of these messages and so on and so forth. Therefore the network is flooded with OGMs.

B.A.T.M.A.N. advanced (BATMAN-adv) is an actual implementation for the B.A.T.M.A.N. protocol using the linux kernel module which functions on layer 2. Different from
most wireless routing protocols such as OLSR, BATMAN-adv is built entirely on Layer 2 in the OSI model, which means the routing information is transmitted in raw frames. This protocol combines and forwards the data to the destination, which makes a virtual switch including the nodes along this route. In this way by linking locally, the nodes do not have to know the global topology which make it stable and invulnerable when there rapid changes in the network. Because of functioning on layer 2, the nodes may even have no IP address. Additionally BATMAN-adv supports the "bridge" function automatically, which means the interfaces within a routerboard can communicate to each other automatically. Moreover because of its mechanism, BATMAN-adv can not be affected by network changes which makes it a robust and steady protocol for mesh network. Also for BATMAN-adv, it’s easy to encapsulate the mesh nodes with non-mesh clients, which makes it suitable for mesh network of high scalability and extendability.

In order to configure and debug the BATMAN-adv kernel module, the batctl tool was developed. It offers a convenient interface to all the module’s settings as well as status information. It also contains a layer 2 version of ping, traceroute and tcpdump, since the virtual network switch is completely transparent for all protocols above layer 2. Fig.3.15 shows the configuration for interface. Fig.3.16 shows the network configuration for our mesh. Different from OLSR, we still have another configuration file called batman in which the batman interface bat0 should be set up correctly in Fig.3.17.

Figure 3.15: Device Configuration in batman-adv Mesh
Figure 3.16: Network Configuration in batman-adv Mesh

```plaintext
config interface 'lan'
  option ifname 'eth0 bat0'
  option type 'bridge'
  option proto 'static'
  option ipaddr '192.168.1.13'
  option netmask '255.255.255.0'

config interface 'mesh'
  option mtu '1532'
  option proto 'batadv'
  option mesh 'bat0'
```

Figure 3.17: BATMAN Configuration in batman-adv Mesh

```plaintext
config mesh 'bat0'
  option 'aggregated_ogms'
  option 'ap_isolation'
  option 'bonding'
  option 'fragmentation'
  option 'gw_bandwidth'
  option 'gw_mode'
  option 'gw_sel_class'
  option 'log_level'
  option 'orig_interval'
  option 'vis_mode'
  option 'bridge_loop_avoidance'
  option 'distributed_arp_table'
  option 'network_coding'
  option 'hop_penalty'
```
3.5.2 Directional Antenna and QoS Application

In order to apply QoS functions for the multimedia communication on our testbed, the best way is to apply a QoS application tool to our embedded system. And the QoS tool in the system may automatically apply the QoS policies in the communication on our mesh routers. For QoS tools such as qos-scripts, all we need to do is to design an appropriate policy in the configuration scripts for our required communication scenario. This seems very easy and efficient for develop QoS functionality in a wireless network based on OpenWrt. However because OpenWrt is an open-source platform, there comes a normal and annoying problem, software compatibility. We tried to install the qos-scripts package on our platform for many times, but it didn’t work well in our system and often a system crash happened. After doing a lot of experiments on Qos in our testbed, we found that qos-scripts can be not be compatible with our version of OpenWrt, attitude adjustment. This is mainly because qos-scripts is a quite old application package for OpenWrt. And after these years, there are many new version of OpenWrt system. However the application package of qos-scripts hasn’t been updated timely. This means we cannot use the existing Qos application packages directly and we should design a new Qos application mechanism based on the hardware and software functions in our testbed. Based on the hardware properties, we find that there are two radio frequencies 2.4 GHz and 5 GHz in our routerboard. Thus naturally we consider to use the two different radios for different priorities in our multimedia communication. Thus in our testbed, for video streaming, the nodes or routers normally use the frequency bands around 2.4 GHz, and for other communication such as audio, The bands of 5 GHz are often employed for the routers. And in the configuration files, these two different radios are associated with different link rate, thus transmission rate in our testbed. Also, in our testbed, for Qos application, the different types of communication between two nodes are also assigned different priorities. Often for different port in this node, we allocate different link rate for different type of communication. As shown in Fig.3.18, for node A and node B, there are two kinds of communication between them. Thus two port are used for the connection between them. Port 8080 is assigned a higher priority so it’s used in the communication requiring higher speed such
as video streaming, however Port 8081 is assigned with a lower priority, so communication such as audio and file transfer is attached to this port. Therefore, with the hardware properties of our routerboard, we proposed a simple and efficient Qos mechanism in our testbed based on different radio frequencies and application ports.

![Diagram of video streaming](image)

**Figure 3.18: Video Streaming Based on Different Transmission Priorities**

### 3.6 Mac Layer Experiment

The research on MAC protocols is a major task for our testbed. Because the MAC layer is partly embedded within the kernel space of the operating system. For our mesh router, an embedded Linux system called OpenWRT is applied in every node. And it’s really hard to modify and change the MAC protocol directly in this system unless you change the kernel modules and packages such as mac80211. However, because embedded systems need to employ cross-compilation method to be installed in the local computers and then transferred onto the mesh routers, this makes the access to MAC protocols more difficult, complicated and even inaccessible. Thus in order to modify or develop MAC protocols in OpenWrt, we need to conduct research on the system architecture and the kernel drivers in OpenWrt first. After investigating the system architecture in Fig.3.19, we find that there are mainly two drivers which should be taken care of, that is mac80211 and ath9k which have access to the hardware drivers and atheros chipset. We find that we can use wireless extension(wext) to connect to mac80211. But wext can
just use the existing mac80211 driver such as configuration of network. In the configuration files, we can specify that we use mac80211 protocol. This is basically wireless extensions and tools in an embedded system. Therefore in order to conduct research on the MAC protocols, we have to investigate the source code package of the mac80211 driver and ath9k driver first. So we propose a method to modify and develop MAC protocols. In this method, we first study the source code files of the drivers, then we can make modifications in the source code package according to our network requirements. And then based on cross compilation, we compile the source code package "compact – wireless". Finally we transfer the generated ipk files into the system of routerboard and install this network network module and package in our OpenWrt.

And mac80211 is a major framework for the development of the drivers for SoftMAC devices. The architecture of mac80211 is shown in Fig.3.19. We can see that mac80211 calls the cfg80211 functions for SoftMAC configurations through cfg80211 for access to the lower level driver and systems and wireless configuration APIs. cfg80211 deals with the configuration and connects to the tools and devices in kernel space by nl80211 and wext. And these setup tools and modes are built within the embedded system as wexts. These tools can only make some minor changes on the MAC layer such as debugging and testing the MAC protocol in use. There are nothing they can do about modifying the parameters or functions in a MAC protocol. In order to modify "mac80211", we need to focus on ath9k driver which is located in compat-wireless in OpenWrt. And there are an amount of documentation on ath9k source available online, here we just introduce an overview about the source code packages which need to be dealt with in order to have an overall understanding of the 802.11 protocol in ath9k. There are some important structures and registers that are applied frequently for the development on the ath9k source code such as ieee80211_hw, ath9k_hw and ath9k_txq.ieee80211_hw is in charge of the configuration for hardware such as number of hardware transmit queues and rate control algorithm. Normally this structure is transferred from mac80211 to the hardware driver while calling ath9k_tx() function in mac80211-ops.c. ath9k_hw means the Atheros hardware which shows the state of the device. It includes pointers to structures like ieee80211_hw, ieee80211_channel, ath9k_desc,
ath9k_txq, ath9k_txq_info, and function pointers of transmit and receive descriptors. **ath9k_txq** represents the queues in hardware with the fields such as queue number, number of queued buffers, max allowed number of queued buffers, etc. All the queues in struct ath9k_hw are initialized in ath9k_init() function in base.c. This structure pointer is also passed to the function ath9k_tx_queue() in base.c from ath9k_tx() in mac80211-ops.c. Using this structure, we can check if the queue is already at its maximum size in ath9k_tx_queue() function.

