USING DIRECTIONALITY IN WIRELESS ROUTING

By

Bow-Nan Cheng

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: Electrical, Computer and Systems Engineering

Approved by the
Examining Committee:

_____
Shivkumar Kalyanaraman, Thesis Adviser

_____
Partha Dutta, Thesis Adviser

_____
Murat Yuksel, Member

_____
Biplab Sikdar, Member

_____
Boleslaw Szymanski, Member

Rensselaer Polytechnic Institute
Troy, New York

April 2008
(For Graduation May 2008)

USING DIRECTIONALITY IN WIRELESS ROUTING

By

Bow-Nan Cheng

An Abstract of a Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: Electrical, Computer and Systems Engineering

The original of the complete thesis is on file
in the Rensselaer Polytechnic Institute Library

Examining Committee:

Shivkumar Kalyanaraman, Thesis Adviser
Partha Dutta, Thesis Adviser
Murat Yuksel, Member
Biplab Sikdar, Member
Boleslaw Szymanski, Member

Rensselaer Polytechnic Institute
Troy, New York

April 2008
(For Graduation May 2008)

© Copyright 2008
by
Bow-Nan Cheng
All Rights Reserved

CONTENTS

LIST OF TABLES vi

LIST OF FIGURES vii

ACKNOWLEDGMENT xiii

ABSTRACT xvii

1. Introduction 1

    1.1 Research Objectives 4

        1.1.1 Fixed Wireless Mesh Context Objectives 4

        1.1.2 Mobile Ad hoc Network Context Objectives 6

        1.1.3 Overlay Network Context Objectives 7

    1.2 Research Contributions 8

        1.2.1 Fixed Wireless Mesh Context Contributions 8

        1.2.2 Mobile Ad hoc Network Context Contributions 9

        1.2.3 Overlay Dynamic Network Context Contributions 10

2. Literature Survey 12

    2.1 Scaling Wireless Routing Protocols 12

    2.2 Scaling using Directional/Sector Antennas 15

    2.3 Free Space Optical Communications 17

    2.4 Terahertz Free Space Optical Communications 20

3. Orthogonal Rendezvous Routing Protocol 24

    3.1 Introduction 24

        3.1.1 Key Design Considerations 26

    3.2 ORRP: Basic Scheme and Design Parameters 28

        3.2.1 Assumptions 29

        3.2.2 Theory 29

            3.2.2.1 Proactive Element 30

            3.2.2.2 Reactive Element 31

            3.2.2.3 Deviation Correction: Multiplier Angle Method . . . 33

        3.2.3 MAC Layer Feedback 37

        3.2.4 Discussion 38

3.2.4.1 Sparse Networks . . . 38
3.2.4.2 Perimeter Nodes . . . 40
3.2.4.3 MAC Layer Issues . . . 41
3.2.4.4 Load Balancing . . . 41
3.2.4.5 Three-Dimensional ORRP . . . 41
3.3 Protocol Analysis . . . 42
3.3.1 Reachability Upper Bound Analysis . . . 42
3.3.2 State Information Maintained at Each Node . . . 44
3.3.3 Average Path Stretch . . . 45
3.3.4 Additional Lines Study . . . 49
3.4 Performance Evaluation . . . 51
3.4.1 Standalone Performance Evaluations . . . 53
3.4.1.1 Effect of Number of Interfaces on Varying Network Densities . . . 57
3.4.1.2 Effect of Number of Interfaces on Network Voids . . . 60
3.4.1.3 Effect of Control Packet TTL on Varying Number of Interfaces . . . 62
3.4.1.4 State Information Maintained . . . 65
3.4.1.5 Effect of Additional Lines on Various Topologies . . . 66
3.4.1.6 Effect of Number of Lines on Network Voids . . . 69
3.4.1.7 Effect of Number of Lines on Varying Number of Interfaces . . . 71
3.4.1.8 Effect of Number of Lines on Varying Network Mobility 73
3.4.1.9 Summary of Standalone Evaluation Results . . . 74
3.4.2 Comparative Performance Evaluations . . . 77
3.4.2.1 Network Density Evaluation vs. AODV, OLSR, and GPSR with GLS . . . 81
3.4.2.2 Number of Connections Evaluation vs. AODV, OLSR, and GPSR with GLS . . . 85
3.4.2.3 Transmission Rate Evaluation vs. AODV, OLSR, and GPSR with GLS . . . 89
3.4.2.4 Summary of Comparative Evaluation Results . . . 92
4. Mobile Orthogonal Rendezvous Routing Protocol . . . 94
4.1 Introduction . . . 94
4.2 The Directional Routing Table . . . 100
4.2.1 Intra-Node Decay . . . 102

4.2.1.1 Time Decay . . . 102
4.2.1.2 Spread Decay . . . 103
4.2.2 Inter-Node Decay . . . 105
4.2.2.1 Exponential Distance Decay . . . 106
4.2.3 Design Variables and Considerations . . . 107
4.3 MORRP: Basic Scheme and Design Parameters . . . 107
4.3.1 Assumptions . . . 108
4.3.2 Near Field Operation . . . 109
4.3.3 Far Field Operation . . . 109
4.3.3.1 Proactive Element . . . 110
4.3.3.2 Reactive Element . . . 111
4.3.3.3 Data Delivery . . . 114
4.4 Numerical Analysis . . . 115
4.5 Performance Evaluation . . . 123
4.5.1 Standalone Performance Evaluations . . . 126
4.5.1.1 Effect of Time Decay Factor in Mobile Environments 129
4.5.1.2 Effect of Distance Decay Factor in Mobile Environments . . . 130
4.5.1.3 Effect of Threshold in Mobile Environments . . . 132
4.5.1.4 Effect of Spread Ratio in Mobile Environments . . . 134
4.5.1.5 Summary of Standalone Evaluation Results . . . 135
4.5.2 Comparative Performance Evaluations . . . 135
4.5.2.1 Effect of Increased Velocity . . . 140
4.5.2.2 Effect of Velocity on Medium-Sized, Lightly Loaded Networks . . . 141
4.5.2.3 Effect of Velocity on Large-Sized, Lightly Loaded Networks . . . 144
4.5.2.4 Effect of Velocity on Heavily Loaded Networks . . . 146
4.5.2.5 Effect of Increased Network Density . . . 149
4.5.2.6 Effect of Increased Data Transmission Rate . . . 150
4.5.2.7 Summary of Performance Evaluation . . . 152
5. Virtual Direction Routing Protocol . . . 153
5.1 Virtual Direction Routing . . . 156
5.1.1 Virtual Interface Assignment . . . 156
5.1.2 State Information Replication . . . 157

5.1.3 Route Query . . . 158
5.1.4 Path Deviation . . . 161
5.2 Performance Evaluation . . . 161
5.2.1 Evaluation of VDR in Churn-less Environments . . . 163
5.2.1.1 Effect of Seed and Search TTL . . . 163
5.2.1.2 Effect of Number of Virtual Interfaces . . . 165
5.2.1.3 Effect of Number of Neighbors . . . 167
5.2.1.4 Evaluation of State Distribution . . . 169
5.2.1.5 Evaluation of Network Load Distribution . . . 170
5.2.2 Summary of VDR Performance Evaluations in Static Networks 172
5.2.3 Evaluation of VDR in Dynamic Environments . . . 173
5.2.3.1 Effect of Churn on Search Success . . . 174
5.2.3.2 Effect of Churn on Path Stretch . . . 175
5.2.3.3 Effect of Churn on Network Load . . . 176
5.2.3.4 Effect of Churn on State Distribution . . . 178
6. Future Work . . . 180
6.1 Extension Future Work . . . 180
6.1.1 Virtual Direction Abstraction Analysis . . . 180
6.1.2 Hybrid Orthogonal Rendezvous Routing Protocol . . . 180
6.1.3 Analysis of Effect of Knobs in MORRP . . . 181
6.2 New Directions with Local Directionality . . . 181
7. Summary and Conclusions . . . 183
LITERATURE CITED . . . 186
APPENDICES
A. Calculating ORRP Reach Probability . . . 197

LIST OF TABLES

3.1 Multiplier Angle Method Definitions 35

3.2 Comparison of Average State Information 45

3.3 ORRP Comparison of Reach Probability vs. Number of Lines 49

3.4 ORRP Comparison of Path Stretch vs. Number of Lines 50

3.5 ORRP Default Simulation Parameter 52

3.6 Topologies Used in ORRP Simulations 53

3.7 ORRP Standalone Sims - Default Traffic Pattern Information 55

3.8 ORRP Reach Probability vs. Number of Interfaces 72

3.9 ORRP Average End-to-End Path Stretch vs. Number of Interfaces . . . 73

4.1 MORRP Parameters Affecting Successful Packet Delivery 107

4.2 MORRP Comparison of Probability of Rendezvous vs. Velocity 119

4.3 MORRP Default Simulation Parameters 125

4.4 MORRP Standalone Sims - Default Traffic Pattern Information 127

5.1 VDR Default Simulation Parameters 162

LIST OF FIGURES

1.1 Increased Medium Usage Efficiency with Directional Antennas Illustration 1

1.2 Wireless Directional Antennas and Free-Space-Optical Transceivers . . . 2

2.1 The Electromagnetic Spectrum 21

3.1 Classification of Research Issues in Position Based Routing Schemes . . 24

3.2 Orthogonal Rendezvous Routing Protocol (ORRP) Basic Example . . . 26

3.3 ORRP Proactive Element Example 30

3.4 ORRP Reactive Element Example 32

3.5 ORRP Deviation Problem Illustration 33

3.6 ORRP Multiplier Angle Method Parameter Illustration 35

3.7 Basic Deviation Correction Example with Multiplier Angle Method . . 37

3.8 ORRP Multiplier Angle Method Example 39

3.9 ORRP Multiplier Angle Method Void Traversal Illustration 39

3.10 ORRP Perimeter Routing with MAM Illustration 40

3.11 ORRP Reachability for Various Topology Areas 44

3.12 Calculating Average Path Stretch 46

3.13 ORRP Path vs. Shortest Path Ratio 47

3.14 ORRP Average Path Stretch Frequency 48

3.15 ORRP Total States Maintained vs. Number of Transmission Lines . . . 51

3.16 Effect of Number of Interfaces on ORRP Reachability and Total States Maintained 58

3.17 Effect of Number of Interfaces on ORRP End-to-End Average Path Stretch and Latency 59

3.18 Effect of Number of Interfaces on ORRP Reachability and Total States Maintained for Networks with Voids 61

3.19 Effect of Number of Interfaces on ORRP End-to-End Average Path Stretch and Latency for Networks with Voids 62

3.20 Effect of Control Packet TTL on ORRP Reachability and Total States Maintained 63

3.21 Effect of Control Packet TTL on ORRP End-to-End Average Path Stretch and Latency 64

3.22 ORRP Total States Maintained vs. Total Nodes in Network 65

3.23 Effect of Number of Interfaces on ORRP Reachability and Total States Maintained 66

3.24 Effect of Number of Lines on ORRP Reachability and Total States Maintained for Square Topologies 67

3.25 Effect of Number of Lines on ORRP End-to-End Average Path Stretch and Latency for Square Topologies 68

3.26 Effect of Number of Lines on ORRP Reachability and Total States Maintained for Retangular Topologies 69

3.27 Effect of Number of Lines on ORRP End-to-End Average Path Stretch and Latency for Rectangular Topologies 69

3.28 Effect of Number of Lines on ORRP Reachability and Total States Maintained for Square Topologies with Voids 70

3.29 Effect of Number of Lines on ORRP End-to-End Average Path Stretch and Latency for Square Topologies with Voids 71

3.30 Effect of Number of Lines on ORRP Packet Delivery Success and End-to-End Average Path Stretch for Varying Node Mobility 74

3.31 Network Density vs. Packet Delivery Success and Control Packet Bytes Sent for Various Routing Protocols 82

3.32 Network Density vs. Average End-to-End Path Stretch and Path Length for Various Routing Protocols 83

3.33 Network Density vs. Average End-to-End Latency and Aggregate Network Goodput for Various Routing Protocols 84

3.34 Number of Connections vs. Packet Delivery Success and Aggregate Network Goodput for Various Routing Protocols 86

3.35 Number of Connections vs. Average End-to-End Path Stretch and Path Length for Various Routing Protocols 87

3.36 Number of Connections vs. Average End-to-End Latency and Total Control Packet Bytes Sent for Various Routing Protocols 89

3.37 CBR Transmission Rate vs. Packet Delivery Success and Aggregate Network Goodput for Various Routing Protocols 90

3.38 CBR Transmission Rate vs. Total Control Packet Bytes Sent and Average End-to-End Latency for Various Routing Protocols 91

4.1 Mobile Orthogonal Rendezvous Routing Protocol (MORRP) Basic Example 96

4.2 The Directional Routing Table (DRT) 98

4.3 MORRP Relative Velocity Seen by Each Interface 102

4.4 MORRP Spread Decay Illustration 104

4.5 Inter-Node Decay Illustration 105

4.6 MORRP Proactive Element 110

4.7 MORRP Reactive Element 114

4.8 MORRP Data Delivery Path 119

4.9 MORRP Reachability Numerical Analysis Calculation 124

4.10 Effect of Time Decay Factor for MORRP Reachability with Various Maximum Mobility Speeds on Various Topology Sizes 130

4.11 Effect of Time Decay Factor for MORRP Far-Field Usage with Various Maximum Mobility Speeds on Various Topology Sizes 131

4.12 Effect of Distance Decay Factor for MORRP Reachability with Various Maximum Mobility Speeds on Various Topology Sizes 132

4.13 Effect of Distance Decay Factor for MORRP Far-Field Usage with Various Maximum Mobility Speeds on Various Topology Sizes 132

4.14 Effect of Near and Far-Field Threshold for MORRP Reachability and Far-Field DRT Usage with Various Maximum Mobility Speeds on Various Topology Sizes 133

4.15 Effect of Far-Field Spread Ratio for MORRP Reachability and Far-Field DRT Usage with Various Maximum Mobility Speeds 134

4.16 Effect of Velocity on Packet Delivery Success for Multiple Routing Protocols - 1,000 Connections, 100 Nodes (1300m x 1300m Networks) . . . 141

4.17 Effect of Velocity on End-to-End Path Length for Multiple Routing Protocols - 1,000 Connections, 100 Nodes (1300m x 1300m Networks) . 143

4.18 Effect of Velocity on End-to-End Latency for Multiple Routing Protocols - 1,000 Connections, 100 Nodes (1300m x 1300m Networks) 144

4.19 Effect of Velocity on Packet Delivery Success for Multiple Routing Protocols - 1,000 Connections, 200 and 300 Nodes (2000m x 2000m Networks) 145

4.20 Effect of Velocity on End-to-End Path Length for Multiple Routing Protocols - 1,000 Connections, 200 and 300 Nodes (2000m x 2000m Networks) 146

4.21 Effect of Velocity on End-to-End Latency for Multiple Routing Protocols - 1,000 Connections, 200 and 300 Nodes (2000m x 2000m Networks) 146

4.22 Effect of Velocity on Packet Delivery Success for Multiple Routing Protocols - 10,000 Connections, 100 Node (1300m x 1300m Networks) . . . 147

4.23 Effect of Network Density on Packet Delivery Success and Total Control Packet Bytes Sent for Multiple Routing Protocols 150

4.24 CBR Transmission Rate vs. Aggregate Network Goodput for Various Routing Protocols 151

5.1 Virtual Direction Routing (VDR) Basic Example 154

5.2 VDR Virtual Interface Assignment 157

5.3 VDR State Information Seeding Example 158

5.4 VDR Route Request Example 1 160

5.5 VDR Route Request Example 2 161

5.6 Effect of Seed and Search TTL on VDR Reachability 164

5.7 Effect of Seed and Search TTL on VDR End-to-End Shortest Path . . . 165

5.8 Effect of Seed and Search TTL on VDR End-to-End Path Stretch . . . 166

5.9 Effect of Number of Virtual Interfaces on Reach Probability and Path Stretch 167

5.10 VDR State Distribution Deviation 168

5.11 Effect of Number of Neighbors on VDR Reachability for Various Routing Strategies 169

5.12 Effect of Number of Neighbors on VDR End-to-End Path Stretch for Various Routing Strategies 170

5.13 VDR: State Distribution Network-wide 171

5.14 VDR: Average Queue Size Distribution Network-wide 172

5.15 Effect of Network Churn Percentage on VDR Reachability for Various Routing Strategies 175

5.16 Effect of Network Churn Percentage on VDR End-to-End Path Stretch for Various Routing Strategies 176

5.17 Effect of Network Churn Percentage on VDR Queue Size Distribution . 177

5.18 Effect of Network Churn Percentage on VDR State Distribution 178

A.1 ORRP Reach Probability Analysis Calculation 197

ACKNOWLEDGMENT

This Thesis is dedicated to my Lord and Savior Jesus Christ, for Whom I echo what Paul writes in Philippians 3:7-8 "*But whatever was to my profit I now consider loss for the sake of Christ. What is more, I consider everything a loss compared to the surpassing greatness of knowing Christ Jesus my Lord, for whose sake I have lost all things. I consider them rubbish, that I may gain Christ and be found in him....*" Thank You for being faithful long before I was born... my one desire, one hope, one joy, is to be spent and fully spent for You.

Through my years here at RPI, I'm thankful to have had the privilege and honor to be able to interact with, learn from, and grow with some amazing people. Each individual has left their mark and I am so thankful God placed each of them in my life:

My advisor Prof. Shivkumar Kalyanaraman for his constant support, encouragement, and guidance throughout my years here at RPI. I still remember once a few years back Shiv had asked me to "get a few people from our lab together and give a practice presentation". When I sent the email out, he replied back to all saying something like "everyone should have such initiative like Bow-Nan!" Even though it was totally his idea, Shiv has always been out to build his students up and encourage them. His commitment to his students from little things like telling us "students names are always first on papers" to big things like making sure we were always covered from a funding perspective so we could focus on research, was always something that challenged me throughout the years. I also admired Shiv's ability to see the big picture and anticipate future trends in the field. I remember about a year into my time here at RPI he advised me to go about a certain problem in a specific way and suggested a few possible outcomes. After about a year of looking into things, it finally dawned on me why he asked me to do it a certain way and I just sat dumbfounded that he had such foresight to see it so far in advance. It was definitely my privilege to learn and study under him.

My co-advisor Prof. Partha Dutta for his consistent patience, understanding,

and support. A few years back I had to take a few Physics classes to satisfy my NSF funding requirements. These classes were a large struggle for me because in most cases, I did not have the background needed to understand the basics. Prof. Dutta was exceptionally patient with the endless slew of questions I had and to this day, I'm still very encouraged and challenged by his commitment as an educator. I hope future generation of students can continue to see and experience his heart for teaching.