First we modify the specific source code in compat-wireless for ath9k. To get the required binary file, we need to compile the modified wireless driver which belongs to the package 'mac80211' in OpenWrt. After obtaining the binary file '*.ipk', the modified wireless driver can be loaded to the routerboard. Mainly we make two modifications to the original ath9k driver, we disable the 'backoff' and 'ack' function in MAC layer. For 'backoff', this function is mainly implemented in the source code files 'mac.c' and 'hw.c', and for 'ACK', this feature is related to the files 'tx.c' and 'hw.c'. So we modify the specific functions in the above files and implement the revised drivers in our routerboards.

Also, we can modify the registers in the source code to explore the MAC protocol. For example, we can set "AR_DIAG_IGNORE_VIRT_CS" to disable virtual carrier sensing and set "AR_DIAG_FORCE_RX_CLEAR" to disable physical carrier sensing which is located in ath9k/hw.c. Also register "AR_D_GBL_IFS_SIFS" represents SIFS timeout, and register "AR_D_GBL_IFS_SLOT" control the SLOT timeout. And register "AR_D_LCL_IFS" is used to manage minimum and maximum contention window and AIFS values. Moreover, we can set "IEEE80211_TX_CTL_NO_ACK" to disable ACK which is located in mac80211/tx.c.

### 3.7 Results and Evaluation

In this section, we do simulations to evaluate our routing protocols and MAC protocols in a mesh network with directional antennas. Our first task is to set up a basic mesh network. Thus Install a Linux distribution "OpenWrt" into MikroTik Routerboard, configure the network interfaces, wireless radios and hardware devices to build a local area mesh network. And set up Motion for video surveillance within the network to test real-time communication. After the
Figure 3.19: MAC80211 Architecture
configuration of mesh network, we start to implement routing protocols that is OLSR and Batman-adv. So we install luci and olsrd in OpenWrt. Then configure olsrd and firewall to implement OLSR as the routing protocol within the network. Also we implement multi-points communication with OLSR. On the other hand, we also develop another new routing protocol Batman-adv. And set up VLC media player to stream the video for communication within the mesh network.

Thus in order to test the routing protocols, we conduct an experiment on 3 points communication. Then using Motion, we can get the video captured for OLSR mesh in Fig.3.20. In this graph, we can find that the camera on the compute in the middle is flashing and the cameras on the other two computers are closed. So only the camera on the computer in the middle can capture the picture which shows a person stands in front of this computer. The red arrow points to the routerboard acting as a node in our mesh network. The blue lights on these routers show that the router is booted and working well. We configure the hardware radio and wireless interface of the three routerboards to form a wireless mesh network together. On the two other computers, we can use the network explore to connect to the computer in the middle. Then after connecting to the computer the middle, we find that the captured video in the centric computer is transferred to the other two computers. And the video is quite clear and steady which proves that the mesh network works well and steadily based on our testbed. In summary, our testbed is capable and suitable for video communication based on the routerboard and OpenWrt system we selected out.

In order to test the routing protocol with directional antenna, we deploy a network for 4 points communication. And we choose to use batman-adv to conduct the experiment in Fig.3.21. Then for testing the directionality, we start to rotate the directional antenna, basically we set node B’s antenna against A and C’ antenna toward A. The we use batctl to check the route D to A, we can find that batman-adv chose the route $D \rightarrow C \rightarrow A$ in Fig.3.22. Also we switch C’s antenna against A and B’s towards A, then we find that batman-adv took the route $D \rightarrow B \rightarrow A$ in Fig.3.23. As show in Fig.3.24, In Scenario 1, the left video is transmitted through the route $D \rightarrow C \rightarrow A$, the right one is transmitted from C to A. The right video is clearer than the left one.
Figure 3.20: Video captured for OLSR mesh
because D needs C to relay data to A. Thus C can be very busy and may be congested. And the communication between D and A will be compromised because of network competition. C may be accept the connection with D when the channel between C and A is very busy. Therefore the video between D and A is not that clear as the video transmitted from C to A. In Scenario 2 shown in Fig.3.25, the qualities of videos are almost the same because D sends video to A through B and will not compete with C. This means that the route from D to A has changed because of our directional antenna. From this experiment, we can know that with the directionality change of our directional antenna, the route path will also change accordingly. So this experiment proves that in our testbed, the directional antenna works well and effectively based on our routerboard and routing protocols installed in OpenWrt which also substantiate the reconfigurability and scalability of our DMN testbed.

![Figure 3.21: 4 Nodes Communication (Batman-adv)](image)

Figure 3.21: 4 Nodes Communication (Batman-adv)

We also conduct experiments on the Qos function introduced in our testbed. As show in Fig.3.27, the two videos are using different ports to transmission real video streaming between two
Figure 3.22: Scenario 1 for Batman-adv

Figure 3.23: Scenario 2 for Batman-adv

Figure 3.24: Scenario 1 for different relay route
Figure 3.25: Scenario 2 for different relay route

Figure 3.26: Transmission based on different MAC protocols
nodes. We can see that the left video is using port 8081, and the right video is doing on port 8080. Obviously, the video on port 8080 has higher quality than the video on port 8081. And also we find that the video streaming on port 8080 is very smooth. we can see that the timer in the right graph shows 08:56 which means the video streaming has been kept for about 9 minutes, on the other hand, the left graph shows that it’s 01:00 which means the streaming video has just been processed for about 1 minutes. The reason why the time shown in the two graphs is different is that because of different priorities, the transmission processes on the two ports are assigned with different link rate. Because of the limited frequency band and link width in our routerboard, so there may be competition between these two transmission processes. Naturally, because transmission with higher priority can obtain larger link width, thus achieve higher transmission rate. In our experiment, obviously, port 8080 in the right priority has higher priority and thus is assigned with higher link rate. Therefore, based on the hardware properties and software functionalities in our testbed, the experiment proves that our Qos mechanism works pretty well in the communication of video stream. And real time communication with different priorities can be allocated with different frequency bandwidth and achieve different transmission rate in multimedia communication.

Figure 3.27: Video performance for different priorities

As shown in Fig.3.26, Node A and Node B concurrently transmit the same video based on different MAC protocols competing for medium access in the same channel(5.745GHz). Node C
receives the videos from A and B to display the status of communication under medium contention. Without ACK at the transmitting end, the transmitter will keep sending data, which may lead to the congestion of the channel. So the rate control algorithm "minstrel" in ath9k driver will adjust the MCS rate to improve the communication.