Prof. Murat Yuksel for his direction in paper writing, stimulating discussions and constant enthusiasm. Murat and I co-authored several papers together and his insight was extremely valuable. I will always remember the time his father-in-law visited and after translating the conversation for an hour in Turkish, Murat turns over to me and says "my father said he likes you because you said you wanted to marry a good Taiwanese girl even though you're away from your culture." As I look back, I'm extremely thankful for Murat's friendship and "fearless" attitude in targeting top conferences despite the odds.

Prof. Paul Schoch for his constant recommendations and support throughout the years. I had the chance to TA for his class my first semester here and it was a tremendous learning experience for me. The following semester when I was unsure where I'd get funding, Prof. Schoch saw me in the hallway and told me he had asked that my name be put on the top tier of the list of people to choose should a position open up. He gave me a chance and put his name on the line for me and I'm forever thankful. Throughout the years, I've had the privilege of TAing several classes he's taught and found him to be an extremely dedicated and effective educator. His commitment to LITEC and constant investment into future students is definitely to be admired.

Prof. Gwo-Ching Wang for her encouragements and dedication to the IGERT family. I had the privilege of meeting Prof. Wang while interviewing for the NSF IGERT Fellowship. Although at first I thought her to be pretty stern, I soon experienced her warm heart and encouraging nature. Prof. Wang's dedication to educating the whole of a student (not just academically) and her constant "going out of her way" attitude continues to challenge and encourage me daily. It is my

hope that future students can learn from her loving and caring heart.

Prof. Biplab Sikdar and Boleslaw Syzmanski for their valuable advice and suggestions while serving on my thesis committee. All the Professors in the Networks Lab (Prof. Alheusein Abouzeid, Prof. Kenneth Vastola, Prof. Koushik Kar) and other professors at RPI were crucial in developing me academically and their stimulating and challenging classes really brought an important perspective in how I do research.

My fellow Networks Lab colleagues for their camaraderie throughout the years. I'll always cherish the days when we'd sit around, let out a large sigh, and say to each other jokingly, "let's just quit" or the times when we'd say things only one of us would understand. Sitting in our windowless lab coding away together and conjuring up ideas to take over the world has been not only a unique, but a memorable experience.

I'm also extremely thankful for Ann Bruno, Priscilla Magilligan, Audrey Hayner, and the rest of the office staff for their patience and help throughout the years. I still remember the day I was sitting in the JEC lounge and Priscilla came up to me and said in a hushed secretive voice, "hey... hey... you look hungry". She then proceeded to slip me a bagged lunch saying "don't tell anyone!". I'm so thankful to have been part of this family these past few years. The numerous times they've helped me get last minute signatures for things, answered difficult questions about scheduling or administrative stuff or even simply slipping me a bag of lunch continues to encourage and challenge me daily.

Throughout my 5 years here, I'm honored to have been able to serve and grow with my KCF family who've been my joy and crown. Seeing passionate young kids come in as a ragtag band of freshman and leave as young men and women ready to conquer the world has been one of the greatest privileges of my life. I am confident that God will continue the good work He's started here and that generations of people can come to know Him personally.

Last but not least, I just wanted to thank my family, who have always been and continue to be a constant source of encouragement in every aspect of my life. Words cannot describe their love, care, concern, patience, admonishment, and direction

throughout the years. They were always the first to say "come home and rest" or "we're praying for you" when I felt tired and defeated and always the first to celebrate with me when things were going well. They are constantly giving and never taking and when I look at their lives, I can't help but be both amazed at how they constantly love and surrender themselves daily while at the same time challenged to imitate their life. I'm forever thankful, grateful, and indebted to them. It is my hope that God will continue to grow us as a family that desires to live for His glory, never forgetting that it always is, always has been, and always will be about Christ and Christ alone.

ABSTRACT

The increased usage of directional methods of communications (e.g. directional smart antennas [17], Free-Space-Optical transceivers [23], and sector antennas) has prompted research into leveraging directionality in every layer of the network stack. In this thesis, we seek to investigate how the concept of directionality can be used in layer 3 to facilitate routing under contexts of 1) wireless mesh networks, 2) highly mobile environments, and 3) overlay networks through virtual directions.

In the context of wireless mesh networks, we introduce Orthogonal Rendezvous Routing Protocol (ORRP), a lightweight-but-scalable routing protocol utilizing the inherent nature of *directional communications* to relax information requirements such as coordinate space embedding and node localization. The ORRP source and ORRP destination send route discovery and route dissemination packets respectively in locally-chosen orthogonal directions. Connectivity happens when these paths intersect (i.e. rendezvous). We show that ORRP achieves connectivity with high probability even in sparse networks with voids. ORRP scales well without imposing DHT-like graph structures (eg: trees, rings, torus etc). The total state information required is $O(N^{3/2})$ for N-node networks, and the state is uniformly distributed. ORRP does not resort to flooding either in route discovery or dissemination. The price paid by ORRP is suboptimality in terms of path stretch compared to the shortest path. However, we characterize the average penalty and find that it is not severe.

In the context of mobile adhoc networks, we introduce *Mobile* Orthogonal Rendezvous Routing Protocol (MORRP) for mobile ad-hoc networks (MANETs) which tracks node movements based on local information through a novel concept called the directional routing table (DRT) which maps interface *directions* to a *set-of-IDs* to provide *probabilistic* routing information based on interface direction. We show that MORRP achieves connectivity with high probability even in highly mobile environments while maintaining only probabilistic information about destinations. MORRP scales well without imposing DHT-like graph structures (eg: trees, rings,

torus etc). We will also show that high connectivity can be achieved *without* the need to frequently disseminate node position resulting increased scalability even in highly mobile environments.

In the context of overlay networks, we introduce Virtual Direction Routing (VDR) which takes concepts introduced in the wireless realm and adapts them to scale flat, unstructured overlay networks. VDR is a scalable overlay network routing protocol that uses the concept of *virtual directions* to efficiently perform information seeding and lookup. State information is replicated at nodes along *virtual orthogonal lines* originating from each node and periodically updated. When a path lookup is initiated, instead of flooding the network, query packets are also forwarded along *virtual orthogonal lines* until an intersection with the seeded state occurs. We show that VDR achieves path search success with high probability even with relatively low seed and search packet TTL even under high network churn. We also show that VDR scales well without imposing DHT-like graph structures (eg: trees, rings, torus, coordinate-space, etc.) and the path stretch compared to random-walk protocols (the traditional method to route in unstructured overlay networks) is very good.

In summary, we provide a framework for utilizing *directionality*, to solve issues resulting from scalability and high mobility in wireless environments. We show that directional can not only be leveraged to provide adequate routing, but can provide dramatic gains in goodput, end-to-end delay, and network reach. We then take this framework and adapt it to the wired environment to scale overlay networks.[1]

[1]This material is based upon work supported by the National Science Foundation under Grant Nos. IGERT 0333314, ITR 0313095, and STI 0230787. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

CHAPTER 1
Introduction

The proliferation of wireless technology in recent years has lead to an explosion of research in cost-effective, self-organizing, and efficient wireless technologies for use by young and old, rich and poor alike. Wireless networks allow for high flexibility in setup and relocation, ubiquitous access, and ease of use at the cost of lower throughput (due to interference, a high-loss medium, and limited available spectrum) and weakened security (anyone within range can intercept the signal). The rapid deployment of broadband wireless systems such as 802.11 wireless local area networks (WLANs), 802.16 wireless broadband and neighborhood area wireless networks, however, raises concerns in scalability. In short, as networks become larger and denser, capacity issues arise from the inherent broadcast nature of the wireless medium and limited unlicensed spectrum available to use at any given time.

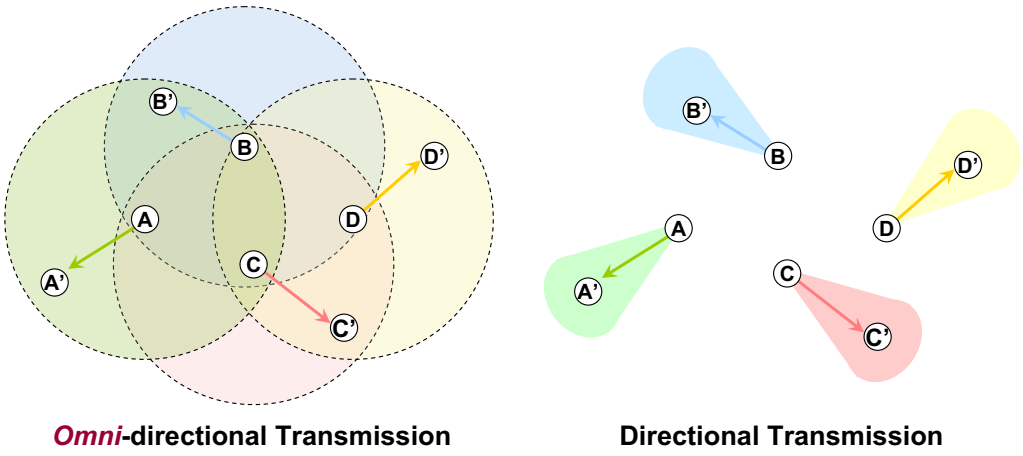


Figure 1.1: Directional antennas utilize the medium much more efficiently

Researchers have tackled the issue of scalability at all levels of the network stack through novel methods such as adjusting transmit power (to minimize interference with neighboring nodes), scheduling coordination (to assure fair use of the

medium), increasing diversity through different modulation, coding, and antenna alignment techniques, among others. More recently, there has been a lot of work on utilizing directional antennas to increase network capacity and efficient medium usage. Figure 1.1 shows the obvious motivation: using traditional omnidirectional antennas, as the network becomes denser, simultaneous transmissions quickly saturate the network. With directional antennas, however, the limited scope of the transmission allows the medium to be shared more efficiently. In addition, directional antennas have been shown to provide more reliable transmission across the board.

Previous work in directional antennas focused heavily on measuring network capacity and medium reuse [17] [18]. In these works, it was shown that with proper tuning, capacity *improvements* using directional over omnidirectional antennas are dramatic - even just 8 directional interfaces results in a theoretical capacity gain of 50X.

Figure 1.2: Wireless directional communications methods such as directional antennas and free-space-optical transceivers have become increasingly available.

Another way to scale networks is to increase network capacity. If the pipe is larger, it becomes easier to push more information through. In effort to increase bandwidth on wireless transmissions, researchers in recent years have been inves-

tigating free space optical (FSO) communications technologies as a compliment to traditional RF methods. Currently available in point-to-point links in terrestrial last mile applications and in infrared indoor LANs [48] [47], FSO has several attractive characteristics like *(i) dense spatial reuse*, *(ii) low power usage per transmitted bit*, *(iii) license-free band of operation*, and *(iv) relatively high bandwidth compared to RF*. Conversely, FSO suffers from *(i) the need for line of sight (LOS) alignment between nodes* and *(ii) reduced transmission quality in adverse weather conditions*. Yuksel et al. [23] proposed several ways to mitigate these issues by tessellating low cost FSO transceivers in a spherical fashion (see Figure 1.2) and replacing long-haul point-to-point links with short, multi-hop transmissions.

Given the seemingly large increases in medium reuse and potential for higher bandwidth in directional forms of communications, it becomes interesting to investigate how directionality can be used to facilitate and even improve wireless networks in all layers of the stack. There are several challenges associated with using directionality in mobile networks. Unlike omnidirectional antennas where neighbor reach depends almost exclusively on range, nodes using directional antennas need also take into account the neighbor's direction and map it to a specific interface in that direction. The problem is complicated even further as nodes closer to a source seemingly incur more dynamism (even small movements can affect perceived direction dramatically) while nodes farther away incur less change. Prior work in directional antennas and FSO technologies have focused on issues with the physical and MAC layer (Layer 1 and 2). Our intent is to explore how directionality can be used in the network layer (Layer 3) to route packets scalably even in highly mobile environments.

Routing in multi-hop wireless networks involves the *indirection* from a *persistent name (or ID)* to a *locator* and has grappled with the twin requirements of connectivity and scalability. Early MANET protocols such as DSR [12], DSDV [10], AODV [11], among others, explored proactive and reactive routing methods which either flood information during route dissemination or route discovery, respectively. Many improvements to existing protocols call for either limited link state dissemination [14, 34], limited route maintenance [111, 112], or complex hier-

archical structures [13] which all continue to rely on flooding to a certain degree. Even in mesh networks which are not mobile, link-states need to be flooded more often than in wired networks. Flooding poses an obvious scalability problem. In response, position-based routing paradigms such as GPSR [5] were proposed to reduce the state complexity and control-traffic overhead by leveraging the Euclidean properties of a coordinate space embedding. These schemes require nodes to be assigned a coordinate in the system, and still require a mapping from nodeID to coordinate location.

In this thesis, we will show that directionality can be used in the network layer (Layer 3) to provide efficient, unstructured, and scalable routing possibilities *without* flooding (as many topology-based routing protocols do) and *without* the need for explicit positioning (as with all position-based routing protocols). We will show that our work not only applies to wireless mesh and mobile adhoc networks, but can also be abstracted to virtual overlay networks and even wired ethernet through novel concepts such as using *virtual directions* to limit the scope of flooding.

1.1 Research Objectives

In this work, we propose and study mechanisms at the network layer that use the concept of *directionality* to route information in a scalable, unstructured manner. Beginning with a basic idea in a fixed wireless mesh scenario, we extend our work to mobile ad-hoc networks, and finally abstract it even further to overlay networks through the concept of virtual directions. This subsection identifies and illustrates the goals and current contributions in each of these cases.

1.1.1 Fixed Wireless Mesh Context Objectives

Wireless mesh networks have attracted interest in the research community because they can complement the cellular model and expand wireless reach in metro-broadband deployment [29, 30, 31]. Their fixed nature, ability to self organize and coordinate, lack of power and processor constraints, and potential to provide ubiquitous wireless access make wireless mesh networks a highly attractive option in providing last mile wireless access. In recent years, several individual communities

have set up easy-to-use community wireless mesh networks as an alternative to wired setups and infrastructure base stations. While there has been quite a bit of excitement for the wireless mesh paradigm, only recently has large scale deployment been considered.

Because of their fixed nature and highly available processing and power resources, research in wireless mesh networks in the network layer has focused heavily on providing high throughput, low latency, and low load end-to-end paths while keeping control overhead to a minimum. Routing metrics such as expected hop count (ETX) [32] have replaced traditional the "next hop" concept in wireless networks as the defacto standard for evaluating the quality of a wireless link and has led to many routing paradigms built to maximize these routing metrics. As wireless mesh networks grow in size and density, however, scalability becomes a more pressing problem. Even maintaining paths from every source to every destination incurs significant overhead because traditional routing protocols flood the network either in route discovery or dissemination phases.

Researchers have addressed the issue of scalability in wireless mesh networks through novel methods that limit flooding network-wide by adjusting dissemination TTL [34, 14], building hierarchical structures [13], employing hybrid proactive-reactive approaches [37], and using MAC backoff timers to perform routing without routes [111, 112, 113]. Even then, however, the broadcast nature of the wireless medium necessitates flooding more often than in wired networks which limits scalability.

Given the promising future of directional communications as a compliment to traditional RF omnidirectional methods, we investigate and detail how directionality can be used in layer 3 routing *without* the need for flooding either in the route dissemination or discovery phase. Specific objectives include:

- **Protocol Development** - We develop a routing protocol that utilizes the inherent nature of directionality to relax assumptions in position-based schemes such as the need for coordinate space embedding and location-to-ID mapping.

- **Protocol Analysis** - We show that a) even with directional forms of communications which inherently limit reach, high reachability can be acheived

without flooding, b) average states maintained network-wide scales reasonably well, and c) path stretch vs. shortest path is not bad.

- **Protocol Evaluation** - We evaluate our protocol against current routing protocols using the metrics of data delivery success, total state maintenance, average path length, end-to-end latency, aggregate network goodput, and control packet overhead under conditions of varying node density, number of interfaces, control packet TTL, and network topologies. We show that our protocol outperforms current proactive, reactive, and position-based protocols both with single omnidirectional transceivers and modified to support multiple directional transceivers.

1.1.2 Mobile Ad hoc Network Context Objectives

There are several challenges associated with using directionality in mobile networks. Unlike omnidirectional antennas where neighbor reach depends almost exclusively on range, nodes using directional antennas need also take into account the neighbor's direction and map it to a specific interface in that direction. The problem is complicated even further as nodes closer to a source seemingly incur more dynamism (even small movements can affect perceived direction dramatically) while nodes farther away incur less change. With protocols reliant on directionality, it is easy to see how mobility can easily disrupt routes because node movement affects routing state much more frequently. It is therefore interesting to investigate how directionality can be applied in layer 3 routing in *highly mobile environments*. To that end, in the MANET context, we investigate the following objectives:

- **Protocol Development** - We develop a protocol that uses *directionality* to solve issues caused by high mobility in mobile ad hoc networks (MANETs).

- **Protocol Analysis** - We analyze our protocol to understand the limitations of the scheme including reach probability in high mobility and total control overhead.

- **Protocol Evaluation** - We compare our protocol with existing proactive, reactive, and position-based MANET protocols under highly mobile environ-

ments to understand the tradeoff in reachability, total states maintained, control packet overhead, end-to-end latency and aggregate network throughput.

1.1.3 Overlay Network Context Objectives

Overlay networks such as peer-to-peer systems have blossomed in recent years for their ability to link and share media files across the internet ubiquitously. Overlay networks are logical networks built on top of other pre-existing networks. Nodes in an overlay network can be thought of as being connected by virtual or logical links irrespective of the underlying network. One of the biggest challenges of virtual networks like peer-to-peer systems is scalable information seeding and discovery. Efforts to mitigate flooding for content search come in general forms. In most cases, set structures are built to hash content to and search is performed by traversing the structure.

In this work, we seek to investigate whether the concept of directionality can be applied to overlay networks and abstracted to *virtual directions*. Mapping content and nodes to virtual directions allows us to be able to employ techniques from the previous sections to perform searches in a scalable, non-flooding manner in unstructured overlay topologies. This work will present address the following objectives:

- **Concept Development** - We explore how nodes in a unstructured overlay network such as peer-to-peer networks or ethernet can be mapped in a globally consistent manner without flooding such that a global virtual direction system can be established.

- **Employ Prior Techniques to Limit Flooding** - We apply techniques previously developed in using directionality for wireless networks to evaluate the viability of using virtual directions in unstructured overlay networks.

- **Protocol Evaluation** - We compare our protocol for routing in unstructured peer-to-peer systems with current standards such as normalized flooding and biased random walks to evaluate reachability, state maintenance, and end to end path stretch.

1.2 Research Contributions

In this section, we present our contributions thus far in each of the contexts listed above.

1.2.1 Fixed Wireless Mesh Context Contributions

In our work, we present Orthogonal Rendezvous Routing Protocol (ORRP), a layer 3 routing protocol using characteristics inherent to directional communications, to forward packets. ORRP is based upon two simple ideas: a) local directionality is sufficient to maintain forwarding of a packet on a straight line, and b) two sets of orthogonal lines in a plane intersect with high probability even in sparse, bounded networks. ORRP assumes that each node has directional communication capability and can therefore have a *local sense of direction* (i.e. orientation of neighbors is known based on a local North). Notice that this is an even weaker form of information than a *global sense of direction* (i.e. orientation of neighbors is known based on a global North) which necessitates additional hardware such as a compass. We show through analysis and performance evaluations that:

- ORRP achieves high connectivity ($\sim$98%) even in sparse networks with voids.
- ORRP state scales on the order of O($N^{3/2}$) and the state is distributed fairly evenly network-wide providing no single point of failure
- ORRP does not flood during route discovery or dissemination and coupled with the inherent nature of directional communications, yields over 30x the aggregate network goodput compared to AODV, 10x the aggregate network goodput compared to OLSR, and 35x the aggregate network goodput compared to GPSR with GLS in fixed environments.
- Although ORRP paths are suboptimal compared to shortest path, our analysis shows that the path stretch is not bad ($\sim$1.2) and that the end-to-end latency generated by these paths coupled with more efficient medium reuse lends itself to much lower end-to-end delays than in AODV, OLSR and GPSR with GLS.

1.2.2 Mobile Ad hoc Network Context Contributions

Heavily extending our previous work, we present Mobile Orthogonal Rendezvous Routing Protocol (MORRP) which leverages directionality and probabilistic hints to route packets in *highly mobile environments* without the need to frequently disseminate information. MORRP facilitates high mobility by abstracting the concept of rendezvous *points* to rendezvous *regions* and forwards packets *probabilistically* based on which direction a destination or rendezvous node is most likely found. These directions shift accordingly to a node's *local* velocity. For example, if a source node is moving north, a node originally *east* of the source will *seem* to be moving *south*. We introduce a novel concept called the *directional routing table* (DRT) which replaces traditional routing tables. The key contributions of MORRP include:

- **Using directionality to solve the issues caused by high mobility in MANETs** - Using only *local* information, any node is able to more efficiently "guess" the direction of a destination and forward probabilistically.
- **The Directional Routing Table** - A replacement for traditional routing tables based on purely *probabilistic routing*. DRTs map a *set-of-IDs* to a specific *direction* which eliminates the need to maintain exact routing information about nodes in a network while lessening the frequency of route dissemination.
- **Routing Based on Probabilistic Hints** - Traditional routing protocols have a hard limit on route expiration. With probabilistic routing, routing information is decayed with time and becomes less and less accurate. Below a certain threshold, the information becomes insignificant.

In our performance evaluations, we show that:

- MORRP yields above 93% reachability even in highly mobile environments for medium-sized networks and 89% reach for large-sized networks with medium density.
- Routing using MORRP accounts for an almost 10-14x higher aggregate goodput compared to AODV, OLSR and GPSR/GLS. These gains come primarily through more efficient reuse of the medium under heavy load.

- MORRP yields 15-20% higher aggregate goodput compared to modified versions of AODV and OLSR for 8 directional interfaces and also ORRP. These gains come by using directionality constructively and scalably to overcome problems inherent with directionality.

- End to end packet latency is very low under MORRP compared to AODV, OLSR, and GPSR/GLS because of more efficient medium reuse.

- As node density increases, AODV, OLSR and GPSR/GLS data delivery success drops significantly due to network saturation but does not affect MORRP much.

- MORRP sends *less* control packets than ORRP and much less than AODV, and OLSR in highly mobile situations.

1.2.3 Overlay Dynamic Network Context Contributions

Abstracting the idea of *directionality* in wireless networks, we present the concept of *virtual directions* to scale *unstructured* overlay networks. We introduce Virtual Direction Routing (VDR), a routing protocol in overlay networks that map neighbors to *virtual interfaces* which are globally consistent. Routing is done by seeding information along *virtual lines* and then sending a route request along virtual orthogonal lines until a rendezvous node is found. The seeding and request packets are biased toward IDs that are closest matched to the source and search ID respectively leading to quicker convergence. The key contributions of VDR include:

- **The Concept of Virtual Directions** - Unlike CHORD [123] or CAN [124] which build structures to facilitate quick search, VDR relies on no structures but instead utilizes the concept of Virtual Directions to eliminate the need for virtual coordinate space or DHT structures to locate items.

- **A flat, highly scalable, and resilient to churn routing algorithm for overlay networks.**

  In our performance evaluations, we show that:

- VDR provides about 9% more reach than a modified random walk seed and search strategy.
- VDR scales much better than normalized flooding
- VDR provides higher reach with less TTL than pure random walk strategies.
- VDR shows a 3-4X reach retention rate going from 0% to 50% network churn compared to VDR-R and RWR, showing itself to be much more robust to network churn.

CHAPTER 2
Literature Survey

The growth of the internet has raised important questions regarding the scalability of computer networks. Initial protocols were often flooding-based and focused on deliverability rather than scalability. As networks began to grow in size and complexity, however, new approaches to limit flooding were necessary. Hierarchical-based approaches to address scalability were heavily examined as a result. More recently, there has been a push to use primitives such as local information and random walks to scale *flat* networks in an unstructured way. Given the trends in utilizing directional communications, we believe that an important way to scale large flat networks is through using directionality. In this chapter, we will examine related work on how routing protocols have scaled in the past and how directionality has changed the game. We also introduce some emerging technologies such as free-space-optics (FSO) and tera-hertz FSO as alternative means of communications.

2.1 Scaling Wireless Routing Protocols

There has been a considerable amount of work on wireless routing protocols in recent years. Classified into five major types (reactive, proactive, hierarchical, position-based, and hybrid of the approaches), these protocols rely on different assumptions and tradeoff different metrics like connectivity, path selection, state maintenance, etc. to route packets through a network.

Reactive protocols like AODV [11] and DSR [12] perform route discovery by flooding the network and delay data from being sent until a route is found. While considerably less state needs to be maintained at each node, route-discovery flooding of the entire network can be costly and inefficient. Reactive protocols trade low overhead in lightly loaded environments for high latency time in route discovery. By contrast proactive protocols like DSDV [10] and OLSR [34] periodically broadcast routing information across the network (or in certain areas of the network), and maintain extensive routing tables at each node. Each data packet references the

routing table of every hop in the packet and forwards accordingly. Reactive routing protocols are useful in generating optimized routes because of their inherent knowledge of network topology. However, in much the same way as reactive protocols, periodic heavy network floods of control packets incur high overhead and can lead to inefficiency especially in lightly loaded systems.

As a response to to apparent issues with scalability of traditional reactive and proactive protocols, hierarchical and position-based approaches were examined. Hierarchical routing protocols such as HRP [13], LANDMAR [27], and L+ [35] splice the network into regions that maintain routing information within the area. Certain nodes within each region are selected to be gateway nodes which maintain overlay routing tables with gateway nodes from other regions. Thus, routing within each region happens normally while routing inter-region is handled by the gateway node. While an important step in achieving greater scalability, hierarchical routing techniques rely too heavily on the special nodes that maintain routing between regions, and increased complexity of reorganization make it harder to implement. In short, there is a higher rate of single points of failures. There has also been work in fish-eyed routing protocols such as HSLS [14] where nodes know a lot about nodes closer to them but little about nodes farther away. Hybrid protocols like ZRP [37] and LGF [36] that combine the various strategies add benefits but still suffer from some form of flooding and capacity constraints.

Recently, there has been a new push toward a new type of routing paradigm for wireless sensor networks that utilizes MAC backoff timers and the broadcast medium of wireless to *self-route* packets without maintaining routing tables. Self-Selective Routing (SSR) [113, 112, 114] and Self-Healing Routing (SHR) [111, 110] are examples of this class of routing protocols. After an initial flood-based "number of hops to destination" request is performed, subsequent packets are forwarded with next hops determined *after the packet has been sent*. Local leader selection techniques for MAC backoff timers based on the distance in hop count from the destination are implemented to choose a next hop to take the packet forward. While effective in situations where node failures are high such as in sensor networks because routes do not need to be maintained, SSR and SHR still uses flooding to find initial paths and

rely on hop count to determine a "gradient" direction to forward in. As a result, SSR and SHR are not suitable for mobile environments where hop count (and not merely node failure rate) changes dramatically.

Position-based protocols like GPSR [5] and TBF [7] tackle the issue of scalability by leveraging geographic position to route packets maintaining little to no state. A packet is forwarded in the "general direction" of the destination until it is reached. While highly scalable in a pure routing-only framework, position-based protocols assume location-to-address mapping techniques such as GLS [6] and HLS [139] and require either node-localization equipment, such as GPS receivers, or node-localization techniques such as AOA [133], APIT [135], and Cricket [134] to specify node positioning. The combined overhead and need for special equipment make geographic routing protocols difficult to fully realize.

In recent years, there has been a big shift from using hierarchical structured schemes, such as GLS [6], which partitions networks into grids that trade location information on a limited basis, to non-hierarchical structured schemes, such as DHT and virtual coordinate-based approaches [40] [39] [38]. DHT/virtual coordinate based approaches such as DPSR [39], VRR [38], among others, build hashes between node IDs and a set structural representation of the nodes. For example, DPSR [39] utilizes fingers that extend from a node while VRR [38] stores hashes in a circular ring format. These structured approaches not only effectively remove the need for a positioning system and network flooding, but also makes routing more scalable. The drawback, however, is that paths are suboptimal.

D. Braginsky et al. [24] proposed an *unstructured* rendezvous-abstraction routing technique for fixed sensor networks based on drawing single lines called Rumor Routing. Events are broadcasted by nodes through random walks and event request packets are sent in a similar way until it intercepts event regions. It was shown that two lines bounded in a rectangle had a 69% chance of intersecting within the rectangle. We see the trend again in routing: moving from flooding, to structured hierarchies, to unstructured techniques to forward packets.

2.2 Scaling using Directional/Sector Antennas

Directional and sector antennas have been the object of numerous studies in the past for their potential in decreasing interference and improving network capacity. Rappaport [105] describes the use of sector antennas on modern cellular base-stations which allows decreasing of the cluster size in order to improve frequency reuse without being afraid of interference. Sectoring even only 120° reduces interference significantly and increases capacity by a factor of 1.714.

The focus of much of the previous work in directional antennas has been understanding and evaluating its spatial reuse property. Nasipuri et al. [19] modified the RTS and CTS exchange in 802.11 to support directionality and showed through simulations a throughput improvement of 2-3 times over omnidirectional antennas. The primary aim of the work was to minimize routing overhead by using directional antenna elements for propagating routing information as routing overheads from omnidirectional transmissions can be costly. Ko et a. [20] proposed directional MAC (D-MAC), a revamp of the current 802.11 MAC scheme to support both directional and omnidirectional operation. The D-MAC scheme showed a throughput boost of about 2 times normal 802.11 operation.

Building on top of previous work, Choudhury et. al. [16] designed the Multi-Hop RTS MAC (MMAC) protocol which uses multi-hop RTS's to establish links between distant nodes, and then transmits CTS, DATA, and ACK packets over a single hop. Their results show that MMAC outperforms 802.11 but the performance is depending on the topology and flow patterns. Ramanathan [17] evaluates the performance of ad hoc networks with beamforming antennas arriving at several important conclusions:

- beamforming antennas yield up to a 118% improvement in throughput and up to a factor of 28 reduction in end-to-end delay in static networks.
- Even simple channel access techniques which allow parallel transmissions yield dramatic results.
- Link power control is essential in exploiting beamforming antennas

- With respect to spatial reuse, switched beams are nearly as good as steered beam antennas.
- Directional neighbor discovery is very effective in low densities.

Work in measuring capacity gains using directional antennas is not limited to simulation evaluations. Building on network capacity bounds formulated by [117], Yi et al. [18] presents an analytical model for evaluating network capacity using directional antennas. Their work showed that with proper tuning, capacity *improvements* using directional over omnidirectional antennas are dramatic - ranging from a factor of $\frac{2\pi}{\sqrt{\alpha\beta}}$ for planned networks to a factor of $\frac{4\pi^2}{\sqrt{\alpha\beta}}$ for random networks where α and β are the beamwidths of the transmitting and receiving antennas. These results show that even by simply using 8 directional interfaces, there is a *gain* in capacity by 50x.

Directional antennas not only improve network capacity, but have been shown to be more stable in terms of link quality and not as affected by routing metrics. Chebrolu et al. [22] showed that 802.11 long distance links using directional antennas result in almost "wire-like" characteristics with error rates as a function of the received signal strength behaving close to theory, time correlation of any packet errors being negligible across the range of time-scales, and links being robust to rain and fog. Under such conditions, routing metrics for wireless links become less and less important.

More recently, there has been a lot of work showing how current ad hoc routing protocols fail when using directional antennas and how they can be adapted to work using directional antennas. Choudhury et. al. [15] proposes Directional DSR (DDSR), a modification to DSR [12] to address issues with interface *handoff*, backoff, and neighbor discovery. Gossain et al. [8] presents Directional Routing Protocol (DRP), a cross layer protocol inspired by DSR which handles route discovery, establishment, maintenance, and route recovery mechanisms all using directional antennas. While much of these efforts in using directionality in the routing layer are important, they come more as a response to having directional communications rather than leveraging directionality as a benefit in routing. By contrast, our work

uses *local directionality* to route packets rather than simply modifying traditional routing protocols to send out using directional interfaces.

2.3 Free Space Optical Communications

In recent years, researchers have looked into using free-space-optics (FSO) to compliment traditional RF networks [53, 23]. Legacy optical wireless, also known as free-space-optics (FSO) communication technologies use high-powered lasers and expensive components to reach long distances. Thus, the main focus of the research has been on offering only a single primary beam (and some backup beams) or use expensive multi-laser systems to offer redundancy and some limited spatial reuse of the optical spectrum [88, 102]. The main target application of these FSO technologies has been to serve commercial point-to-point links (e.g. [101, 100]) in terrestrial last mile applications and in infrared indoor LANs [96, 102, 95, 94, 93, 97]. Though cheaper devices (e.g. LEDs and VCSELs) have not been considered seriously for outdoor FSO in the past, recent work shows promising success in reaching longer distances by aggregation of multiple LEDs or VCSELs [98, 99].

Another line of work on FSO communications has been on achieving reliable *high-speed* links for optical interconnects, where auto-alignment or wavelength diversity techniques are reported to improve the misalignment tolerances in 2-dimensional arrays [77, 76, 75, 74, 78, 79]. These techniques work only over small ranges (e.g. 1μm - 1 cm) and some of these are cumbersome involving highly sensitive mechanical tracking instruments. Moreover, they are designed to improve the tolerance to movement and vibration but not to handle mobility.

While phased array structures are well known in the RF world, and work in the last decade has made multiple spatial channels possible using smart antenna techniques [72, 73], RF cannot match the high degree of integration and number of spatial channels possible using FSO. For example, a small 1sqft array (smaller than a laptop screen) can allow the integration of around 1000 pairs of transceivers. With each pair operating at 100 Mbps, this can lead to an aggregate backhaul capacity of 100 Gbps! No RF technology can match this performance; expectations of RF techniques with smart antenna techniques in the 5 GHz unlicensed spectrum top off

at about 1-5 Gbps. Moreover, RF with spatial arrays (especially in unlicensed bands like 5 Ghz) requires placement of complex, high-speed, mixed signal electronics that would make the unit operate at higher power levels and would be far more expensive. It is interesting to note that today's laser-based FSO techniques could be extended to form spatial arrays, but such equipment would be very costly and high-power. Moreover, such laser-based equipment would not have the form factor, weight and power characteristics to be mounted on ad-hoc infrastructures like balloons, tree-tops etc.

In addition to the above benefits and potential for spatial integration/spectral reuse, FSO also has critical limits relevant to achieving high-speed communication links. First, *FSO requires a clear line-of-sight (LOS)* between the transmitter and receiver of a link because these frequencies are absorbed by almost any obstruction (eg: walls, trees, automobiles etc). Though non-LOS optical operation is possible under certain conditions (e.g. indoor infrared [94]), such operation is primarily for short-range, half-duplex LOS (a.k.a Point-and-Shoot (P&S)) links only within a single room (very short distance of 1-10m), expecting the availability of multiple reflected LOS paths. Indoor infrared also requires stringent eye-safety requirements: IEC Class 1 allowable exposure limit (AEL) [69, 70].

The LOS issue has been tackled recently through the use of very low cost components organized in a multi-hop network to get around LOS issues. Akella et al. [106] showed that error characteristics of FSO over multiple hops are very promising. Further, a promising recent trend is the usage of optoelectronic solid-state devices like LEDs for lighting. Packaging of multiple LEDs is shown to be very promising for durable lighting devices and commercializations are taking place for various daily life usage of such solid-state lighting devices [68, 67, 90]. Similar to IrDA deployment in hundreds of millions of devices today, we believe that massive deployment of integrated optoelectronic devices which can *illuminate* and *communicate* will provide sufficient FSO-MANET nodes to overcome the LOS problems. In addition, as FSO-MANET will be complementary to the traditional RF MANETs, LOS problem will be a secondary problem.

The second FSO limitation in regards to high-speed communications is *various*

forms of attenuation. Recall that current FSO links in the field using lasers are limited to a few kilometers [107], though satellite communications has routinely used FSO links ranging several thousands of kilometers. The terrestrial limitations occur primarily due to atmospheric attenuation (e.g., fog, rain, snow) and geometric attenuation (due to beam divergence). Considerable FSO work especially in industry has been on characterizing link availability under various conditions [71, 101, 100], with higher availability in clear-conditions. These studies showed that dense fog affects FSO transmission far more than other conditions, and that an average of 99.98% in all conditions is considered very good availability for FSO. Also, addition of microwave RF backup provides even higher (carrier-class) availability percentages (e.g. 99.999%).

The key limitation of FSO regarding *mobile* communications is the fact that *LOS alignment must be maintained* for communication to take place successfully. Since the optical beam is highly focused, it is not enough if LOS exists: the transmitter and receiver pair should be aligned; and the alignment must be maintained to compensate for any sway or mobility in the mounting structures. Mobile communication using FSO is considered for indoor environments, within a single room, using diffuse optics technology [94, 96, 103, 92, 91, 90, 89, 87]. Due to limited power of a single source that is being diffused to spread in all directions, these techniques are suitable for small distances (typically 10s of meters), but not suitable for longer distances.