Figure 3.28: Video for different MAC

In Fig.3.28, the left video is using the modified MAC and the right video is generated with the default one. We can see that video for Modified MAC has blurs, this is mainly because "minstrel" reduces MCS rate without acknowledge in MAC layer leading to the switching of modulation and coding in Physical layer. This experiment proves that using the ath9k drivers in OpenWrt source code, we can modify the existing mac80211 protocols in OpenWrt and also develop new MAC protocols using this embedded system. By investigating the system architecture and device drivers in OpenWrt we can have access to part of the MAC protocols and figure out a method to modify even design our required MAC protocols. Also we find that by modifying the parameters and registers in ath9k driver, we can also control part of the functionalities of the physical layer of our routerboard such as MCS which means modulation and coding scheme. In summary, we have proposed a effective and convenient method to modify and develop the MAC protocol based on OpenWrt in our testbed and our testbed can be used to
conduct research on MAC protocols.
CHAPTER 4
NOVEL OLSR ROUTING FOR LOW PROBABILITY OF DETECTION

4.1 Introduction

4.1.1 Background and Basics of Low Probability of Detection (LPD)

In the recent decades, secure or covert communication in wireless networks has become a popular topic in various areas with civilian, industrial, and military applications. Normally the specific information in the wireless networks is protected from the detection or interception by an eavesdropper with complex encryption or private and public key validation policies. However, there are many specific areas where standard cryptographic security protocols are not enough for protecting the data. Therefore, to deal with these special scenarios, LPD communication protocols which can compromise the detection of date delivery by adversaries are proposed instead. And in the mobile ad-hoc networks (MANETs), there may exist electromagnetic disturbance or attacks which interfere with the normal wireless communication systems. Therefore, Anti-Jam (NJ), Low Probability of Detection (LPD) and Low Probability of Interception (LPI) are of significant importance for these systems [21].

In order to deal with the detection and attacks against the encryption algorithms from the potential adversaries, a lot of secure communication technologies have been proposed to protect the required data for the specific users. Commonly, most of applications choose to resolve the issue in the physical layer by applying the spread spectrum techniques to suppress the strength of the information to mix the actual signal with the noise in the chance of the adversaries [9]. Thus, recently most of the works and research are focused on the physical layer which leads to the fast development of advanced and robust encryption protocols. Also based on the advance of manufacture, more accurate and configurable network devices such as directional antennas are
devloped to provide more detailed functions. In this way, now there are limited study and research on networking protocols and mechanisms such as MAC protocol and routing protocol to guarantee secure LPD communication in other network layers other than just physical layer. Moreover, with the development of cryptography techniques, in theory even the most complex encryption algorithm may be broken with some brute force methods such as side-channel analyisis from the potential adversrasy with very accurate and advanced devices. Also the transmission of the signal can be detected by an adversary with highly sensitive detection abilities.

Based on the basic limiations on LPD communication in [7], the control algorithms for power and devices can be implemented in the physicla layer. Furthermore, the functions of the LPD communication can extend from the protection for the data involved to the transmission from the soruce to the end against the detection of a targeted adversary. However, in order to design such a covert communication system with LPD, new policies in other layers such as routing layer are also indispensable besides physical layer. Therefore in this chapter, we will focus on the design of routing protocols for LPD communication with multi-beam directional antennas. Normally, directional antenna is a very useful device to be applied in MANETs because of the concentrated radiated energy and extended range in some specific directions to increased spatial reuse and increased signal to noise plus interference ratio (SINR). Also they can achieve different functions and requirements in LPD communication. However there exist some issues which can prevent the directional antennas to fulfill the requirement in LPD communication. The extended transmission range of directional antenna can lead to the high probability of detection especailly when there exists an adversary with high sensibility in the specific communication area. Therefore in order to keep the high network utilization by directional antenna, new protocols in MAC layer and routing layer should be proposed to minimize the probability of detection in a highly sensitive network with potential adversaries.

In our research, a novel scheme is proposed to optimize the proactive link state routing protocol (OLSR) in MANETs with multi-beam directional antennas (MBDA), to fulfill the fundamental requirements of reliable LPD for secure communication. We considered the impacts
of transmission power, noise and information constraints on the LPD performance in our enhanced OLSR routing. A power control algorithm is applied to control the radiation footprint of a group of mobile nodes by considering the radiation pattern, network topology, mobile node positions, and the worst-case propagation model. Also a modified neighbor discovery is proposed as an prerequisite for applying directional antennas for LPD in secure communication. In order to exploit the benefits of directional antennas, an innovative routing shortcuts mechanism is proposed to optimize the multipath routing protocol for better network performance and efficiency. Considering all the limitations and requirements, we propose a multi-beam OLSR protocol to exploit the advantage of multi-beam transmissions and select the proper multi-point relays (MPRs) to reach all nodes with a limited number of broadcasts. Moreover, an innovative method based on social networks analysis is used for the selection of MPRs for our multi-beam OLSR. Finally the simulations for our novel protocols are performed in different scenarios. The results reveal that the enhanced OLSR for LPD is suitable for mobile, large and dense networks with heavy traffic, and can satisfy critical, real-time applications.

4.1.2 Significance of Neighbor Discovery on Routing Schemes

In MANETs, neighbor discovery is a pivotal and initial step and prerequisite to construct the network topology and design networking protocols especially for the networks equipped with directional antennas. Commonly, most of neighbor discovery algorithms depend on sweeping procedures. In this way the node periodically broadcast the advertisement messages in all directions towards the maximum distance in order to reach the distant neighbors. Thus periodic transmission may be incepted and detected in a long period of time. Also the information such as speed, location and movement may be captured by the adversary. Therefore for wireless networks with directional antennas, the range of the periodic advertisement messages should be decreased [41] [54]. Therefore in LPD communication, it’s very important to keep the radiation pattern of the system at the minimum power which is required to keep the efficient communication. Considering the requirements of LPD communication, a novel neighbor discovery policy for LPD should be developed for networks with directional antennas. There are
basically two types of neighbor discovery algorithms which are random access and synchronized search based schemes. Research about the common random access based approach can be found in [49] [55].

For the random access based algorithms, on the receiving side, the node uses the omni-directional antenna for the possible RTS control message from around. Thus the actual maximum transmission range for this method can not reach as long as the range using directional antennas on both the receiving side and sending side. Another vulnerability is the random access method always tries to apply the longest transmission range for all the neighbors around which can be easily detected by an adversary.

As we all know, nowadays directional antennas are now mostly used in wireless networks with tier-2 backbone nodes in 3D environment. A typical example is the airborne networks of high mobility in our research. In this way, because of the high mobility, it becomes difficult and unstable for a node to figure out the correct transmission direction to reach and keep contact with a potential neighbor. Of course, a node will have to spend much more time on neighbor discovery in 3D space than in 2D.

As we all know, the optimal transmission probability is inversely proportional to the beam width of the directional antennas, which means it may have to implement more transmissions for the antennas with narrower beams. In order to increase the coverage area, narrower beams are required to retain higher antenna gains. Thus, in this case, these random access based algorithms require more transmissions and thus increase the probability of detection. Therefore, in order to solve the neighbor discovery problem with LPD, a modified scan-based neighbor discovery protocol is proposed for multi-beam antennas. This modified algorithm recursively and dynamically applies the scanning process including two steps: power increment and beam scanning. In the beam scanning step, with a low transmission power, each beam starts to scan using its own beam width to select out some potential nodes in every potential angle. Also in the power increment step, the specific node gradually increases the antenna gain and then jumps back to the beam scanning process. In this way two steps alternate recursively to scan for all the
potential neighbors for this transmission node in LPD communication.

4.1.3 Basic of OLSR

Recently OLSR has become a very popular proactive routing protocol for mesh networks. Basically OLSR is a global link state routing algorithm which is designed to store the routing information in the whole network. Because of less average delay, OLSR is often applied for the specific applications which require low average end-to-end delay such as airborne networks. Especially OLSR can keep the efficiency and stability when mesh networks are of high mobility and density. Another benefit is that OLSR is also suitable for the mesh networks in which the source and destination node can change very quickly.