For outdoors, *fixed* FSO communication techniques have been studied to remedy small vibrations [85, 86], swaying of the buildings have been implemented using mechanical auto-tracking [88, 84, 83] or beam steering [82], and interference [81] and noise [80]. LOS scanning, tracking and alignment have also been studied for years in satellite FSO communications [66, 65]. Again, these works considered long-range links, which utilize very narrow beamwidths (typically in the microradian range), and which typically use slow, bulky beam-scanning devices, such as gimballed telescopes driven by servo motors.

FSO spherical structures were studied and some of its elementary features such as alignment were built and operated at very short distances and very low speeds

[23]. These studies showed promising results and we plan to build several fully-structured prototypes of 3-d FSO spheres which will constitute a lab-based prototype of a demonstrable FSO-MANET working at high speeds and longer communication distances. FSO is very attractive for power-scarce MANET applications such as sensor networks [108].

As discussed earlier, in comparison to RF physical communication characteristics, FSO has critical differences in terms of error behavior, power requirements, etc. Implications of these physical FSO characteristics on higher layers of the networking stack has been studied in recent years. The majority of the FSO research in higher layers has been on topology construction and maintenance for optical wireless backbone networks [64, 63, 62]. Some work considered dynamic configuration [61], node discovery [60], and hierarchical secure routing [59, 58] in FSO sensor networks. However, no deep investigation of issues and challenges that will be imposed on MANETs by FSO has been performed.

A key FSO characteristic that can be leveraged at higher layers is its *directionality* in communication. Though the concept is similar to RF directional antennas, FSO can provide much more accurate estimations of transmission angle by means of its directionality. Previous work (including ours) showed that directionality in communication can be effectively used in localization [109, 57], multi-access control [16, 55], and routing [1, 15, 46, 56, 8]. In addition to directionality, our proposed FSO nodes introduce highly-intermittent disconnectivity pattern (i.e. aligned-misaligned pattern) which affects transport performance [23]. Also, establishment of an FSO communication link implies that the space between the communicating nodes is Euclidian, which can be leveraged to better design routing and localization protocols.

2.4 Terahertz Free Space Optical Communications

Another interesting direction researchers have looked into to provide high-speed communication is using Terahertz (THz) FSO [145, 141, 142]. Up until recently, the Terahertz (10^{12} Hz) region of the electromagnetic spectrum (from about 100 GHz to 10 THz) has been relatively unexplored due its general inaccessibility [140, 145, 144]. As shown in figure 2.1 THz radiation lies in between two well-

understood frequency ranges: the radio and optical frequency spectrums.

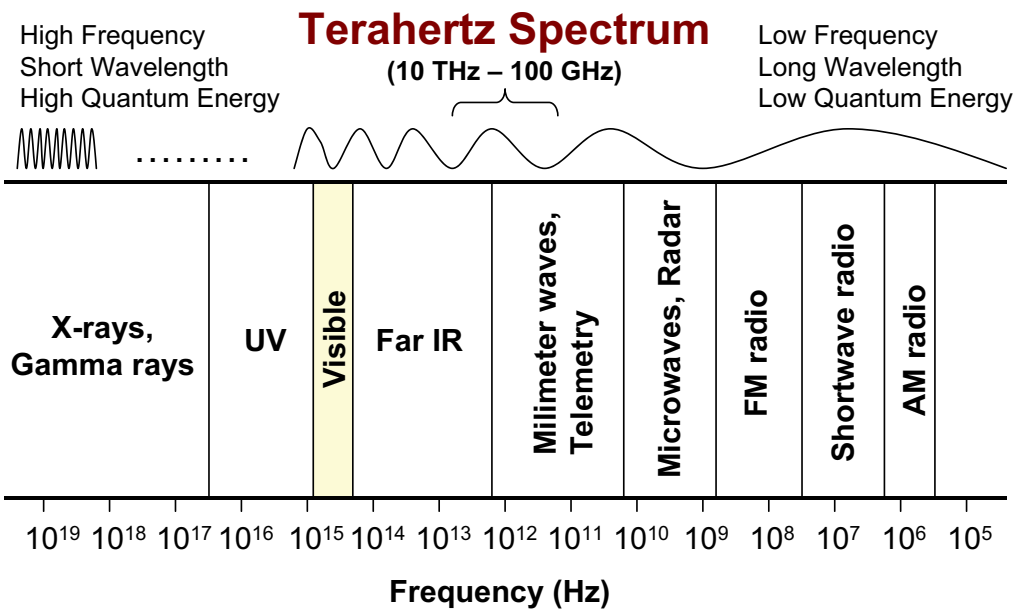


Figure 2.1: The Electromagnetic Spectrum

In the lower frequencies, which include radio frequencies (RF) for AM and FM radio and microwaves the sources are based on electron generation governed by classical electro-magnetic theory. Radiation in these frequencies propagate with wave-like characteristics and pass through most objects relatively easily with little attenuation. By contrast, higher frequencies encompass the optical regime including IR radiation, visible light, and UV. In these frequencies, light is generated by quantum transitions and typically propagates in free-space according to laws of optics. The result is a very directional beam that often requires LOS for successful propagation.

Terahertz radiation sits in between these two frequencies and exhibits characteristics of both: on one hand, it is very directional in nature, behaving much like light and on the other hand, it is very wave-like in nature, seeing physical objects as transparent. Only recently, through the development of modern micro-fabrication techniques and the capability to fabricate structures on the order of the THz wavelength of a few 10s of micrometers in electronic and hybrid optoelectronic devices,

was the doorway opened to the development of THz sources and receivers.

There are many benefits of THz communication systems that make it an attractive option to explore. First, frequencies above 300 GHz are unallocated by the Federal Communications Commission (FCC), opening up large swaths of bandwidth and frequency bands to operate at. Second, unlike FSO, THz FSO waves propagate through opaque objects resulting in the possibility of room to room communications. Third, because THz radiation operates very similarly to RF waves, there is possibility for higher bandwidth for communications without resorting to switching to all-optical solutions. The all-electronic conversion from the THz carrier to microwave links are relatively straightforward.

One of the major drawbacks of THz waves, however, is the heavy absorption through atmospheric water vapor and the lack of available high power sources. This makes THz communications relatively useful for short-range or space-born communications and only recently has there been much success in utilizing THz communications. Kleine-Ostmann et al. [140] first demonstrated in 2004 the transmission of an audio signal via a THz communication channel using a room temperature semiconductor THz modulator based on the depletion of two-dimensional electron gas. Instead of a continuous carrier wave, they used THz pulses that were amplitude modulated.

The available frequency range, ability to propagate through opaque objects and the relative short-range of the communications present several interesting applications. It is easy to see that THz communications might have an impact in short-range tactical communications where the maximum distance is about 100 meters. The beamlike qualities of THz waves reduce the ability of distance adversaries to intercept the transmissions and the high atmospheric absorption and attenuation lead to virtually undetectable waves at farther listening posts. Additionally, adversaries might not have the technological capability to detect, intercept, jam, or "spoof" a THz signal. Another possible application is to compliment traditional RF and FSO networks with another failsafe method of communications whether in satellite or infrastructure-mode networks.

Because of the potential for high speed communications in the THz regime

and the *directional* nature of the THz beam, it becomes increasingly interesting to see how directionality can be leveraged in this upcoming technology.

CHAPTER 3
Orthogonal Rendezvous Routing Protocol

3.1 Introduction

Wireless mesh networks have attracted interest because they can complement the cellular model and expand wireless reach in metro-broadband deployment [29, 30, 31]. Routing in multi-hop wireless networks has grappled with the twin requirements of connectivity and scalability. Early MANET protocols such as DSR [12], DSDV [10], AODV [11], OLSR [34], among others, explored proactive and reactive routing methods which either flood information during route dissemination or route discovery, respectively. Improvements such as SSR [112] and SHR [111] which uses MAC backoff timers to do local leader selection after packets have been sent to limit route maintenance still rely on a flood-based initial route discovery. Even in mesh networks which are not mobile, link-states need to be flooded more often than in wired networks. Flooding leads to issues with scalability. Leveraging positioning information to route packets instead of the traditional topology-based techniques, position-based routing paradigms such as GPSR [5] were proposed to reduce the state complexity and control-traffic overhead by leveraging the Euclidean properties of a coordinate space embedding. These schemes require nodes to be assigned a coordinate in the system, and still require a mapping from nodeID to coordinate location which is often assumed. In this paper, we focus on routing with even less information, i.e. scalable, efficient routing without explicit positioning.

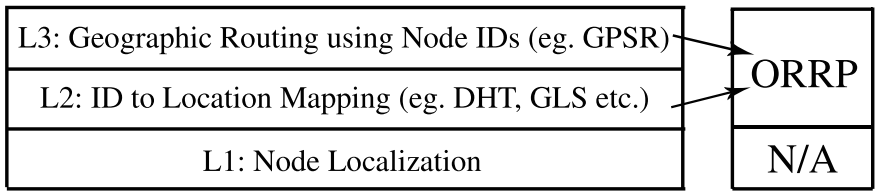


Figure 3.1: Classification of research issues in position based routing schemes

A recent trend in wireless communications has been the desire to leverage di-

rectional forms of communications (eg. directional smart antennas [17] [15], FSO transceivers [23]) for more efficient medium usage and scalability. Previous work in directional antennas focused heavily on measuring network capacity and medium reuse [15] [17] [18]. It has been shown that directional antennas can be used to provide high quality links which tend to be more wire-like and less sensitive to routing metrics [22]. In this chapter, we utilize directionality for a novel purpose: to facilitate layer 3 routing without the need for flooding either in the route dissemination or discovery phase.

Our protocol, called Orthogonal Rendezvous Routing Protocol (ORRP) is based upon two simple ideas: a) local directionality is sufficient to maintain forwarding of a packet on a straight line, and b) two sets of orthogonal lines in a plane intersect with high probability even in sparse, bounded networks. ORRP assumes that each node has directional communication capability and can therefore have a *local sense of direction* (i.e. orientation of neighbors is known based on a local North). Notice that this is an even weaker form of information than a *global sense of direction* (i.e. orientation of neighbors is known based on a global North) which necessitates additional hardware such as a compass. Figure 3.2 illustrates an example operation of ORRP.

Consider a source node S that wishes to send packets to a destination node D. Both nodes S and D have their own local notions of orientation. Source S sends route discovery packets in four orthogonal directions and the destination D does likewise for route dissemination packets. The route discovery packets will rendevous at a node touched by a route dissemination packet at up to two rendezvous points on the plane. We refer to the intersection that facilites a shorter path as the rendezvous node R. Node R directs packets from source S to the destination D. Node D's state is only maintained on the two orthogonal lines, which implies that the total state complexity is $O(N^{3/2})$ for an network of N nodes. If each node chooses its local orthogonal directions independently, ORRP state information is fairly evenly distributed throughout the topology resulting in no single point of failure. Further, there is no flooding by either source S or destination D. All these factors enable scalability without imposing the requirement of an explicit hierarchical structure

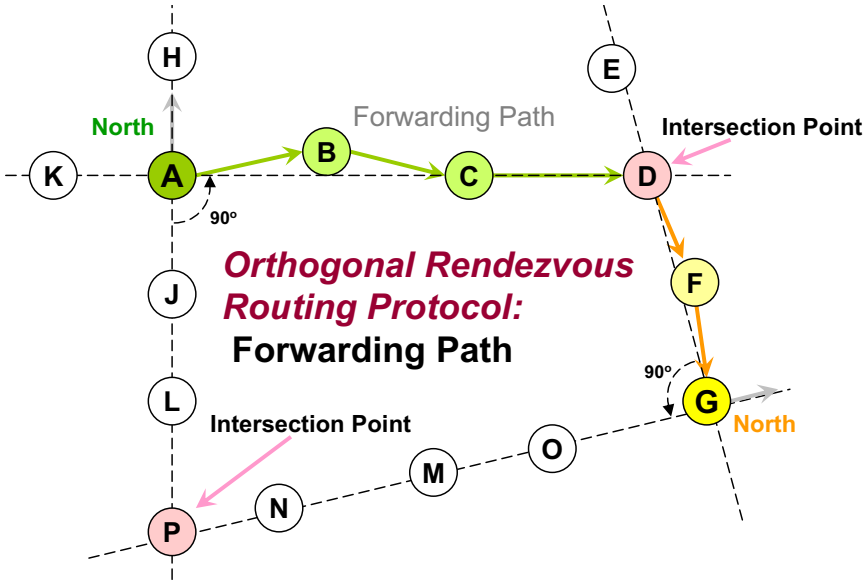


Figure 3.2: ORRP Basic Example: Source sends packets to Rendezvous node which in turn forwards to Destination

[6, 13]. In other words, ORRP offers a scalable, *unstructured* indirection method for routing in contrast to the hierarchically structured methods suggested in prior work. However, the ORRP paths chosen are suboptimal, i.e. have a stretch factor greater than 1 compared to the corresponding shortest paths. However, we show that this factor is not too large on average.

The rest of this chapter is organized as follows: We first outline key design issues of ORRP in the next subsection. Section 3.2 deals with the specifics of ORRP including assumptions, concepts and examples. Section 3.3 provides performance analysis including basic Matlab simulations to formulate upper bounds on reachability and average shortest path while section 5.2 examines these issues in more realistic packetized simulation environments.

3.1.1 Key Design Considerations

To fully realize the implications of ORRP, it is important to understand what issues traditional geographic routing protocols face. The problem of end-to-end wireless geographic routing using network localization can be broadly categorized

into three layers as shown in Figure 3.1. The lowest layer L1 is the localization scheme that obtains node coordinates [6] [4] while the second layer L2 maps these coordinates to node "identifiers" like a name or a number. Once these two are established, the third layer L3 uses this information to perform geographic routing. Current research in geographic routing protocols (e.g. GPSR [5], TBF [7], GLS [6], Landmark [27]) often tackle one of the three layers and assume the others to be a given. When taken separately, schemes in each layer can be shown to be extremely scalable. However, combining the effects of maintenance of the three layers can be rather costly. ORRP provides a simple, lightweight alternative to tackle layers L2 and L3 while removing the need for layer L1 all-together.

Specifically, ORRP focuses on and attempts to optimize based on the following considerations:

- *Connectivity Under Less/Relaxed Information compared to Position-based Protocols* - Position-based protocols such as GPSR [5] or TBF [7] operate under the assumption that each node has a globally consistent view of its own as well as other's geographic positions. ID-to-location mappings (location discovery problem) are assumed to be a given. While this assumption is appropriate given the lowering cost of GPS receivers and several proposed methods of solving the location discovery issue [25, 26], maintaining global view of the network in this way can be costly, unavailable (e.g. GPS receivers need "sky access" and cannot be used indoors) and might not be scalable in larger or highly dynamic networks. ORRP eliminates the need for location discovery by utilizing the fact that two pairs of orthogonal lines mostly have intersection points. These "rendezvous points" act as forwarders of data increasing scalability.

- *Efficient Medium Reuse* - Topology-based routing protocols generally fall into two camps: proactive (e.g. DSDV [10]) and reactive (e.g. DSR [12], AODV [11]). Proactive protocols consistently flood the network with control packets to maintain up-to-date routing tables at each node. While this ensures high packet delivery success even in mobile environments, scalability is limited due to the sheer number of control packets needed to maintain up-to-date routing

tables. Reactive protocols attempt to solve this issue by requesting routes "on demand" and then caching those routes. While this works for less mobile environments, similar issues with scalability arise. ORRP mitigates these issues by forwarding control packets proactively only in orthogonal directions thereby freeing the medium for data, and then reactively requesting routes when one is not cached and is needed. These route requests do not flood the network unnecessarily because they are transmitted only in orthogonal directions and once a rendezvous node receives these request packets, it stops the forwarding.

- *Less State Information Needed to be Maintained* - Because ORRP only maintains routing information in orthogonal directions, scalability is increased.

In order to optimize and bring out the advantages listed above, there are several tradeoffs associated with ORRP:

- *Increased Path Stretch* - ORRP optimizes connectivity and efficient medium reuse with little agreed-upon information. The cost of less information is that packets often take paths longer than shortest path. We will show that although ORRP paths are suboptimal, under normal circumstances, the average path stretch is close to optimal.

- *Limited Reachability* - Due to possibility of no intersection of orthogonal lines, some source and destination pairs might not have rendezvous points resulting in unavailable paths. While several corrective measures are suggested in ORRP, we will show that under normal operation, the packet delivery success is extremely high.

3.2 ORRP: Basic Scheme and Design Parameters

In this section, we will detail the assumptions, specifications, and mathematical aspects of ORRP. Specifically, we will 1) address assumptions made by ORRP including hardware requirements and other cross-layer abstractions, 2) detail the proactive and reactive elements of ORRP, and 3) explain path deviation correction and void traversal with the Multiplier Angle Method (MAM).

3.2.1 Assumptions

ORRP relaxes many of the assumptions made by position-based routing protocols while still providing high connectivity. ORRP makes no assumptions on location discovery and uses packets forwarded in orthogonal directions to find paths to the destination from a given source. To do so, ORRP makes three major assumptions:

- *Neighbor Discovery* - We assume that any given node will know (i) its 1-hop neighbors and (ii) the given direction/interface to send packets to reach this neighbor. In practice, the link layer often takes care of neighbor mappings through ARP and MAC packets. Note that this assumption can be removed by implementing a simple "hello" protocol at the routing layer. In our evaluations of ORRP in non-mobile environments, our results include overhead associated with periodically broadcasting "hello" packets to neighbors. The reason we include this requirement in our description of ORRP is because in Chapter 4, we use this assumption in all our evaluations.

- *Local Sense of Direction* - Each node must have its own local perception of direction (i.e. each antenna/transceiver knows its own orientation with respect to the "local north").

- *Ability to Transmit/Receive Directionally* - Nodes must be capable of communicating directionally over their transceivers. This can be done by various hardware including directional and smart antennas [15], and FSO transceivers [23]. FSO transceivers are a particular interest due to their fine-grained transmit angle and ability for several dozen to be tesselated together oriented in several directions on a single node [23].

3.2.2 Theory

The basic concept behind ORRP is simple: knowing that in 2-D Euclidean space, a pair of orthogonal lines centered at different points will intersect at two points at minimum, rendezvous points can be formed to forward packets as shown in Figure 3.2. To achieve this, ORRP relies on both a proactive element which makes up

the "rendezvous-to-destination" path and a reactive element which builds a "source-to-rendezvous" route on demand. Nodes periodically send ORRP announcement packets in orthogonal directions and at each node along the orthogonal route, the node stores the route to the source of the ORRP announcement and the node it received the announcement from (previous hop). When a source node wishes to send to some destination node that it does not know the path for, it sends out a route request packet (RREQ) in its orthogonal directions and each subsequent node forwards in the opposite direction from which it receives the packet. Once a node containing a path toward the destination receives an RREQ, it sends a route reply packet (RREP) in the reverse direction back to the sender and data transmission begins. In the following subsections, we will detail and explain the tradeoffs associated with each element of ORRP.

3.2.2.1 Proactive Element

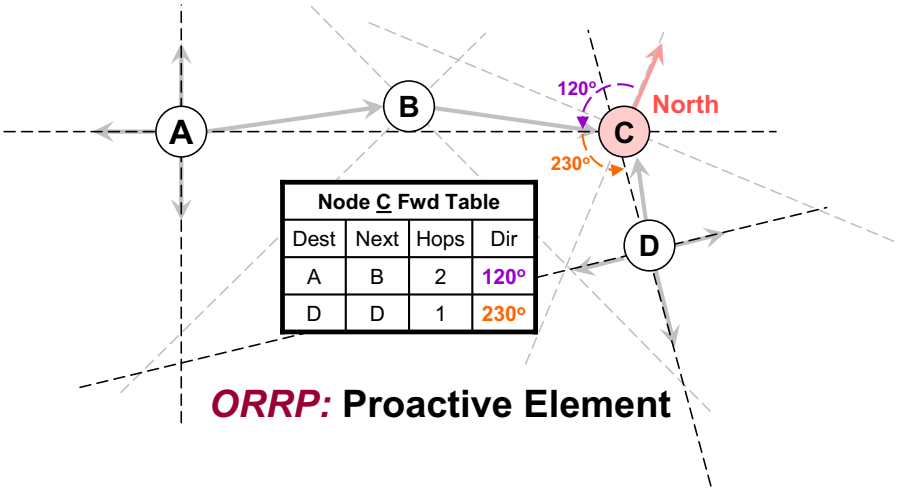


Figure 3.3: ORRP Proactive Element: Announcements used to generate rendezvous node-to-destination paths.

In order for a source and destination to agree upon a rendezvous node, pre-established routes from the rendezvous node to the destination must be in place. Because each node has merely a local sense of direction, making no assumption on position and orientation of other nodes in the network, it can only make forwarding

decisions based on its own neighbor list. After a set interval, each node sends ORRP announcement packets to its neighbors in orthogonal directions as shown in Figure 3.3. If there is more than one neighbor in a specific interface direction, ORRP randomly chooses one to send to. When those neighbors receive these ORRP announcement packets, it adds the source, previous hop, and hop count into its routing table as a "destination-next-hop pair" and forwards it out the interface exactly opposite in direction from the interface it received the packet. Although we currently only consider hop-count to be the metric for path selection, it is easy to adapt ORRP to use other heuristics such as ETX [32] among others.

It is important to note that each node does *not* maintain a complete picture of the network which limits the state information maintained through constant control packet flooding/updating. This results in increased scalability as the network is freed up for data transport. Moreover, only forwarding in orthogonal directions provides enhanced medium reuse. Based on mobility speeds, energy constraints, and other factors, parameters that can be modified for higher performance of ORRP announcements include *announcement send interval* and *forwarding entry expiry time*. Because the forwarding table only maintains information about destination and next hop, overhead in storage and maintenance is minimized as well.

3.2.2.2 Reactive Element

In order to build the path from source to rendezvous node, an on-demand, reactive element to ORRP is necessary. When a node wishes to send packets to an destination that is not known in its forwarding table, it sends out a route request packet (RREQ) in all four of its orthogonal directions. Again, if there is more than one neighbor associated with a specific interface, ORRP will randomly choose one and send it to the neighbor. When neighbor nodes receive this RREQ packet, it adds the reverse route to the source into its routing table and forwards in the opposite direction. It is important to note that RREQ paths might be different than announcement paths if there are more than one neighbors associated with a specific interface direction.

In a 2-D Euclidean plane, by sending a RREQ packet in all 4 of its orthogonal

directions, it is highly likely to encounter a node that has a path to the destination. When a node with a path to the destination receives the RREQ, it sends a RREP packet back the way the RREQ came. Because each node along the path stored a reverse route to the source, it is able to forward the RREP back efficiently after recording the "next-hop" to send to this particular destination. When the source receives the RREP, it generates a "destination-next-hop" routing entry and forwards packets accordingly.

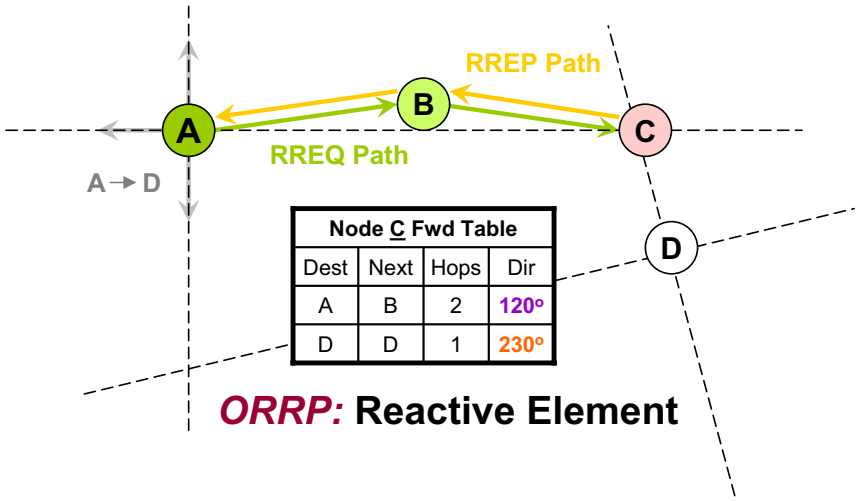


Figure 3.4: ORRP Reactive Element: RREQ and RREP Packets to generate source-to-rendezvous node paths

Figure 3.4 illustrates the process of sending RREQ and RREP packets while showing the ORRP path selected. Unlike AODV, DSR or other reactive protocols, RREQ packets are *not* forwarded until they reach the destination, but only until it intersects a rendezvous node. The proactive element of ORRP takes care of the rendezvous node-to-destination path.

It is important to note that ORRP path is *not* equivalent to the shortest path for most cases. As mentioned earlier, we gained connectivity under relaxed assumptions at the cost of suboptimal path selection (increased path stretch). We will show later, however, that the path selection is close to optimal resulting in a fairly nonexistent cost.

3.2.2.3 Deviation Correction: Multiplier Angle Method

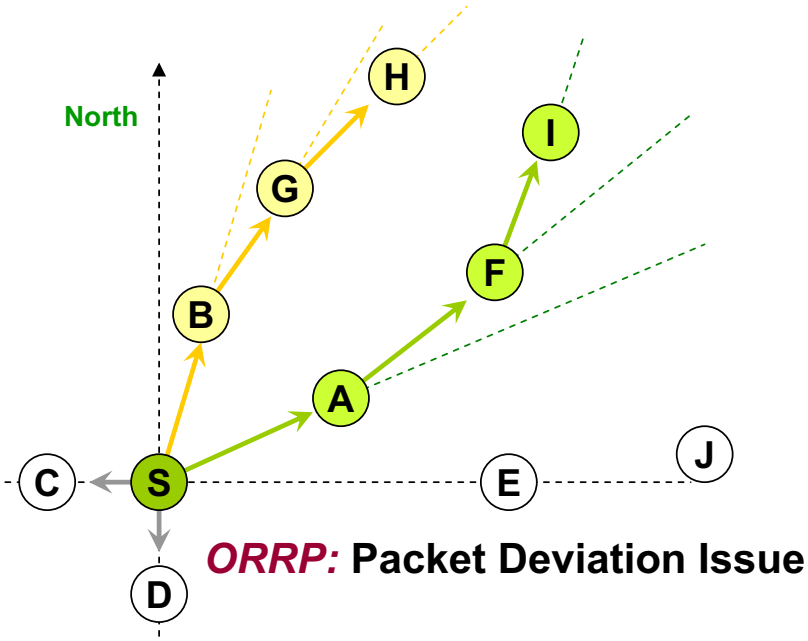


Figure 3.5: ORRP Deviation Problem Illustration

Up until now, we have considered only situations where nodes forward in orthogonal directions assuming that neighbors are all aligned on a straight line. In reality, however, straight line paths in random networks rarely exist. Figure 3.5 shows the potential problems associated with trajectory deviations. In the example, node S is sending ORRP announcement packets out on its orthogonal faces. While node C and D are perfectly aligned toward the orthogonal lines, sending the packets North and East pose a problem. Because node E is out of range of node S, node S elects to send the ORRP announcement packet to node A which is the closest to orthogonal in the east direction without forming an angle larger than 90° from the source.

When node A receives the announcement packet, it wishes to forward on its opposite interface but finds itself in a similar situation as node S and forwards the packet to node F which in turn faces the same crisis and forwards to node I. It is easy to see that node I is very far off the line dictated by the orthogonal path. Similar issues can be apparent in all other directions as well.

In the same way, in the "north direction", node S forwards to node B which, by virtue of the fact that no other nodes in the exact opposite direction from the receipt transmitter exist, forwards to node G. Even though node G has a neighbor that is in the exact opposite direction from the direction it received node B's packet, it is clear to see that we are still very far from the line dictated by the orthogonal path. This issue further complicates things when paths that are supposed to be orthogonal to each other intersect due to deviations in sending.

Although ORRP works on *path intersections* and as a result, does not need to enforce the rule that packets sent in orthogonal directions must remain true to their path, upholding this rule increases the probability of finding intersections. [24] shows that that two straight lines randomly drawn in a Euclidean plane have a 69% chance of intersecting within a given area. We will show in later sections (section 3.3) that *two* pairs of orthogonal lines have about a 98% chance of intersecting.

To address the deviation issue, it is important to clarify a few key concepts and limitations. First, deviation corrections can only be done when the deviation is greater than the conical spread of the directional antenna or transceiver. Interfaces oriented in a circular fashion, so that each of the antennas attached to a particular node operate at a set angle from the local "north", have a coverage much like a pizza pie. Depending on the beam width and assuming no overlap in spread, a node can be at various degrees of deviation from the actual orientation of each particular antenna even though it is within the beam spread/coverage area. ORRP does *not* deal with deviations that occur *within* one antenna coverage area.

Next, ORRP assumes that the relative distances from one hop to another are relatively equal. In dense networks, this is a safe assumption due to the sheer volume of nodes. It will be shown that in sparse networks, lack of nodes leads to mute relative distance issues. Finally, all deviation corrections are done at the RREQ and ORRP announcement level so that data transmission does no such calculations per hop.

ORRP addresses the issue of deviation correction by a *multiplier angle method (MAM)*. Each RREQ and ORRP announcement packet has an additional field in the packet header: deviation multiplier. For simplicity, we assume that all nodes have

equal number of transceivers each separated with equal distances. The deviation multiplier is used to calculate the deviation angle from the desired angle at which a packet was sent. Table 3.1 defines a few key parameters which are illustrated in Figure 3.6.

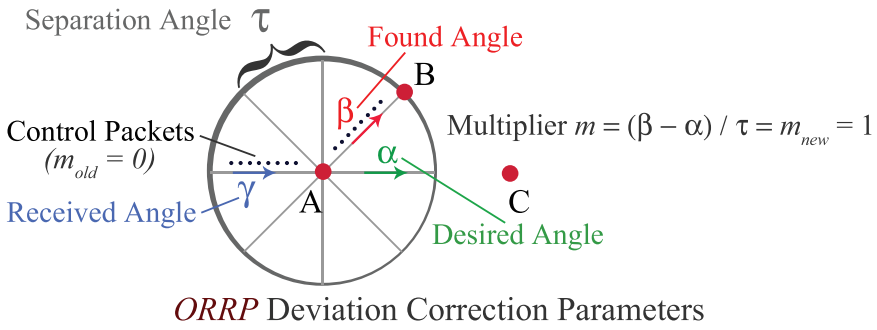


Figure 3.6: ORRP Multiplier Angle Method Parameter Illustration

Table 3.1: Multiplier Angle Method Definitions

Received Angle (γ)	The angle node received packets from.
Deviation Angle (θ)	The angle to add/subtract in order to correct the deviation.
Desired Angle (α)	The desired angle to send out.
Found Angle (β)	The angle of transceiver found cloest to desired angle with neighbor nodes.
Separation Angle (τ)	The angle of separation between each transceiver.
Multiplier (m)	The value to multiply τ by to find new desired angle.

When searching for a next-hop within the corresponding antenna/transceiver beam width, ORRP cycles through all its neighbors and finds one which requires an antenna-deviation angle yet is still confined to less than $\pm 45°$ (if packet is at originator) or $\pm 90°$ (if packet is merely a forwarder) of the original direction. If a packet is at the originator, only $\pm 45°$ needs to be searched because each of the four

orthogonal directions is sending. So, giving each direction a 90° coverage effectively covers all directions. In the forwarding case, however, because only one direction is considered with potentially "void" spots, a greater angle range is given to traverse "voids" yet ensure packets are not forwarded directly the opposite direction. If no neighbor is found satisfying these conditions, the packet is dropped and an error is flagged. The following equations are used to calculate angle to send and what state to store in each packet (all angle values are between 0° and 360°):

$$Dev\ Angle\ \theta = min(+\frac{\pi}{2},\ 2 * (\tau * m)), m\ positive \tag{3.1}$$

$$Dev\ Angle\ \theta = max(-\frac{\pi}{2},\ 2 * (\tau * m)), m\ negative \tag{3.2}$$

$$Desired\ Angle\ \alpha = \gamma + \pi - \theta \tag{3.3}$$

$$Multiplier\ m = \frac{(\beta - \alpha)}{\tau} \tag{3.4}$$

At each hop, the node unpacks the multiplier from the packet header and calculates a desired angle to send out based on (3.3). It then searches through its neighbors which have corresponding transceiver angles and finds one with the closest angle to the desired angle. When one is found, a new multiplier is calculated based on (3.4) and stored into the forwarding packet header before the packet is sent out. The process is repeated until the packet arrives at the destination. Algorithm 1 breaks down the process step-by-step.

Algorithm 1 Multiplier Angle Method

1: Unpack old multiplier m
2: Calculate angle needed to correct deviation θ (From eqs 3.1 and 3.2)
3: Calculate desired angle α (Eq 3.3)
4: Find interface with direction closest to α that has a neighbor (found angle β)
5: Calculate new multiplier m (Eq 3.4)

An example of our proposed *multiplier angle method* for deviation correction is shown in Figure 3.7. Node S is sending packets along the line. Because it has no nodes along the line in range of its transceivers, S opts to send to node A which is at a transceiver angle of τ from the desired angle α and encode multiplier m of one

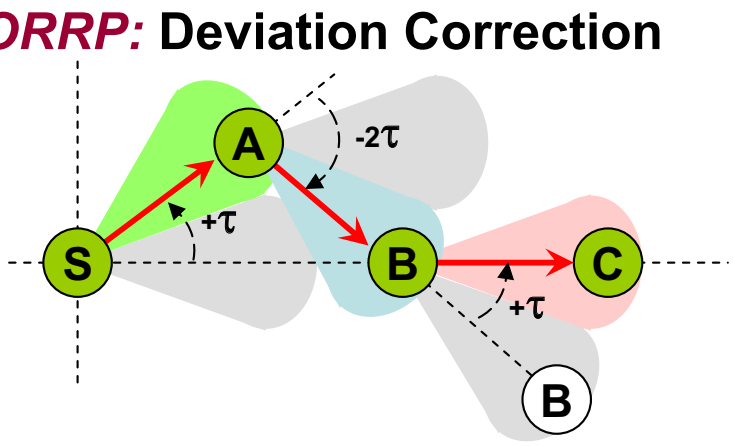


Figure 3.7: Basic Deviation Correction Example with Multiplier Angle Method

into the packet header. When node A receives S's packet, it calculates the desired outgoing interface based on (3.3) and as a result, sends to Node B while encoding a multiplier m of zero because there is no deviation from desired angle and found angle. The rest is self explanatory.

Potential problems may arise if the problem is cascading: Suppose node A wishes to send in the correct direction but has no neighbors in that direction. So, we continue with the original method of choosing a neighbor closest to the deviation angle and sending it. However, ORRP still maintains the multiplier angle method and corrects large deviations with larger forwarding angles. In dense networks, there should be no issues obtaining proper nodes to forward in a straight line.

3.2.3 MAC Layer Feedback

It is often helpful to provide MAC layer feedback to routing protocols to help in choosing better next-hops to forward packets. The MAC layer tells the network layer routing protocols that the next hop is unreachable. This often occurs when:

- A node wants to resolve a destination hardware address (via ARP) but the maximum number of retries is exceeded. This usually means that either the neighbor is no longer active, has moved out of range of the transceiver (possibly to an area covered by a neighboring transceiver).

- When RTS packets are sent but no corresponding CTS packets are received and the maximum number of retries is exceeded
- When a data packet is transmitted but never acknowledged at the link layer (with 802.11 MAC) and the maximum number of retries is exceeded.

ORRP resolves such issues in different ways depending on the type of packet. If a callback is issued from the MAC layer for a data packet, ORRP attempts to see if the corresponding next hop has simply switched interfaces (but still remains "in range") and if so, modifies the transmission interface. If the next hop is no longer within transmission range of any of the interfaces, it is dropped. Future alternatives can be to issue a new route request. The same technique is utilized for RREP packets. The reason why the specified next-hop is so important is because of rigid routes formed using traditional routing tables. Chapter 4 removes the rigid routes requirement by introducing *weak-state* information that decays over time rather than relies on a hard timeout.

For RREQ and announcement packets that issue a MAC layer callback, ORRP simply finds another neighbor associated with a specific direction and sends out to that neighbor. If no neighbor is found, MAM is again employed.

3.2.4 Discussion

In this subsection, we will see how ORRP deals with sparse networks and corner routing in addition to examining protocol implications, potential issues, and future considerations.

3.2.4.1 Sparse Networks

Although the concept of ORRP centers around sending packets in four orthogonal directions, it easily adapts to sparse network cases as ORRP merely seeks for rendezvous points between source and destination probe packets. ORRP works based on the assumption that source's and destination's "probe packets" will eventually intersect at a point. That intersection point, however, need not necessarily be along the orthogonal paths. If in the process of sending out RREQ packets, a path is navigated in a curve-like fashion (as opposed to a straight line) due to lack

of nodes, which intersects with a node that knows the path to the destination, then a path from source to rendezvous node to destination can easily be built.