For a typical link state routing protocol, flooding based mechanism is applied to update the state information of the links in the network. However, flooding the network is vulnerable and should not be allowed in LPD communication, because the periodic update information may be easily detected by the potential adversary. Thus, link state protocols should be optimized in order to fulfill the LPD requirements. Among the link state protocols, OLSR is often applied to wireless mesh network. The advantage of OLSR is that it introduces an efficient link state update mechanism rather than brute force flooding the whole network. In OLSR, only a subset of nodes participate in the update of link state in the network. These subsets of links or neighbors that are designated for link state updates and are assigned the responsibility of packet forwarding are called multi-point relay (MPR). Therefore, different from conventional link state protocol, only MPRs are selected to forward the link state update packets. The smaller the number of nodes selected as MPR, the higher the efficiency of protocol compared to link state routing.

Because in OLSR a source node computes the optimum path to the destination node with the help of its MPRs, thus the number of MPRs has a significant influence in the overall performance in the network. With multi-beam directional antennas, it’s even more efficient to select out the MPRs because the extended transmission range compared with omni-directional antennas. Thus, in order to further minimize the number of MPRs, we propose a new OLSR protocol with multi-beam directional antenna. In this novel OLSR, first the node selects out the
neighbors that can act as MPR using the neighbor list and beam table. Then in order to ensure LPD requirements, the beams corresponding to the MPRs are chosen to send out the link state packets sequentially which means in one slot, only one beam is used to announce the link state packets to decrease the probability of detection. In highly dense network such as UAV swarming networks, MPRs can help a lot to avoid flooding the network and thus satisfy the requirements of LPD communication. After efficiently updating the link state, we also extends the conventional OLSR into multi-path routing with the introduction of multi-beam antennas. The multipath routing algorithm takes advantage from the large and dense networks and attempts to mainly address the problems of the scalability, mobility and link instability of the network by using multi-beam antennas.

However, in LPD communication, as we mentioned above, more routes means higher probability to be detected by the potential adversary. Thus, in order to deal with the limitations of LPD, more fields or parameters such as power level should be considered into the routing table to make decision on the multi-path selection. Also a power control algorithm is applied for the multi-beam antenna to implement LPD communication. Also based on the update of the adversary information, route shortcut may also be used to exploit the advantages of multi-beam directional antennas. To deal with the occasional change in network topology and adversary information, route back and route recovery can also be implemented in this multi-path OLSR routing.

4.1.4 Social Network Analysis Methods

Basically, a MANET is a dynamic or mobile wireless network which may have no typical infrastructure which means the nodes within may have random speed and arbitrary structure.

Therefore prediction for the potential topology is of significant importance for routing reliability and stability in MANETs. Therefore, the pivotal issue comes to the basic element in the networks, that is the node, especially those are more important or predictable in speed and location than its fellow neighbors to make more accurate prediction for the future topology. Recently, an effective and non-conventional analysis method is proposed and applied to select out
the typical or characterized information by using a phenomenon called small world dynamics which is common in social networks [17].

In social networks, some centrality characteristics are introduced for the computation for the importance of the individual nodes. Thus, some special nodes can become more stable than its fellow neighbors in a network of volatile topology. By using the characteristics of the centrality metrics, these specific can be accurately selected out for the prediction of basic topology in mobile networks. Typical the importance of a node is representd using three different metrics including degree, closeness and betweenness centrality [28].For simplicity, the metrics may be roughly calculated using the empirical information, mobility model and previous estimations. Degree centrality measures how many connections each node has and has been used to connect to the networks. For examples, using degree centrality, the center node in the area of interest can be easily find out in airborne network with UVAs. Closeness centrality measures the average distance to all other nodes in the network and has been introduced to the research of interference and influence.

Also betweenness centrality is used to choose out the nodes which can be represented as the bridges between different groups of nodes. Nodes of high betweenness centrality can help a lot to form an alternative routing path between different source and destination pairs. Centrality prediction is popular for selection of realy nodes in large scale networks [16]. The pivotal point of centrality is to introduce node centrality as metrics to select out better relay nodes, and forward the packets to the nodes which are more central than the other nodes. For airborne network, the mobility pattern of the UAVs are often predictable and arranged regularly the commander nodes. Also these UAVs can be used to collect and transfer the empirical data in the real time networks. And the commander node can use these collected data to analyze the patterns and behaviors to predict the node importance based on the fundamental metrics originated from social network analysis.

Because of the characteristics of wireless mesh network, the social network analysis approaches can also be used to help optimize the routing protocol such as OLSR and AODV.
especially under the random mobility pattern and strict LPD conditions. Because of strict LPD requirements, the number of transmission should be limited especially for the broadcast or flooding messages to update the link state in proactive protocols such as OLSR. Thus, the social network analysis methods may be a good fit for OLSR protocol in mobile networks. Mainly, the social network analysis methods can be applied to help find out the potential MPR nodes especially in a network with random mobility. With the introduce of multi-beam directional antennas, the number of MPR nodes can be significantly reduced while considering mainly high degree and betweenness centrality in LPD communication.

4.1.5 Our Contributions

In this chapter, in order to guarantee the low probability of detection with potential adversary which has high sensibility of detection in MANETs, we proposed some innovative methods and algorithms to design an enhanced OLSR with multi-beam directional antennas in LPD communication. As the prerequisite of our routing protocols, a new scan-based neighbor discovery algorithm is designed with multi-beam directional antennas. This neighbor discovery method can not only avoid the flooding of the network when broadcasting messages in LPD communication, but also exploit the benefits of the multi-beam directional antennas by scanning the section within the range of each individual beam simultaneously.

Considering the fundamental limits of LPD communication, a power control algorithm is introduced in our routing protocol. Some special metrics such as decoding error probability is applied into the routing table in our OLSR to satisfy the requirement of LPD.

For the kernel concept MPR in OLSR, we introduce some social network-based analysis methods to select out the best MPR nodes. The metrics such as degree and between-ness centrality are applied to analyze the importance of nodes and choose the optimum MPRs in our mesh network.

After the selection of MPRs, we propose a multi-path OLSR protocol using multi-beam directional antennas in LPD communication. The protocol can ensure low probability of detection and acceptable network performance when there exists high sensibility detection from a potential
adversary. Also in order to optimize the enhanced OLSR, a shortcut algorithm is proposed based on the extend range of direction antennas.

The following part of this chapter is organized as follows. In section 2, related works about LPD communication and OLSR routing protocols are described. In section 3, we introduce the new neighbor discovery protocol for LPD and its advantages with multi-beam directional antennas. The link state update and social network-based analysis methods applied for MPR selection in OLSR with multi-beam directional antennas are introduced in section 4. Section 5 presents the modified multi-path OLSR in LPD communication. Evaluation and simulation are presented in Section 6. Finally, we conclude this paper and introduce some future work.