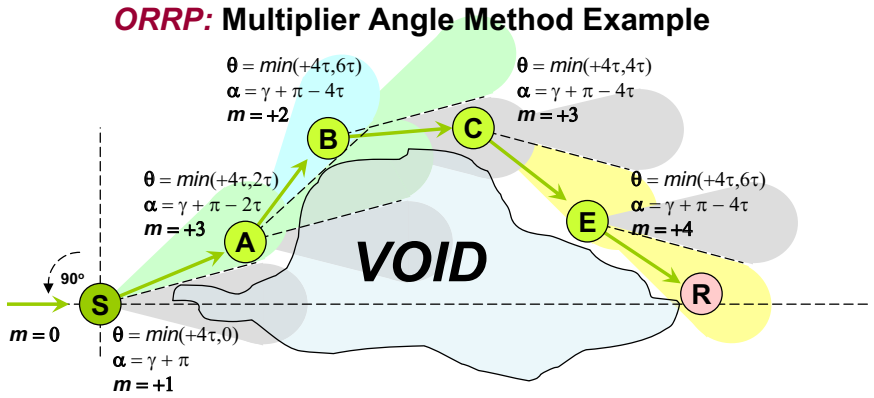


Figure 3.8: Multiplier angle method to traverse voids in sparse networks while maintaining direction

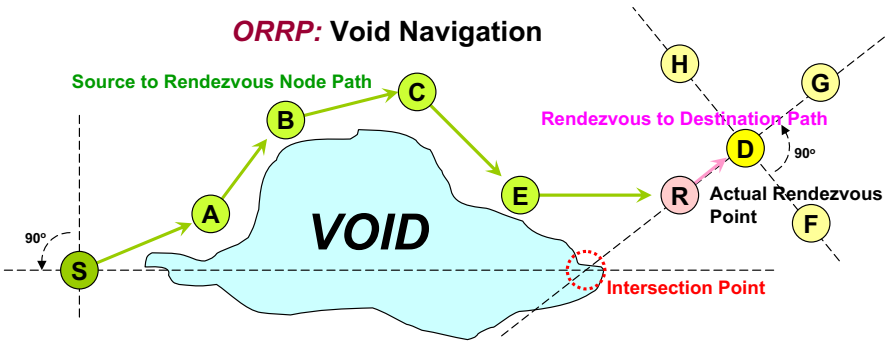


Figure 3.9: Traversing voids in sparse networks with differing intersection points

Figure 3.8 illustrates using ORRP's *multiplier angle method* of deviation calculation to navigate around an area devoid of nodes (only one direction is shown). Assuming that node R contains a path to S's intended destination, S's RREQ packets can traverse the perimeter of the void until it reaches node R. Calculations for each step of the way are shown and derived according to (3.1)-(3.4). Figure 3.9 shows a complete path selection from source to destination given a sparse network and no nodes at intersection points.

The multiplier angle method (MAM) differs from GPSR's perimeter routing and many other face routing techniques in several ways. Firstly, because ORRP

seeks only intersections with rendezvous nodes that contain a path to the destination, it is not trying to reach a specific node (assuming that rendezvous nodes will successfully deliver to destination). This allows for much higher flexibility and less stringent requirements for path selection. Secondly, MAM is an inherent nature of ORRP and *not* a special case that switches on and off like GPSR's perimeter routing. Additionally, GPSR's packets maintain additional states such as the node it entered the perimeter routing, points on the coordinate space, and destination information whereas ORRP's MAM requires only one state updated at each node resulting in reduction in overall space. MAM, therefore, offers a much more unstructured and lighter alternative to GPSR's perimeter routing.

3.2.4.2 Perimeter Nodes

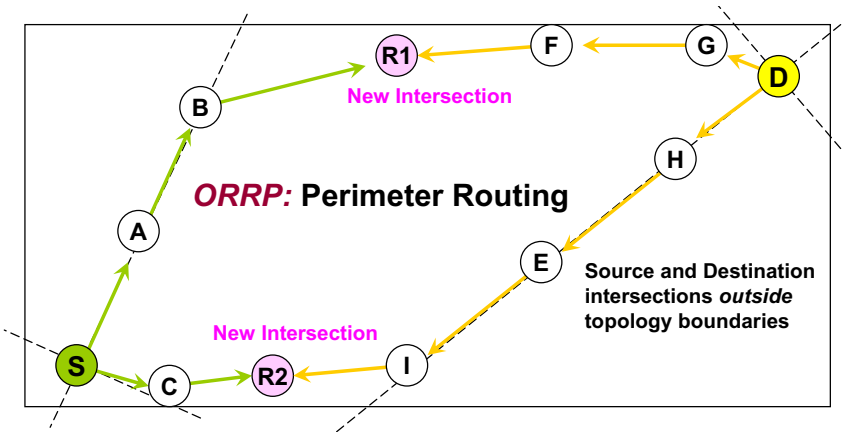


Figure 3.10: Forwarding along perimeter is using MAM deals with corner cases where node intersections are outside of topology boundary. Appropriate TTL for ORRP announcement and RREQ packets must be set to minimize excessive state

Our analysis in section 3.3 shows that "corner nodes" have a much higher probability of having no intersection points within the network topology with purely straight line paths. The multiplier angle method allows for state information to be propagated along the network perimeter as long as its send angle is within $\pm\frac{\pi}{2}$ of the desired direction. Figure 3.10 shows the problem as well as how MAM mitigates the issue. While this prevents packets from traversing back on itself, it is important to set a TTL on ORRP control packets to ensure that perimeter nodes do not get

saturated with state information. Section 5.2 describes simulation results on TTL's effect on reachability, path length, and state maintenance.

3.2.4.3 MAC Layer Issues

R. R. Choudhury et al. [45] bring up several concerns with the nature of directional antennas' asymmetric gain resulting in collisions and hidden terminal problems. The main result shows that straight line routes are inefficient because of higher interference in the direction of ongoing communications. M. Sekido et al. [46] propose several MAC level solutions to the problem without taking obscure paths to avoid hidden terminal problems and because ORRP focuses more on the routing layer, we do not feel these MAC layer issues are a problem.

3.2.4.4 Load Balancing

It has been shown that network congestion can be controlled and limited by routing packets using two-phase routing algorithms [50] [49]. Current wireless networks measure route cost through hop count. In high-traffic networks, by choosing the shortest path, nodes with many connections will become saturated with packets. Busch et al. [50] has shown that by drawing a perpendicular bisector between source and destination and forwarding packets from source to a random point on the perpendicular bisector which in-turn forwards to destination when that point is reached, load can be balanced across the network. In much the same way, ORRP inherently implements a seemingly two-phase routing algorithm because it provides rendezvous abstractions whereby the source sends to the rendezvous node and the rendezvous node sends to the destination. As a result, there is potential for studies in unstructured load-balancing techniques with ORRP. These are beyond the scope of this thesis.

3.2.4.5 Three-Dimensional ORRP

While we have been focused solely on two-dimensional topologies, it is easy to see how ORRP can be expanded into real-life applications in three-dimensions. In order for this to occur, we assume that while the node is three dimensional with transceivers tessellated in a spherical fashion, there exists a cross-section that is

parallel to the ground whereby transceivers are still tesselated in a circular fashion covering an omnidirectional spread when taken together. Additional antennas directed in the vertical (height) directions are mathematically projected onto this surface and essentially "combined" with the respective antenna on the cross-section its projection falls on. Routing tables are built and MAM is calculated solely based on this 2-D cross-section. Although we do not examine the implications of this setup, we feel it should not have too many differences when compared to the 2-D setup.

3.3 Protocol Analysis

As mentioned in the introduction, ORRP provides connectivity with less information at the cost of suboptimal path selection. In this section, we will examine metrics of *reachability* and *average state complexity* with network growth under a set of conditions and topologies while also observing *path stretch* to determine how much inefficiency in path selection we are trading off to utilize ORRP. Note that for all numerical analysis, our model does *not* consider details such as angle deviation correction and whether a rendezvous node at the particular point exists. Specifically, we will attempt to characterize bounds on how varying topologies affect reachability, state complexity, and path stretch in the base case. In short, we are simply *drawing lines* and finding intersections without examining things like actual nodes being at intersection points or angle correction. This explains the discrepancy in actual simulation data vs. our characterization of reachability and path stretch.

3.3.1 Reachability Upper Bound Analysis

For our numerical analysis, given a Euclidean area over which nodes are scattered, a source-destination is said to be unreachable if all rendezvous points are outside the boundaries of the topology area. In order to determine the reachability upper bound in this case, it is important to isolate cases where ORRP will fail based on source and destination location and orientation. Assuming a Euclidean 2-D rectangular topology $0 < y < b$ and $0 < x < a$ with nodes randomly oriented with "north" between 0° and 90°, we claim that an upper bound in packet delivery

success utilizing ORRP is 99.4% in a perfect square topology without consideration of nodes and angle correction. In this analysis, we simply draw lines.

The general idea behind obtaining the reachability upper bound is to find intersections between orthogonal lines between the source and destination. In cases where all the intersections lie outside of the rectangular area for a particular source and destination oriented in a certain way, ORRP fails to find a path. Notice that this analysis assumes that ORRP probe packets do not travel along perimeters of the Euclidean area under consideration and therefore inspects a worst-case upper bound on reachability. In actual simulation implementation, we use very simple techniques (see Sections 3.2.2.3 and 3.2.4.2) to achieve 100% reachability in ORRP.

Our analysis begins with randomly selecting two source and destination pairs along with random orientations. We then formulate the equations of the orthogonal lines generated by these two nodes and randomly selected orientations and find their intersection points. If at least one of these intersection points lies in the boundaries of the topology, then we consider that particular source-destination pair as reachable. By iterating through all possible orientations for each possible source-destination pair, we find a percentage of the total combinations that provide reachability vs. the total paths chosen. Because different Euclidean-area shapes will no doubt yield different reachability requirements, we calculated the reachability probability for various area shapes by using Matlab. We refer the reader to Appendix A for a detailed description of our reachability analysis.

Figure 3.11 shows the varying degree of reachability depending on the topology shape. As can be seen, topologies that spread nodes in single direction such as a rectangle or ellipse with one of the sides much greater than the other yield poor results for reachability due to the fact that ORRP intersections often fall outside of the topology area more easily under those situations. While at first this seems rather disappointing, it is important to note that random topologies rarely fall into a rectangle with one side much longer than the other and even so, ORRP's MAM enables rough forwarding along perimeters to find intersection points, significantly enhancing reach.

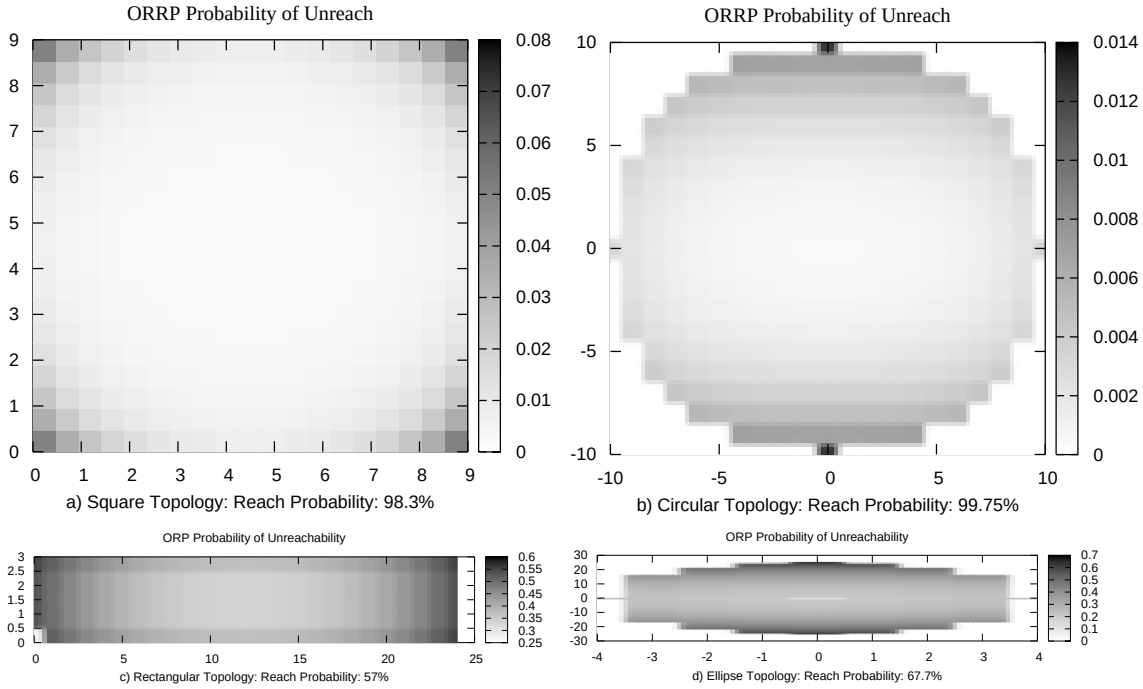


Figure 3.11: ORRP Reachability for Various Topology Areas: Nodes in darker regions are less reachable. The strength of the darkness of a point shows the probability that a node located on that point will be unreachable by any other node on the area. It can be seen that topology corners and edges suffer from the highest probability of unreach.

3.3.2 State Information Maintained at Each Node

One of the major hindrances to network scalability is the amount of state information each node is required to maintain. In completely proactive routing protocols, nodes trade routing tables and other information on a regular basis to keep routes up to date. While this helps maintain connectivity even in highly mobile environments, maintaining such a vast amount of state information at each node requires extensive coordination and information transfer resulting in networks that scale poorly. Because ORRP only forwards routing announcements in orthogonal directions and only nodes along those lines maintain state information about the node sending announcements, it is expected that ORRP will incur less overhead in state maintenance. We ran Matlab simulations for a square topology of nodes and calculated the total amount of state information each node maintained with respect to the total number of nodes in the system. Because the granularity in our

Table 3.2: Comparison of Average State Information

	GPSR	DSDV	XYLS	ORRP
Node State	O(1)	O(n^2)	O($n^{3/2}$)	O($n^{3/2}$)
Reachability	High	High	100%	High (99%)
Name Res.	O(n log n)	O(1)	O(1)	O(1)
Invariants	Geography	None	Global Comp.	Local Comp.

simulation was one, we were able to calculate the total amount of state information maintained by iterating through each possible node and orientation combination and taking the average of the distance of the orthogonal lines to the borders of the topologies. This was used to calculate average total state maintained at each node. Our results showed that with rectangular and circular topologies, state scales on the order of $N^{3/2}$ with N being the number of nodes.

Table 3.2 shows the ORRP's state information maintenance compared to other protocols. Compared to GPSR with location mapping factored in, ORRP requires more state information to be maintained at each node but requires much less structure and global information to be shared. Looking at the opposite extreme, DSDV provides full connectivity and optimal path selection at the cost of a scalability. In comparison to XYLS [33], ORRP requires less information (Local compass vs. global compass) while achieving virtually similar reach.

3.3.3 Average Path Stretch

Because ORRP trades off optimal paths for connectivity under less information, it is important to see what conditions lead to unacceptable path choices and how much sub-optimality we are trading off for connectivity in an unstructured manner. We begin first by attempting to analyze and understand what kind of stretch values we should expect and then move onto Matlab and NS2 [28] simulations for more realistic values.

Suppose two nodes are trying to communicate with each other using ORRP as shown in figure 3.12 where d is the path length between the two points and a and b are the lengths of the two piece ORRP Path (souce-to-rendezvous node and rendezvous node-to-destination). Because there can theoretically be two interception points

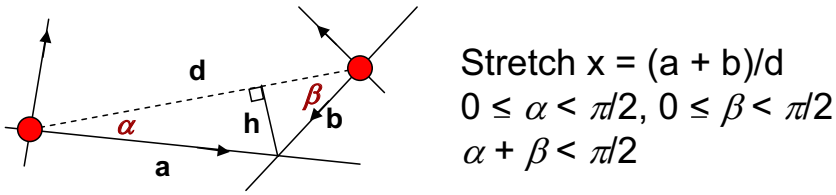


Figure 3.12: Calculating average path stretch (ORRP Path/Shortest Path) between two nodes

between the pair of orthogonal lines emanating from the two nodes, path selection is based on the shorter of the two paths. The conditions listed in figure 3.12 bound the selection to the minimum ORRP Path. Stretch is defined as the ratio between the path selected (in this case, $a + b$) and the shortest path (d). Due to the nature of orthogonal lines, α and β are between 0 and $\pi/2$ and because there is an equal probability for each node to be oriented in a certain manner, α and β are uniformly distributed.

$$h = b\sin\beta = a\sin\alpha \tag{3.5}$$

$$d = b\cos\beta + a\cos\alpha \tag{3.6}$$

$$x = \frac{a+b}{d} = \frac{\sin\alpha + \sin\beta}{\sin(\alpha+\beta)} \tag{3.7}$$

Equations 3.5 and 3.6 come from basic trigonometry. Equation 3.7 represents the stretch x in terms of two uniformly distributed angles α and β. We know that the *probability density function* (PDF) of a random variable that is uniformly distributed is merely the inverse of the interval. The result is the *PDF* of α and β to be $\frac{1}{\frac{\pi}{2}}$ and $\frac{1}{\frac{\pi}{2}-\alpha}$ respectively, to satisfy the conditions listed in figure 3.12. The minimum stretch possible is merely the shortest path and therefore, one. The maximum stretch occurs when both α and β are at $\pi/4$ and $x = \sqrt{2} \approx 1.414$. As a result we expect the mean of the stretch to be somewhere between 1 and 1.414.

$$E[X] = \int_0^{\frac{\pi}{2}} \int_0^{\frac{\pi}{2}-\alpha} \frac{\sin\alpha + \sin\beta}{\sin(\alpha+\beta)} \left(\frac{1}{\frac{\pi}{2}-\alpha}\right)\left(\frac{1}{\frac{\pi}{2}}\right) d\beta d\alpha \tag{3.8}$$

$$E[X] = 1.125$$

Equation 3.8 gives the expected value of the random variable X with respect to the two uniformly distributed angles α and β. Integrating the values over the chosen intervals yields a mean of 1.125 for the ORRP path stretch in *unbounded regions* (12.5% path stretch). Although not quite exactly shortest path, we can see that the stretch is still very low and in most cases, acceptable. Similar analysis leads to a variance of 0.0106 and therefore we can expect most of the path selections to be relatively close to shortest path.

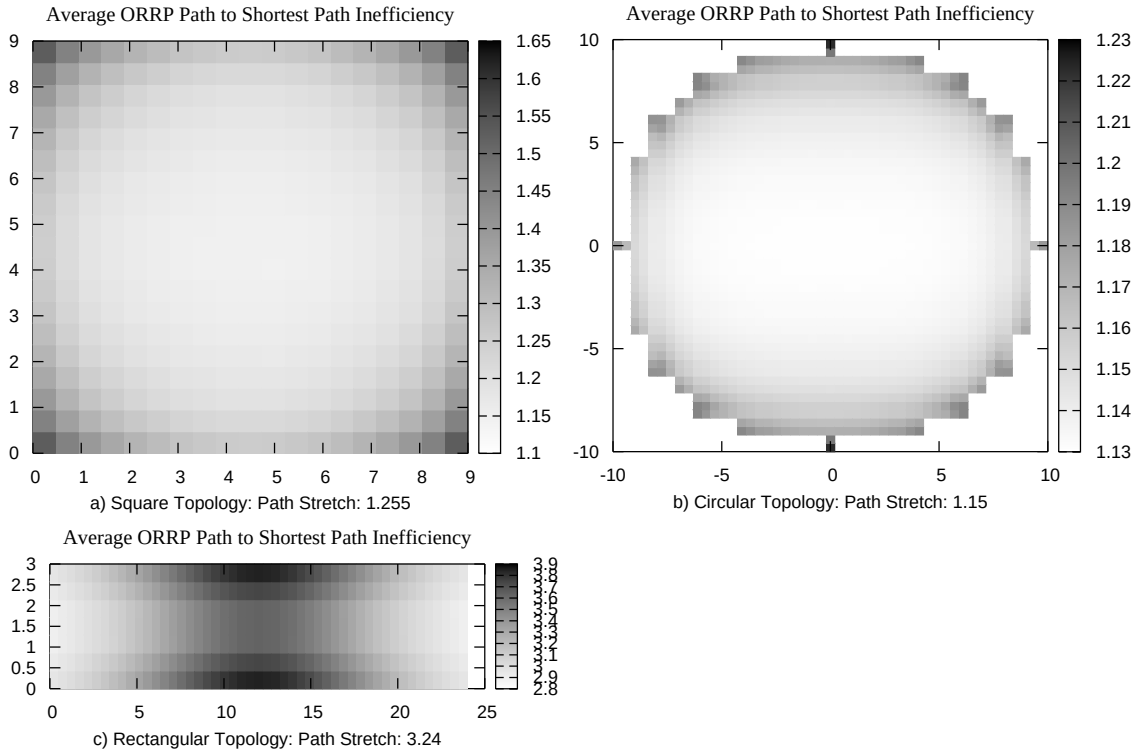


Figure 3.13: ORRP Path vs. Shortest Path Ratio: A node in darker regions have higher likelihood of having longer paths to a destination on the area. Topology corners and edges suffer from the higher stretch in symmetric topologies.

Using Matlab, we created several grid topologies and iterated through every

possible source-destination pair along with every possible orientation for each node. We then built paths (distances) from the source to rendezvous node to destination and compared with the shortest path. If no rendezvous nodes were found within the boundaries of the topology, a path length of the perimeter of the topology was used in calculations, as this is the worst possible path length if packets are routed along perimeter. Figure 3.14 gives the distribution of average stretch values for a square topology. As shown, the stretch values are confined between 1 and 1.414 and lean toward 1 as suggested by our calculated mean and variance.

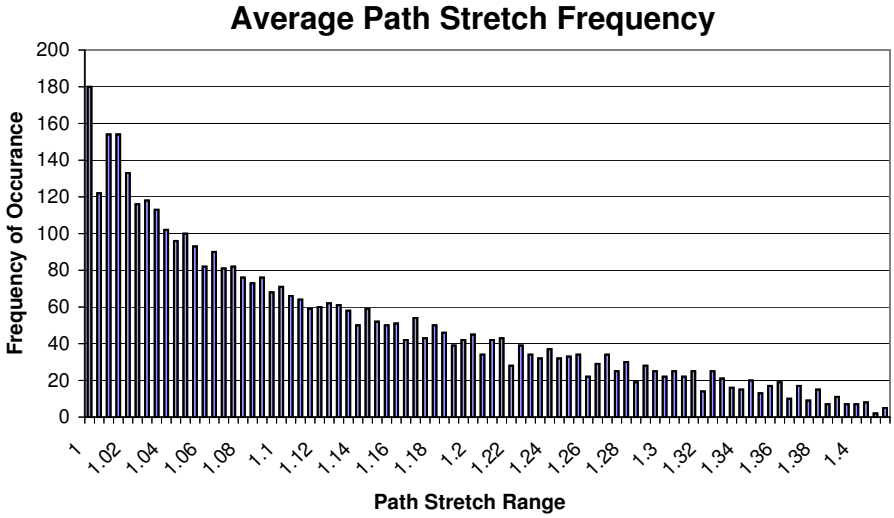


Figure 3.14: Average stretch (Frequency distribution of ORRP path stretch in square topology network). The stretch values are conned between 1 and 1.414 and lean toward 1 as suggested by our calculated mean and variance.

Figure 3.13 shows evaluated topologies along with ORRP path to shortest path ratios for nodes in each region. As expected, the rectangular topology yielded the highest path discrepancy with an average path stretch of 3.24. This is most likely due to the fact that in the reachability evaluations as shown in Figure 3.11, the rectangular shape had the highest amount of unreachability resulting in the perimeter case needing to be invoked the most. The highest path discrepancy appeared in the middle of the rectangle due to the fact that nodes in the middle allow for the longest ORRP paths, reaching the left and right edges while the shortest path is extremely short (the middle to anywhere else directly is short). The results from the other

topologies are also consistent with expectations in that the circular topology, with the greatest reach probability, yielded the smallest average path stretch.

3.3.4 Additional Lines Study

While our study focuses using a pair of orthogonal lines (one at the source and one at the destination) to build routing paths, it is interesting to see the effect of adding additional forwarding directions into the scheme. Specifically, we wish to see how the addition of lines affects reach probability, path stretch, and states maintained in the network. Our analysis was performed in Matlab with a grid network under varying topological boundaries without employing any deviation correction.

Like in [1], our analysis begins with randomly selecting two source and destination pairs along with random orientations. We then formulate the equations of the lines generated by these two nodes and randomly selected orientations and find their intersection points. The equations of the lines will be different depending on whether we are looking at 1, 2, or 3 lines. If at least one of these intersection points lies in the boundaries of the topology, then we consider that particular source-destination pair as reachable. By iterating through all possible orientations for each possible source-destination pairs, we find a percentage of the total combinations that provide reachability vs. the total paths chosen. Because different Euclidean area shapes will no doubt yield different reachability requirements, we calculated the reachability probability for various area shapes by using Matlab in a grid network. Table 3.3 shows the reach probability vs. the number of lines used for calculations.

Table 3.3: ORRP Comparison of Reach Probability vs. Number of Lines

	1 Line (180°)	2 Lines (90°)	3 Lines (60°)
Circle (Radius 10m)	58.33%	99.75%	100%
Square (10m×10m)	56.51%	98.30%	99.99%
Rectangle (25m×4m)	34.55%	57%	67.61%

It can be seen that the addition of more lines yields significant gains from the one to two line case but only slight gain afterwards. Particular interest is given to the rectangular case where even with three lines, the raw reach probability is

very low. We suspect the reason for this is the slim shape yielding to much more path intersections outside of the topology area. [1] showed that most of the unreach happens at the topology perimeters and even with additional lines, these perimeter nodes need a very high degree of angular match between lines before a path can be made. The result is that by adding only 30° more to match on, the angle of incidence is still too high to find an intersection within the area.

A similar analysis is done to find path stretch. If a source and destination pair has a line intersection within the topology boundaries, the shortest total distance (from source to intersection point and intersection point to destination) is selected as the path. This distance is divided by the distance between the source and destination to obtain a path stretch. In cases where there is no intersection inside the topology boundaries, we simply add the distance of the perimeter as that is the maximum path we can obtain with MAM. Table 3.4 gives the Matlab calculated path stretch for 1, 2, and 3 lines.

Table 3.4: ORRP Comparison of Path Stretch vs. Number of Lines

	1 Line (180°)	2 Lines (90°)	3 Lines (60°)
Circle (Radius 10m)	3.854	1.15	1.031
Square (10mx×10m)	4.004	1.255	1.039
Rectangle (25mx×4m)	4.73	3.24	1.906
Grid (No bounds)	1.323	1.123	1.050

Table 3.3 and 3.4 show the reachability and path stretch numerical analysis results for 1-3 lines all equidistantly separated from each other. While for reach probability, the effect from one to two lines is dramatic, it can be seen that very little gain is achieved by adding additional lines. In the case of path stretch, however, the addition of additional directions to send announcement and RREQ packets result in much better path selection as more packet interceptions occur. We suspect that in sparser networks or networks with voids, the gains would be negligible as control packets would take similar paths with MAM. It is important to note that with MAM, almost all the corner case reach issues can be resolved with only 2 lines.

Figure 3.15 demonstrates the potential increase in state maintenance needed

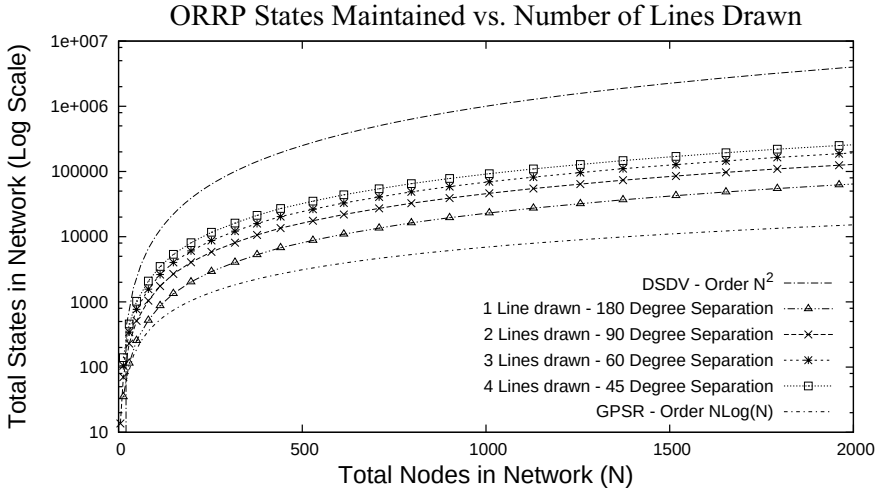


Figure 3.15: Total states maintained in network with respect to the number of transmission lines used. As number of lines increase, the number of states maintained throughout network increases.

with the addition of transmission lines. While increasing steadily, it is still much less than order N^2.

3.4 Performance Evaluation

In this section, we provide performance evaluations of ORRP under various parameters and against several proactive and reactive routing protocols with omni-directional interfaces. The simulations were performed using Network Simulator [28], with nodes using the standard IEEE 802.11 MAC and the antenna range set to 250m (NS2 default). RTS/CTS is turned off because this is standard practice in actual deployment. Unless otherwise specified, each node is randomly positioned in a square 1300m x× 1300m area.

All simulations were averaged over 2 runs of 5 different randomly generated flat topologies (total 10 trials) and the 95% confidence intervals of the runs plotted. ORRP nodes were outfitted with n (divisible by 4) interfaces with each interface having a beam-width of $360/n$ degrees and for each run, each node randomly chooses an interface as the *local north*. ORRP Hello packets are sent out every second and Announcement packets every 4 seconds. When comparing against *reactive* routing

protocols like DSR and AODV which require no periodic updates, the standard NS2 defaults were used. For OLSR, the hello interval is set to one second and the topology control update interval is set to 4 seconds to match ORRP announcement intervals. For all simulations, MAC layer feedback is employed for all the routing protocols. ORRP MAC layer feedback implementation is described in section 3.2.3. The implementation and defaults for OLSR can be found at [138]. Table 3.5 lists the default simulation parameters used and the actual traffic pattern used in each scenario is detailed in each individual section.

Table 3.5: ORRP Default Simulation Parameter

Parameter	Values
Transmission Radius	250.0m (NS2 Default)
Number of Interfaces	12 Directional Interfaces (for ORRP)
TTL for RREQ/Announcement Packets	10
Topology Boundaries	1300m x× 1300m - No Mobility
Queue Length	250
Announcement Interval	4.0s
Route Timeout	5.0s
Hello Interval	1.0s
Simulation Time	70s
CBR Packet Size / Send Rate	512 bytes / 2Kbps

During the performance evaluations for ORRP, several topologies were used. For each category of topologies, 5 random topologies were generated and the average neighbors and the 95% confidence interval of average neighbors recorded. Voids generation in the topologies are explained in greater detail in section 3.4.1. Table 3.6 lists the information for each topology class generated.

The performance evaluations for ORRP are broken up into two major sections: *standalone evaluations* and *comparison evaluations*. Standalone evaluations deal purely with ORRP and adjust several knobs to understand how each plays a part in the protocol. Comparison evaluations take ORRP and evaluate it against several proactive and reactive routing protocols. In the following subsections, we will provide metrics and evaluation conditions for both the standalone and comparative evaluations as well as the results from our simulations with respect to each

Table 3.6: Topologies Used in ORRP Simulations

Number of Nodes	$1300 \times 1300m^2$	$1300 \times 1300m^2$ (With 2 Voids)	$2500 \times 400m^2$
50 Nodes	4.8 ± 1.25	5.1 ± 1.22	6.8 ± 2.06
100 Nodes	9.1 ± 0.79	10.6 ± 0.91	13.3 ± 1.10
150 Nodes	14.4 ± 0.99	16.1 ± 2.94	20.7 ± 2.15
200 Nodes	19.9 ± 2.62	N/A	N/A
250 Nodes	24.5 ± 2.63	N/A	N/A
200 Nodes	29.4 ± 1.33	N/A	N/A

condition.

3.4.1 Standalone Performance Evaluations

Standalone evaluations deal solely with ORRP under various conditions. In this section, we outline the metrics and the purpose and specifics of each conditions evaluated. The metrics used for the *standalone evaluations* are as follows:

- **Reach Probability** - Because ORRP relies on the intersection of announcement and route request paths, there is a possibility that no path will be found (See section 3.3.1). We evaluate reach probability by sending only a few (around 1 or 2) packets from all nodes to all nodes ensuring little to no congestion drops and collecting the number of CBR packets received as compared to the number sent. It is important to understand that this metric is different than *data delivery success* which is utilized and described in the *comparison evaluations*. We will show that ORRP has fairly high ($\sim$98+%) reach even in sparse networks with voids.

- **Total States Maintained Network-wide** - It is important to understand how the number of states network-wide grows under varying conditions to understand what kind of overhead will be required to maintain these states. Because the network is fixed and we do not simulate changes in link quality, it is assumed that the states each node will maintain will not change with time. We therefore measure this by summing the number of entries in each

node's routing table at the end of the simulation as the total states maintained network-wide. While this value fluctuates with the number of connections, we assume that for the most part, connection information is temporary and announcement information is more permanent (it gets retransmitted periodically) and therefore takes priority. We will show that ORRP maintains and grows at roughly order $N^{3/2}$ states network-wide.

- **State Distribution** - Structured and hierarchical routing protocols leave some nodes with more state information than others. This is problematic because those nodes are accessed more frequently (utilizing added power or causing bottlenecks) and results in multiple single points of failure. We evaluate the state distribution by creating a running average of the states each node maintains over the course of the simulation and graph the result on a 3D topology graph where nodes with more states will have higher "peaks". We will show that ORRP distributes state fairly evenly causing no single point of failure.

- **Average End-to-End Path Stretch** - Path stretch is defined as the path taken over the actual shortest path (as computed by Djikstra's algorithm). In NS2, each CBR packet keeps track of the shortest path from the source to each node along the path and outputs this to a trace file. We therefore measure path stretch by taking the actual number of hops traversed by each CBR packet and dividing it by the calculated shortest path. We will show that ORRP path stretch is not bad ($\sim$1.2-1.5) under most conditions. It is important to note that with denser topologies, the path stretch is expected to be larger because shortest paths are calculated by absolute distances greedily (i.e. it will always choose the next hop that is closest to the destination but still within transmission range) while actual transmission can choose any node within transmission range (even if the next hop node is physically very close).

- **Average End-to-End Latency** - Latency is the amount of time for a packet to travel from source to destination. This time includes the time for request and reply exchanges as data packets are buffered during this time. At times,

multiple hops over better links are faster than long-haul hops. We evaluate latency by taking the difference between the received time and the send time of each CBR packet. We will show that ORRP has fairly low latency even while not traversing on the shortest path.

To evaluate the metrics listed above for our standalone analysis, we utilized a simple *traffic pattern*: First, we allowed ORRP to perform its messaging for about 10 seconds to ensure states are properly seeded network-wide. Then, we send 512 byte CBR packets from all nodes to all nodes at a rate of 2Kbps for 2 seconds with thee start times of each of these connections varied between 10 seconds after the start of the simulations and 60 seconds. Since the simulations went on only for 70 seconds, it provided a good 10 second buffer for rogue packets to either be dropped due to TTL or reach the destination. We limited the amount of data sent to ensure no packet is dropped due to medium saturation or excessive collisions. Table 3.7 lists the default traffic pattern used in our standalone simulations of ORRP.

Table 3.7: ORRP Standalone Sims - Default Traffic Pattern Information

Parameter	Values
Number of CBR Connections	All-to-all
CBR Packet Size	512 Bytes
CBR Transmission Rate	2Kbps
CBR Transmission Duration	2.0 seconds
CBR Start Time Range	10.0 - 60.0 seconds randomly generated

In the standalone analysis, we evaluate each of the metrics above under varying conditions. These conditions and accompanying explanations of why they are important are listed below:

- **Varying Number of Interfaces** - ORRP randomly chooses a neighbor in an interface direction to send announcement and RREQ packets. When there are fewer interfaces, the *granularity* of each interface broadens to cover more neighbors. This leaves a greater risk of announcement and RREQ packets with potential intersections "missing" each other. This problem is complicated

more as fewer interfaces leave fewer options to apply angle correction and can potentially lead to nodes veering far of original intended path resulting in lower possibility of intersections. Additionally, with the smaller number of interfaces, the medium is used less efficiently and more nodes are affected with each transmission.

- **Network Voids** - Voids cause issues because line intersection points potentially lie within the void. ORRP's multiplier angle method (see section 3.2.2.3) was implemented to help nodes traverse around voids while maintaining a general forward direction as well as help route around perimeters. We wish to understand how ORRP fairs with respect to network voids. In our tests, we generate two voids by modifying the CMU scenario builder code coupled in the official NS2 distribution (specifically the setdest.cc file) to take in two inputs: *max* and *min void radius*. Upon scenario generation, two X-Y points are randomly chosen as the center of the void and two random void radiuses generated between the *max* and *min void radius* parameters specified. When nodes are placed in a network, positions are randomly generated. Our inserted code assures that the positions that are generated are not within the radius of each of these voids. In our void scenarios, we generate square topologies of $1300 \times 1300m^2$ with two randomly generated voids that have a radius between 200 and 300 meters.

- **Varying Announcement and RREQ TTLs** - It makes sense that with an increase in control packet TTL, more state is propagated network-wide resulting in a higher potential for rendezvous. On the flip side, the network is utilized unnecessarily. We seek to find a balance between over-sending control packets, thus flooding the network, and achieving high connectivity.