4.2 Related Work

Because of the significance of covert communication, recently a lot of research is focused on the issue of requirements of LPD communication. In [21], the authors presented the methods and protocols for power control algorithms in LPD with directional antennas. The radiation pattern of omni-directional antennas and directional antenna are discussed. They conclude that the radiation power can never be greater than the predefined maximum value to guarantee the LPD requirements. However, the corresponding networking protocols for this power control algorithm is not discussed. In [8] [7] [9], the authors studied the theoretical limitations in LPD communication. They have proposed a square root law for covert communication over AWGN channels. Based on this paper, the fundamental limits for LPD is explained and detailed algorithms for power control is discussed. In [41] [56], the performance of neighbor discovery in MANETs with directional antennas are discussed. The authors conclude that scan-based all-directional algorithm gives better performance than the algorithm which uses omni-antennas. Also if designed properly, all-directional neighbor discovery algorithms can discover neighbors in less time than those algorithms that use omni-directional antennas. However, this paper does not cover the specific neighbor discovery algorithms for directional antennas. [19] also presents a novel neighbor discovery protocol with synchronized search using directional antennas. The transmission power is compromised to minimize the unnecessary high transmission power.
exposure time. And the drawback is that it’s not effective enough for distant neighbors.

In [2], a new OLSR is introduced for UAVs equipped with directional antennas. This protocol is focused on MPR nodes and . However, these protocol are only designed for single beam directional antennas. In [30] [44], a learning-based OLSR is proposed. The kernel idea is to tune OLSR using Boltzmann learning algorithm to make it more reliable, more energy efficient and more adaptable to the rapidly changing network topology and infrastructure. However, this algorithm does not deal with OLSR with directional antennas. And the MPR selection method is not optimized and may not be effective in LPD communication. In [17] [28] [16], social network analysis for MANETs is introduced and discussed. Network centrality calculation and prediction protocol are proposed and discussed. Also some efficient prediction functions are validated for centrality calculation in practical networks in real time. However, these research are mainly focused on social networks and human behaviors prediction. The application of this network analysis method to other MANETs is not introduced. And the coordination of these centrality with network protocols is also not discussed in these works. [51] [12] introduce the research for mobility models in ad hoc networks. A wide range of mobility models are discussed such as random walk and random waypoint. However, these research only deal with the conventional issues with mobility, the LPD problem is not introduced in these models. But in our work, the limits in LPD need to be considered when applying a conventional mobility model in a complex airborne network with UAVs and potential adversary aircrafts.

4.3 Narrow Beam Scan-based Neighbor Discovery Protocol

In LPD communication, the transmission power is controlled by the mobility model and sensibility of the potential adversary. Thus, for initialization, the power level is suppressed to a threshold to search for the closest neighbor. Firstly, a low power level is set as the initial value for the scanning step which is also called beam scanning. For our neighbor discovery algorithm, a number of slots are designed for a pair of neighboring nodes in every TDMA section [56]. And all the nodes can be classified into 3 different states: transmitting, receiving and idle.

During the state of transmission, we assume that each beam which covers its own range of
neighborhood can send out the broadcast packets in one control slot in LPD communication. Based on the prediction on the potential position of the adversary, each beam in the corresponding range or section will be selected out to start the scanning in one time slot. Thus in this way, the transmission node can scan every corresponding section around for potential neighbors. In the state of receiving, a node will open all its beams to accept the incoming packets from its potential neighbors. After this receiving node receives an announcement packets from one beam, in another time slot it will send back its information as acknowledgment packet to the neighbor in this specific beam where it received the announcement packet. Also after the transmission node receives the acknowledgement packet, it will send out its own information for the receiver within one time slot. Therefore after this three-way handshake which takes up to three time slots, this pair of neighboring nodes can share its information with each other such as node ID, location, transmitting power level and adversary information in the whole network topology.

After the transmission node finishes the beam-scanning step for all its beams. Then there may be adversary information such as location, sensitivity and noise from the neighbors found in this beam-scanning step. If the neighbors contain the information of the adversary, then the transmission node can adaptively improve its own transmission based on the information of the adversary. And if the transmission node acquires no information about the adversary, the transmission node will restart the beam-scanning step using a default power increment. This protocol allows each node to perform neighbor discovery in a bounded time and gradually acquire the neighborhood information without violating the requirements of LPD communication. In this two-step iteration, this scan-based protocol enables nodes to discover all the potential links between a given pair of nodes from short distance to long distance. In addition to discovering neighbor nodes and storing the neighborhood information, this protocol can also gather the information about the adversary such as previous location, sensitivity and noise from the neighboring nodes.

In Fig.4.1 below, we present the algorithms for this scan-based neighbor discovery algorithm with multi-beam directional antennas in LPD communication. We can see that first in
Initialization step, every node will set its own status including transfer state and power level. Also based on the principles of multi-beam directional antennas, each beam in the same node should handle the data concurrently and cannot change its TX/RX mode individually. Thus based on these assumptions, the beams can be used for scanning the potential neighbors because each beam should maintain its own beam table to record the node information such as IDs within its corresponding range. Then for each node, starting from the initial low power level, the node iterates through different increasing power levels to search out the potential neighbor with a relative low power level in this section. After setting the scanning power level, each beam also resets its orientation to the initial scanning direction in its own corresponding section. Then each beam sends out the announcement message in its current orientation within the section, as shown in the right figure. After a receiving node B receives ANC message, B sends out the ACK message to node S in the receiving beam. Then when node S receives this ACK message, another confirmation message will be sent out immediately in the same beam. At the same time, node S will stores the information of the ACK message such as node ID, beam ID in the corresponding beam table. Finally when the receiving node B receives the Conf message, then node B also keeps the information of node S in the beam table. Therefore based on this scanning mechanism, the probability of detection is compromised to its minimum in an efficient way. And because of the limited number of beams in multi-beam directional antennas, the drawbacks of the conventional flooding method can be significantly eliminated. And the benefits of multi-beam directional antennas can also be exploited by scanning the potential neighbors in the corresponding section for each beam simultaneously.

4.4 Link State Update in OLSR with Multi-Beam Directional Antennas

For proactive link state routing protocols, link state announcement packets are frequently broadcast to deal with the rapid change of network topology in mobile ad hoc networks. In conventional OLSR, a very important concept named MPR is introduced to help forward the link state packets in order to reduce both the size of the control packets and the number of required links used for updating link state around the whole network. Furthermore, OLSR can reduce the
**Input:** Initial Power Level $P_{R_0}$, Maximum Power Level $P_{R_{max}}$, Beam ID, Scan Direction $D$

**Output:** Neighbor List

1. Initialization:

   for each node $u \in N$ do
   
   $n.set\_status();$

   $n.init\_Beam(\text{Beam}\_ID);$   

   end for

2. for $P_{th} < P_{R_{max}}$ do
   
   Reset Scan Direction $D$;

   for each transmitting node $T$ do

   for each beam $B$ of $T$ do

       while $D \neq Final\_Direction$ do

           if $B.find\_Neighbor()$ and $B.rx\_Ack()$ then

               $B.add\_Neighbor();$

               $B.send\_Conf();$

           end if

       end while

   end for

3. for each receiving node $R$ do

   if $R.rx\_Ack()$ then

       $R.send\_Ack();$

   end if

4. if $R.rx\_Conf()$ then

   $R.get\_Beam().add\_Neighbor();$

end if

5. $P_{th} += func(add\_Inf())$

end for

Figure 4.1: Beam-wise Neighbor Discovery Steps.
overall overhead for control packets. Also OLSR is more efficient than other classical routing
protocols especially when the networks are of high density.

The typical MPR is shown in Figure 4.2. By using MPRs, periodic updating of link state
is efficiently implemented and flooding the network can be avoided. In conventional OLSR, for
node S, neighbors are A, B, C and two-hop neighbors are E, F and G. Node C cannot reach F and
G due to the limited range of omni-directional antenna. In order to reach all the two-hop
neighbors, node A and node B will be selected as MPRs to forward the link state update packets
for node S. Therefore there are two links originated from S for link state updating. However, if
multi-beam directional antennas are equipped for these nodes. Then because of the extended
range, fewer number of nodes may be selected as MPRs. As shown in Figure 4, with directional
antennas, node S includes A, B, C, E as its neighbors. Thus if node E is selected as MPR, then all
the two-hop neighbors including F, G, D can be accessible through E. Thus by using directional
antennas in OLSR, the number of the links used for updating link state can be efficiently reduced.
In highly dense network, it’s especially efficient to use multi-beam directional antennas for
updating the link state in OLSR protocols because the number of related links can be significantly
minimized by multi-beam directional antennas.

For LPD considerations, the link state process can be scheduled sequentially which means
in one slot, the source node only sends out link state packets to one of its MPRs using the
corresponding beam of its antennas. And in another slot, the node repeats the previous step and
continue to send the link state to another MPR. Thus in this way, each time this node announces
its link state to a partial number of nodes for the whole network topology. In order to select out
the best MPR nodes, some social network analysis methods are used [28]. The centrality of a
node in a network is a measure of the structural importance of the node; typically, a central node
has a stronger capability of connecting to other network members. In our case, degree centrality
and betweenness centrality are emphasized in our mesh networks with UAVs. Based on
Freeman’s measures, Degree centrality is calculated as the number of direction connections that
involves a given node [17]. Thus a node with high degree centrality maintains a large number of
Figure 4.2: MPR with omn-directional and directional antennas.
connections with other nodes in the network. In OLSR, we know MPR nodes are required to have the exactly same characteristic to help forward the periodic link state packets. Degree centrality for a given node \( q_i \), where \( a(q_i,q_j) = 1 \) if a direct link exists between \( q_i \) and \( q_j \), can be represented by the formula below:

\[
C_D(q_i) = \sum_{j=1}^{N} a(q_i,q_j)
\]  

(4.1)

And Betweenness centrality is usually applied to compute the degree or level for control over the information passing between the other groups of nodes. A high betweenness centrality means the node will be better at the coordination between the different groups of nodes which are related to this node in the middle. In our case, MPR nodes are responsible for forwarding link state messages to other nodes and their interactions with the links can influence the efficiency of link state updating in the whole network topology. Betweenness centrality, where \( g_{jk} \) is the total number of geodesic routes which connect to \( p_j \) and \( p_k \), and \( g_{jk}(p_i) \) represents the number of those geodesic routes covering \( p_i \), can be computed by the formula below:

\[
C_B(p_i) = \sum_{j=1}^{N} \sum_{k=1}^{j-1} g_{jk}(p_i)/g_{jk}
\]  

(4.2)

Based on the computing methods about degree and betweenness centrality, we can apply the metrics into the selection of MPR in OLSR. The steps for selecting out the MPRs with multi-beam directional antennas are summarized as follows.

Based on neighbor list \( N(S) \) of node \( S \), for every neighbor \( X_1,X_2,X_3,...,X_n \) in \( N(S) \), combine the neighbor lists \( M = N(X_1)\cup N(X_2)\cup N(X_3)\cup...\cup N(X_n) \), then for any node in \( M \) but not belongs to \( N(S) \), the node is classified into \( N_2(S) \) which means the 2-hop neighbors of \( S \).

For each node \( i \) in \( N(S) \), based on the calculation of degree centrality, select out the node with degree greater than a threshold \( D_{min} \), and add this node to a set \( C_D(S) \) which means the high
degree neighbors of node S.

Also for each node j in N(S), considering all the links between node S and the 2-hop neighbors in N_2(S), using the calculation formula for betweenness centrality, select out the node with betweenness greater than a threshold B_{min}, and add this node to a set C_B(S) which represents the high betweenness neighbors of node S and its 2-hop neighbors.

Combining the degree set C_D(S) and betweenness set C_B(S), depends on the number of beams of our multi-beam directional antennas, w nodes in N(S) can be selected out as MPR to forward the link state packets in this round. The selected MPR can form the MPR set for node S. This MPR set contains the minimum number of MPRs which can connect to all the 2-hop neighbors for node S.

After selecting out the MPR set for node S, then S can start to update the link state sequentially considering the LPD requirements. In order to minimize the probability of detection by the adversary, in one time slot, node S only uses one beam to connect to one of its MPRs. And in the next slot, node S starts to send link packets to the next node in the MPR set using another beam. The node S repeatedly sends out link state packets to its MPR until all the nodes in the MPR set have been reached. In the way, node S can announce its link state to the whole network with multi-beam directional antennas while keeping a low probability of detection. In Fig.4.3, we can see that S selects out node A as MPR to forward the link state packets to 2-hop neighbors. We can see only three 1-hop neighbors are selected as MPRs. And for node A, although it also has some neighbors, only node B is chosen as the MPR node. And when updating the link state, the links between node A and other nodes in white color will be shut down, node A just needs to forward the latest link state messages to node B. Therefore, repeatedly, in this way, all the MPR nodes will be selected out and activated to forward the link state packets in the whole range of the mesh network with multi-beam directional antennas.

4.5 Multi-path OLSR in LPD

4.5.1 Limits of Reliable Communication with LPD

In our research, covert or secure communication through LPD/LPI communication is very
Figure 4.3: MPR selection with multi-beam directional antennas.
significant and indispensable for data and network security in airborne networks. Actually, nowadays the concept of secure communication has spread out in various areas ranging from military operations, social stability, industrial materials protection to privacy for individuals engaged in wireless networks. In order to implement reliable LPD communication, normally steganography is applied, however with the development of cryptography, even the most advanced encryption system can be studied and blocked.

In mesh networks, when the source node tries to connect with the destination node, the potential adversary attempts to detect or even intercept the transmission signal. When the source sends out low-power signals, the adversary will have to classify it as either noise on the channel or the signals of the source node to destination. If the adversary classifies the received signal as the transmission from the source to the destination, then the adversary can potentially capture the information such as location of the source which make LPD communication vulnerable. If the noise between the source and adversary has non-zero power, the source can communicate with destination by reducing the transmission power while tolerating a certain probability of detection. Thus, this means that the source node potentially transmits non-zero mutual information between destination with a fair low probability of detection. In Fig.4.4, we can see that there are basically two divided areas in the airborne network including safe area and detection area. For node B, in safe area, we can increase the transmission power of the directional antenna to connect to node D directly which can bypass node C and thus decrease the number of hops with directional antennas. This is because the directional transmission range is much greater than MDR which means the maximum detection range of omni-directional antennas. However, the same directional TX will be forbidden for node B to send out messages to node A if the directional power level is greater than SDS which represents signal detection sensitivity for the potential adversary in detection area. Also we can see in the safe area, the effective isotropic received Power(IRP) for receiving nodes can be greater than MDR, however in detection area IRP needs to keep below MDR.

Although by constraining the power level $P_f$, the source can send out signals to the destination with pretty low probability of detection. However there are other issues about the
decoding process on the destination side. When the transmission power is reduced to a low level, the probability of decoding error $P_e$ may reach the threshold. This means when lower power level is applied in the source side, then the signal cannot be decoded correctly in the destination side. Thus, in order to correctly decode the signal, the transmission power cannot be too low for the destination. In summary, the power control algorithms need to be considered in the OLSR protocols with multi-beam directional antennas.

![Group radiation pattern with directional antennas](image)

**Figure 4.4: Group radiation pattern with directional antennas**

The adversary identifies if the source node is sending out signal depending on the received signals in the specific channels. The probability that the adversary raises a false alarm is represented by $\alpha$ while the source node is not working. Also the probability that the adversary does not detect the transmission from the source node when it’s actually transmitting is represented by $\beta$. Thus in order to achieve low probability of detection, the sum of error probabilities needed to be minimized. For the root square law [7], there exists a threshold $\epsilon$, when $\alpha + \beta \geq 1 - \epsilon$. In this case, it seems that the source node sends out $O(\sqrt{n})$ bits of data without the detection by the adversary with $n$ uses of LPD channel.
The decoding error probability is denoted as $P_e$. Thus the power control algorithm should focus on the tradeoff between $\alpha + \beta$ for adversary and $P_e$ for the destination. Denote the probability distribution of the adversary’s observations when the source does or not transmit as $P_1$ and $P_0$, respectively. The threshold $\epsilon$ can be restrained by the total variation distance $\nu_T(P_0, P_1)$. And the source can get the lower bound based on the variation distance between $P_0$ and $P_1$ [8].

$$\alpha + \beta = 1 - \nu_T(P_0, P_1) \quad (4.3)$$

Assume the channel is AWGN for simplicity. Thus, based on the basic concepts, we can know that $P_0 = P^m_W$ and $P^m_W$ and $P^m_S$ conform to the standard Gaussian distribution. And for the source node, we can get that $P_1 = P^m_S$. Thus based on the above equations and Pinsker’s inequality which can be used to compute the total variation distance, we can compute the upper bound for the variation distance which can be represented below [7]:

$$\nu_T(P^m_W, P^m_S) \leq P_f/(2*\sigma^2_w) * \sqrt{n/2} \leq \epsilon * f(n)/\sigma^2_w \quad (4.4)$$

For the equation above, $P_f$ represents the average symbol power on the source side with the limitation $P_f \leq c f(n)/\sqrt{n}$. Thus, by controlling the average power of the source, the detection ability of the adversary can be limited.

Moreover, on the destination side, the decoding error probability $P_e$ can be represented as follows:

$$P_e \leq 2^{nR-n/2+\log_2(1+P_f/\sigma^2_b)} \quad (4.5)$$

Therefore from the equation above, we can find that in order to decrease the decoding
error probability $P_e$, the averaged power should be suppressed under a certain level. In summary, by applying the power control algorithms, we can set the most beneficial boundaries for the transmission in a wireless network potentially with an adversary with a sensitive detection ability.

### 4.5.2 Multi-path OLSR with LPD

After analyzing the power control algorithms, we can apply this algorithm into our multi-path OLSR. The basic network topology is show in Figure 4.5. There exists a routing path form S to D. But when the adversary approaches the link between node A and B. For LPD consideration, node A should determine if the original path from A to B can still work or it should search for another new path. Thus for this modified OLSR, the routing table should introduce more fields as shown in Table 4.1 below. $P_e$ represents the decoding error probability, and $\alpha + \beta$ is the adversary’s detection error probability at the same level of transmission power level for node A. For simplicity, we assume the threshold for $P_e$ is 0.012, and the threshold for $\alpha + \beta$ is 0.75. And we can find that the link from A to B exceeds the threshold, so this link cannot work any more. Considering the detection error probability $\alpha + \beta$, node F can be chosen as the next hop for node A. Thus a new routing path can be selected out from S to D, that is S-A-F-G-C-D. This new route detours around the adversary to ensure low probability of detection at the cost of longer path and more hops. In Table 1, we can find out the required parameters that should be involed in the modified OLSR protocol in LPD communication. And here, we assume the mobility model for the potential adversary fits the Gaussian-Markov model. And table 1 is a simplified routing table model for all the nodes in mesh network in LPD communication.

<table>
<thead>
<tr>
<th>Dest.</th>
<th>Beam ID</th>
<th>Next Hop</th>
<th>Link W</th>
<th>$P_e$</th>
<th>$\alpha + \beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>1</td>
<td>B</td>
<td>0.8</td>
<td>0.09</td>
<td>0.4</td>
</tr>
<tr>
<td>E</td>
<td>6</td>
<td>E</td>
<td>0.5</td>
<td>0.03</td>
<td>0.9</td>
</tr>
<tr>
<td>F</td>
<td>7</td>
<td>F</td>
<td>0.5</td>
<td>0.04</td>
<td>0.9</td>
</tr>
<tr>
<td>P</td>
<td>3</td>
<td>P</td>
<td>0.5</td>
<td>0.07</td>
<td>0.5</td>
</tr>
<tr>
<td>S</td>
<td>5</td>
<td>S</td>
<td>0.8</td>
<td>0.02</td>
<td>0.8</td>
</tr>
</tbody>
</table>

With the multi-beam antennas in OLSR, there can be more than one routes between two
Figure 4.5: Basic topology for LPD communication
nodes in a network of high density. For LPD communication, because of the mobility of the adversary, thus, it’s necessary to search for more back or auxiliary paths for every relay node in a routing path. The Figure 9 shows the possible multi-path routing in the network of high density. The black line represents the original path cached locally without considering the adversary. Thus when the adversary approaches, the previously optimum path may not escape from the range of detection. However with multi-beam directional antennas, other beams can be used to find new auxiliary route to assist the transmission as the backup route. As shown in Fig.4.6 below, the route 1-2-8-3-10-4-5-6 is a auxiliary path which can achieve low probability of detection by escaping the detection range of the adversary by means of multi-beam directional antennas. Because equipped with multi-beam, thus more qualified paths can be selected out based on the routing rules for LPD. Therefore, a multi-path routing can be proposed in order to improve the performance and keep low probability of detection. In our previous paper about fence routing, we discussed about how to choose main path node[]. And for multi-path OLSR, the optimum path is selected out in the same way. However, the difference is that besides considering the link width and distance, the power level and the corresponding probabilities such as $P_e$ and $\alpha + \beta$. Also for simplicity, only a part of nodes who has the largest number of neighbors can use its multi-beam to transmit packets and these nodes are called key nodes such as node 2. And the step are shown as follows:

1. Use Dijkstra algorithm to search for the optimum based on the modified routing table and power control algorithm. Considering the mobility model of the adversary, for a single node $i$, it first controls and adjusts its power $P_i$ from low to high as described in neighbor discovery part, then it can decide the best transmission power level which can lead to the largest number of candidates $C_i$ that can act as the next hop node in LPD communication. Then from the candidate set $C_i$, select out the neighbor $N_i$ with lowest decoding error probability $P_e$ and higher detection error probability $\alpha + \beta$. Thus for every node, repeat this step, the optimum path $R = e_1, e_2, ..., e_n$ and the corresponding power set $P_1, P_2, P_3, ..., P_n$.

2. After selecting out the optimum path $R$ for node $i$, check the neighbor list at each node $i$, if there exists nodes in the neighboring beams, and the distance from node $i$ to the neighbor can
also ensure low decoding error probability $P_e$. Then node i can be chose as key nodes and apply multi-beam antennas to search for more than one routes to the destination.

3. For each key node $j$, using the modified routing table to search for a destination node $D'$ which is located behind the node $j$ in the optimum route $R$. And from node $j$, an auxiliary path $R_j$ is selected out to backup the transmission in case of adversary approaching. And in order to improve the performance, the best $K$ auxiliary paths can be implemented based on the highest detection error $\alpha + \beta$ and the lowest decoding error $P_e$. Therefore a multi-path routing from source to destination is selected out.

4. With the link state updating, the key node $j$ can determine which path to discard or kept, or if there is a new path to join based on the latest information about the network topology and adversary mobility. For example, if the adversary leaves, then all the paths can be used to improve the network performance.

Figure 4.6: Multipath OLSR for LPD in dense network
4.5.3 Routing Shortcut Mechanism

After building the multi-path routing using the previous selection algorithm, we find that although some routes can satisfy the limits for LPD, the number of hops may be significantly increased, which may lead to high end-to-end delay and compromise the overall network performance. However with the benefits of the extended transmission range in directional antennas, we find that some shortcut route can be taken to reduce the number of hops in the auxiliary routing path, because the backup nodes don’t necessarily need to be connected to their corresponding key nodes. They can also try to connect to the key nodes in the next hops in the main path. Thus in order to compute the long-range shortcut transmissions, the forwarding nodes need to have knowledge about the network topology, the position of other nodes in the path and also the location and sensitivity information about the adversary. Position information is propagated together with LS information updated periodically. A Topology Table(TT) is kept to store the LS information, corresponding sequence number and position information.

Based on the previous selection algorithm, when the adversary leaves or the detection sensitivity weakens, our modified OLSR checks each node in the routing path starting from the destination node. When we find a node located at bearing $\theta$ whose distance is shorter than the direction transmission $R_D$ of the beam $B$, then this node becomes a shortcut candidate. Also the power control algorithm will check if the node is reachable given the maximum power $P_{\text{max}}$ at which the transmission is allowed toward $\theta$ by LPD requirements. And the candidate will also be checked if it’s a neighbor of the previous key node in the main path. If yes, OLSR attempts shortcut transmission. If not, then try another shortcut candidate. And if all shortcut transmissions fail, then keep the previous OLSR path. As show in Fig.4.7, we can see that node 10 is located in the transmission range of node 9, also node 10 can meet the requirements of the power control algorithms. And based on the main path cached, we can check that node 10 is a neighbor of the key node 4. Thus in this way, we can take a shortcut transmission from node 9 directly to node 10, and the hop with node 3 can be crossed to reduce the top number of hops in this auxiliary path. Accordingly, we can apply this shortcut algorithm to all the auxiliary paths to find out some
shortcut candidates to reduce the hops of the routing path, thus improve the overall networking performance with this modified OLSR protocol in LPD communication. In Fig 10, the new shortcut path is 1-2-9-10-5-6, and we find that the number of hops is the same as the hops in the main path. And this new shortcut obviously can achieve better network performance than other auxiliary paths without shortcuts.

Figure 4.7: Multipath OLSR for LPD with shortcut in dense network

4.6 Performance Evaluations

After proposing the modified OSLR protocol for LPD communication with multi-beam directional antennas, we also do some simulations to validate the performance of this new OLSR protocol. All the nodes are randomly located in the range about 20km X 20 km. And we assume the average node distance is 1km and the average transmission range is 1.8 km for the directional
antennas. Every node has a buffer which can hold at most 200 packets and the highest link speed is about 10 Mbps. For simplicity in simulation, we normalize the adversary detection strength and set the maximum strength as 1 which means all the links within the range will be distinguished by the adversary and no auxiliary path can be built in this case. First considering the mobility and detection of the adversary, we do simulation to compare the throughput and delay under different detection strengths in the same network topology. As show in Fig.4.8 and Fig.4.9, without adversary detection, OLSR does not need to consider the LPD limits, thus can achieve the highest throughput and lowest end-to-end delay. In Fig.4.8, we can see that the influence of adversary detection in throughput is not that significant when the packet generation rate is lower than 50 packets/s. However when the packet generation rate reaches higher than 100 packets/s, we can see that the detection strength can take a significant effect in the average through. With the increasing of the detection strength, the network throughput is suppressed to a pretty low level.

![Figure 4.8: Throughput for different adversary detection ratio](image)

Also, in order to conduct a detailed research on the shortcut algorithm for the auxiliary path in the modified OLSR in LPD communication, we do some simulations on the network performance including average throughput and end-to-end delay under different detection strength of the adversary. In Fig.4.10, as the increase of the detection strength, the average
Figure 4.9: Delay for different adversary detection ratio

throughput with or without shortcut algorithm decrease sharply. And also we find that the
shortcut algorithm can help OLSR achieve much higher throughput than OSLR without shortcut
algorithm. For example, when the detection strength is 10

Figure 4.10: Throughput with and without shortcut algorithm
4.7 CONCLUSION

During my PhD study, my research has focused on routing protocols, hardware testbed and LPD communication in wireless networks. These three topics are also dependent on each other. I have learnt a lot about network protocols and security from these topics.

In summary, in chapter 2 we have presented three novel routing protocols for the hierarchical network with multi-beam directional antennas. For the high-level network, we proposed a fence routing protocol that could efficiently exploit the benefits of multi-beam directional antennas. Our results showed that fence routing could achieve higher throughput and lower delay than other protocols. For the low-level network, we have designed a moth-inspired routing protocol composed of three different sections, i.e. line section, fan section and circular area, to overcome the sink’s singular mobility. The simulation results showed that moth routing can achieve higher throughput and lower packet loss rate than conventional mobile routing protocols. We then used ant-inspired scheme to implement a cross-level routing protocol. Our results showed that the shortcut-enhanced trail-leaving scheme could significantly increase the reaching probability to the commander node.

In chapter 3, we introduce a wireless mesh network testbed equipped with directional antenna to conduct research on networking protocols. First, we select out proper hardware such as
routerboard and embedded platform OpenWrt for our testbed. Then we configure and install OpenWrt in our routerboards using cross compilation. After deployment of hardware devices, we set up and build a wireless mesh network and employ Motion to test the performance of real-time communication. We introduce two different mesh routing protocol OLSR and Batman-adv and do experiments on mesh networks with these two protocols to evaluate the network performance in different scenarios equipped with directional antennas. Also, in order to investigate the MAC protocols, we analyze and conduct experiments on the wireless drives in OpenWrt. Then we propose an effective and convenient method to modify and develop MAC protocols in mac80211. Then we do some experiments such as disabling BACKOFF in CSMA and ACK functionality in MAC layer to prove the feasibility of our proposed methodology. Future works will focus on develop new MAC protocols for multi-beam directional antennas.

In chapter 4, we have presented an innovative OLSR protocol with multi-beam directional antennas in LPD communication. In order to design this OLSR protocol, first we propose a new scan-based neighbor discovery method to exploit the benefits of mulit-beam directional antennas and avoid the flooding in network for LPD communication. Also we discuss the power control algorithms in LPD communication and introduce the metric such as decoding error probability into the routing table in OLSR. Moreover, a new MPR selection method is proposed considering the limits of LPD and benefits of multi-beam directional antennas. And based on this novel MPR selection, a modified secure OLSR protocol is designed for the mesh network with high density in LPD communication. Also in order to optimize this new OLSR protocol, we also propose a shortcut algorithm to reduce the redundancy of the hops in the auxiliary path for LPD limits. Finally some simulations are conducted to validate the efficiency of this secure OLSR protocol for mesh networks under different levels of adversary detection. And our results show that the OLSR protocol can ensure the low probability of detection and acceptable network performance when adversary detection exists in the networks. And the shortcut algorithm can take effect in a network with high density in LPD communication.
REFERENCES