- **Varying Topologies** - Our analysis in section 3.3.1 showed that by simply drawing lines, the probability of intersect within a bounded region varies with different topologies. To route along the perimeter and attempt to maintain straight line paths in the presence of voids, we introduced MAM (see section 3.2.2.3). It is important, therefore, to test the effectiveness of MAM to

provide high reachability with varying network topologies.

- **Varying Number of Transmission Lines** - ORRP send packets along two orthogonal lines. Our analysis in section 3.3.4 shows that adding lines should increase reach and provide lower path stretch. We wish to understand the tradeoff between the increase in the number of states maintained vs. the gains with sending along additional lines.

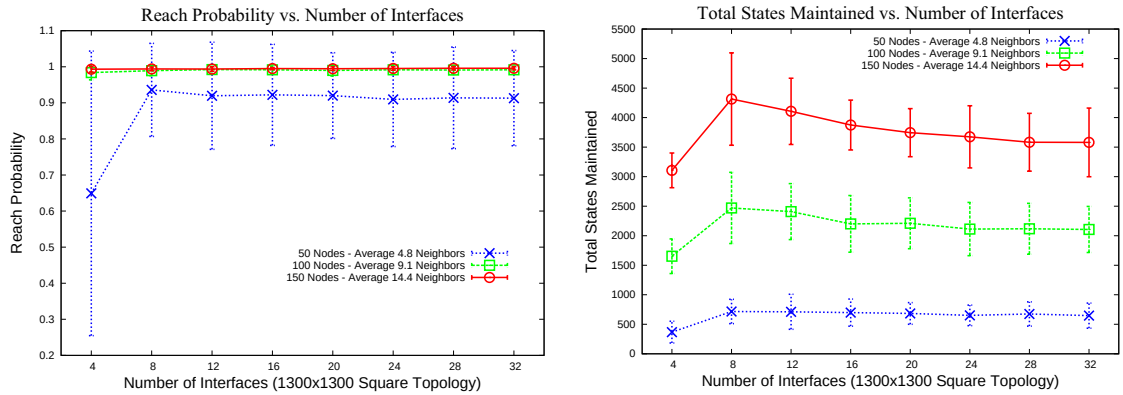
In the following subsections, we will present and discuss our results from the standalone performance evaluations.

3.4.1.1 Effect of Number of Interfaces on Varying Network Densities

One important consideration for nodes with multiple transceivers/antennas is to find a tradeoff between the number of interfaces vs. performance gains. In this section, we will examine the tradeoffs in *reachability*, *total state maintained network-wide*, *average end to end path stretch* and *average end to end latency* with varying number of interfaces per node. We ran the simulations two times on five different fixed topologies (10 total runs per set) and under 3 different set of node densities (sparse with an average of 4.8 neighbors, medium with an average of 9.1 neighbors, and dense with an average of 14.4 neighbors) and varied the transceiver orientations and *local norths* in each run.

We speculate that by increasing the number of interfaces and thus increasing the granularity of angle calculations, reachability should increase simply because there are fewer neighbors assigned to reach interface. This allows for tighter control on next hop (instead of randomly choosing a next hop), increasing the odds of an announcement-RREQ "hit". Furthermore, this tighter control on next hop should theoretically lead to better paths and lower end-to-end latency as well because straight lines are maintained more accurately. In each experiment, we suspect that the state remains fairly constant with increasing number of interface simply because announcement intervals remain fixed across each run.

Figure 3.16 shows that in dense networks, varying the number of interfaces had little to no effect on reachability as all nodes were reachable. As the network became sparser, however, we see a sharp increase from 4 to 8 interfaces. We suspect that



(a) (b)

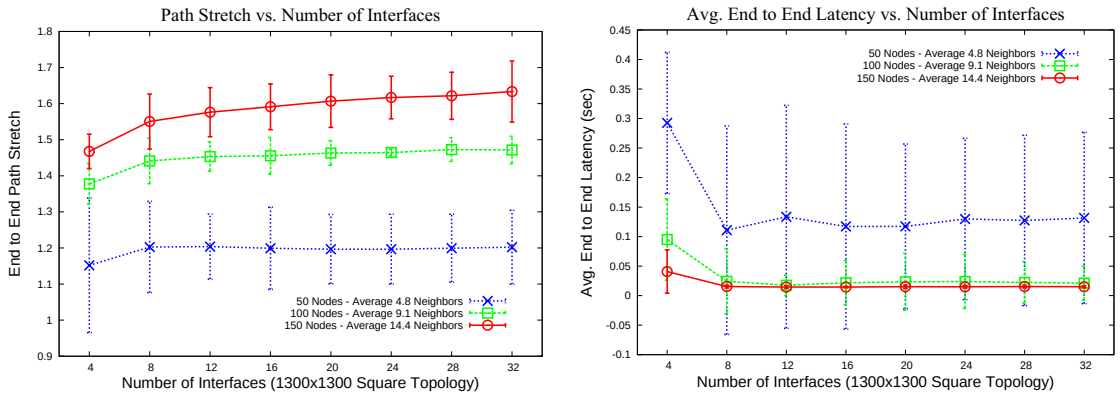
Figure 3.16: Effect of number of interfaces on ORRP reachability and total states maintained for dense, average, and sparse networks. In sparse networks, increasing number of interfaces provides significant benefits at first but diminishing returns after 8 interfaces. With the exception of going from 4 to 8 interfaces, total states maintained network-wide decreases up to a certain point with an increase in the number of interfaces.

one of the major reasons for the increase in reach probability is the sheer number of nodes each transmission "cone" encompass. With fewer interfaces, each transmission "cone" needs to reach a lot more nodes than finer grained interfaces. This could result in packets being delivered orthogonally, but not necessarily intersecting due to poor node choice by the sender. Also, because 4 interfaces is not enough to perform adequate angle correction (even "correcting" a path by shifting by 1 interface essentially forwards packets 90° from the intended direction), announcement states are not adequately being seeded and RREQ packets often find it hard to keep moving "forward". Up to a certain point, however, the granularity has less effect, especially in sparser networks.

Surprisingly, figure 3.16 also shows that there is a fairly large increase in total states maintained network-wide from 4 to 8 interfaces and continues to decrease with increasing number of interfaces. As with the reachability, we believe that the increase in states from 4 to 8 interfaces stems from a large change in ability to perform MAM angle correction. With only 4 interfaces, there is little to no

angle correction because again, even shifting transmission by 1 interface essentially forwards packets 90° from the intended direction.

The reason why there is a slight *decrease* in states from 8-16 interfaces (and it is much more noticeable with denser networks), is because in the announcement phase, each node randomly chooses a neighbor in a set antenna/interface direction to send to. In cases where there are more than 1 neighbors associated with a specific interface direction (such as in denser networks), announcement packets at 2 different intervals sending out the same direction might potentially be sent to 2 different neighbors. There is, therefore, an increase in state maintained simply because the neighbor to first receive the announcement will have an entry for the state until it expires and the neighbor to receive it later will have also have an entry for the state. The result is consistent as the decrease in number of states maintained network-wide happens only when the average number of neighbors per node is close to or more than the number of interfaces. The total state is also consistent with our initial Matlab analysis, which showed that ORRP state scaled on order $N^{3/2}$ (roughly 650 states for 50 nodes, 2100 states for 50 nodes, and 3600 states for 100 nodes).



(a) (b)

Figure 3.17: Effect of number of interfaces on ORRP end-to-end average path stretch and latency for dense, average, and sparse networks. There is a huge decrease in latency going from 4 to 8 interfaces as well as a steady increase in path stretch in dense networks with increasing number of interfaces. Path stretch increases with larger densities.

Figure 3.17 shows our data for average end to end path stretch and latency. With the increase in node density, path stretch increases as expected. Because ORRP has no notion of neighbor distances, it arbitrarily chooses a neighbor in the interface direction it wishes to send. At times, this neighbor could be one that is closer to the destination geographically or sometimes it could be farther. Therefore, it makes sense that with a denser network (more neighbor choices), the average path stretch will be higher (nodes might choose neighbors that are closer to itself and require more hops to destination).

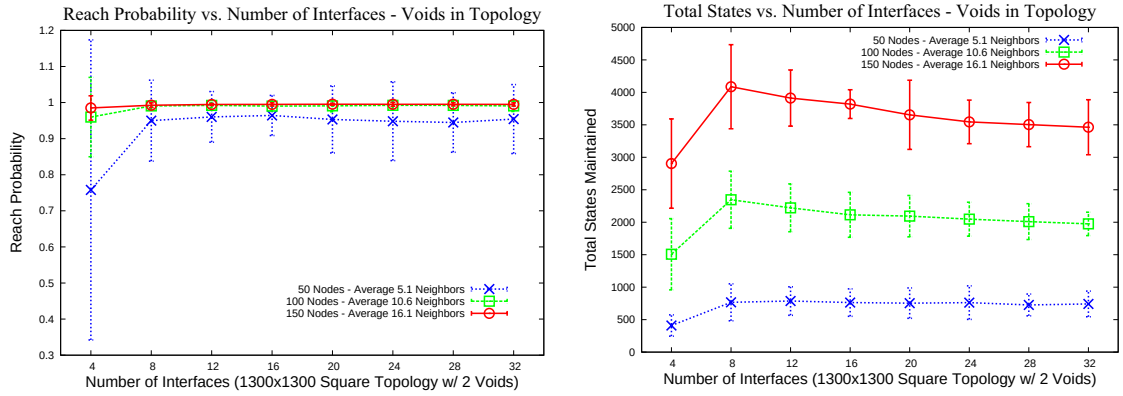
Although it was expected that with an increase in number of interfaces, denser networks will no doubt decrease in path stretch due to finer granularity in selecting a next-hop neighbor to send, we were surprised to find that this was not the case. With increased network density, increasing the number of interfaces actually led to a slight increase in end-to-end path stretch. To reconcile this issue, we defer back to our explanation of the number of states maintained network-wide. With the fewer number of states maintained network-wide due to lessened "randomness" in choosing next-hop paths in a specific interface direction, rendezvous paths are more rigid resulting in longer paths chosen.

It is interesting to note, however, that with the exception of going from 4 to 8 interfaces, which we explained was a difficult transition due to lack of angle correction choices, latency remained fairly steady throughout with the increase in the number of interfaces. It can also be seen that despite traveling through longer paths, it seemed that end to end latency was actually less for denser networks than sparser networks. We speculate this is due to better connections between shorter hops compared to longer distance links.

3.4.1.2 Effect of Number of Interfaces on Network Voids

Navigating through voids in our network topology results in higher reliance on the MAM of deviation correction. Because the MAM's efficiency increases with a higher granularity of transmission interfaces (the more interfaces to choose from lead to better ability to control path curves), we hypothesize that by increasing the number of interfaces, more efficient paths could be found and we'd obtain higher

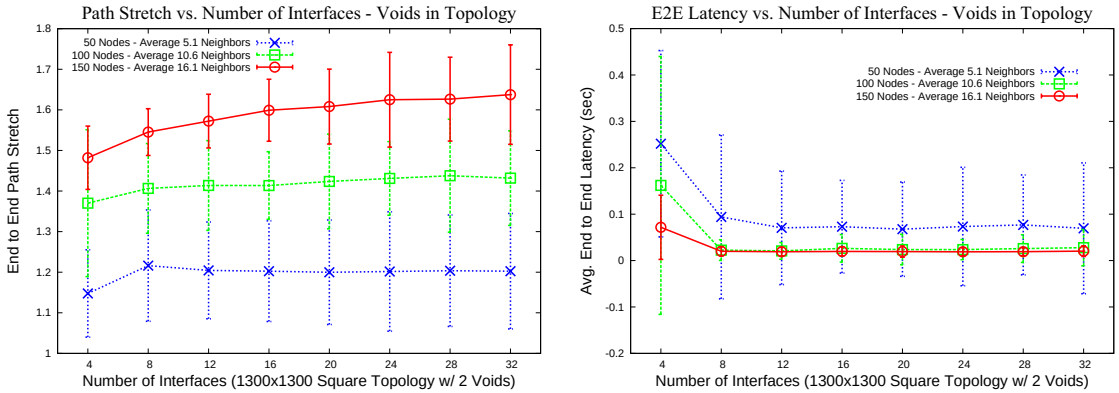
reachability. The conditions for the simulations were consistent with section 3.4.1.1 with the only difference being that the topologies included two voids and had an average of 5.1, 10.6, and 16.1 neighbors per node for the sparse, average, and dense network cases respectively. Void topology generation is described in section 3.4.1 with a void radius between 200 and 300 meters utilized for each void. Figures 3.18-3.19 show our results.



(a) (b)

Figure 3.18: Effect of number of interfaces on ORRP reachability and total states maintained for dense, average, and sparse networks with two voids. In sparse networks, increasing number of interfaces provides significant benefits at first but diminishing returns after 8 interfaces. As with the earlier case, with the exception of going from 4 to 8 interfaces, total states maintained network-wide decreases up to a certain point with an increase in the number of interfaces.

Much like in section 3.4.1.1, our results showed a noticeable increase in reachability with an increase of interfaces from 4 to 8 in both the sparse and average network density case. Again, this is expected due to lack of angle correction options with only 4 interfaces and these results explain the large change from 4 to 8 interfaces in the other figures as well. Total state information network-wide was seen to decrease from 8 to 32 interfaces due to lessened randomness in choosing next hop neighbors in a specific interface direction. As explained previously, having less interfaces meant that each interface "covered" more neighbors. When announcements are transmitted at set intervals, it randomly chooses a neighbor in the direction it



(a) (b)

Figure 3.19: Effect of number of interfaces on ORRP average end-to-end path stretch and latency for dense, average, and sparse networks with two voids. There is a huge decrease in latency going from 4 to 8 interfaces as well as a steady increase in path stretch in dense networks with increasing number of interfaces. Path stretch increases with larger densities.

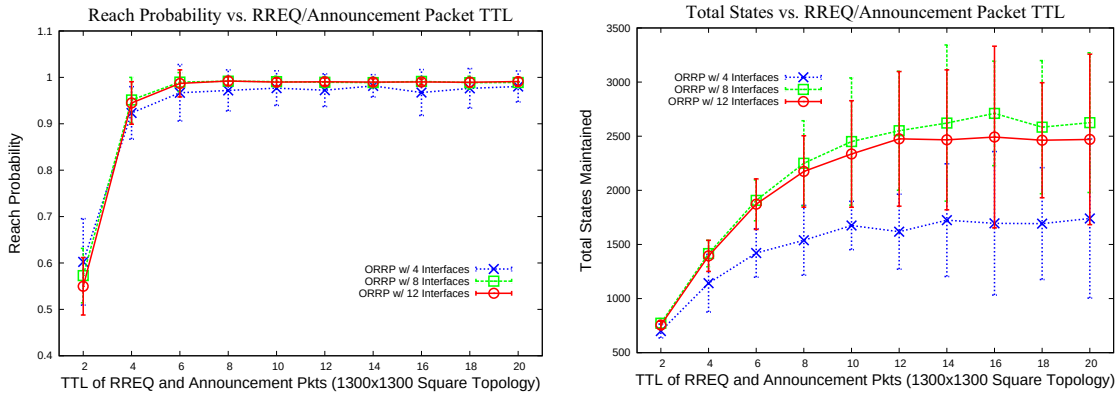
wishes to transmit packets in and sends it to that neighbor. If an interface has multiple neighbors, state information is potentially propagated to both neighbors at different intervals. The overlapping period between when the first state expires at the first node and when the 2nd state arrives at a new node is what causes the extra states network-wide. When there are fewer interfaces, this randomness and overlapping states is removed.

The state issue is also what causes increased path stretch as the number of interfaces increase. The more states are seeded network-wide, the more path choices are available. The only surprising difference in comparing the simulations with and without voids is that with voids, the average end-to-end latency difference in dense and sparse environments is much smaller. This is perhaps due to sparse networks not having many alternatives in path selection to traverse voids, resulting in similar path choices for various end to end paths.

3.4.1.3 Effect of Control Packet TTL on Varying Number of Interfaces

MAM attempts to minimize deviations in path. In sparse networks, however, announcement packets scheduled for orthogonal directions might initially be sent

through the same path due to lack of neighbor options. In traditional routing announcements, one of these packets would be dropped to minimize overhead. In ORRP, however, there is a potential for the packets to "split" to different paths as neighbor density increases. ORRP limits a continual flood of announcement and RREQ packets through packet TTL. While in many cases, packet drops would occur at the network perimeter due to ORRP's MAM forwarding conditions, TTL plays an important role in amount of state needed to be maintained at each node. For our simulations, we used a 100 node (9.1 average neighbors per node) square $1300 \times 1300m^2$ network topology with the default all-to-all traffic pattern (Table 3.7) and varied the control packet TTL. Figures 3.20 and 3.21 show our results.



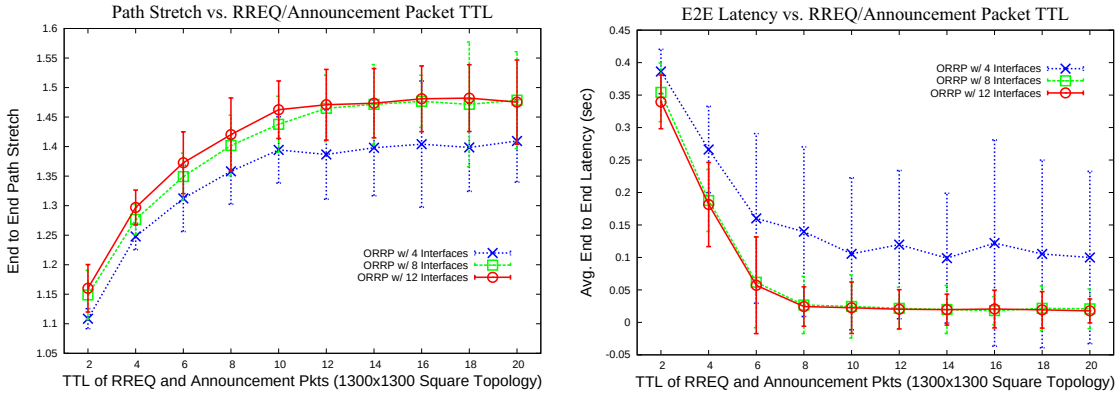
(a) (b)

Figure 3.20: Effect of control packet TTL on ORRP reachability and total states maintained for various number of interfaces. Higher TTL results in higher reach and states maintained network-wide until a saturation point.

Figure 3.20 shows the effect of TTL on the reachability and total states maintained network-wide. Our results showed that varying the number of interfaces did not affect the outcome of the TTL study under average density conditions (9.1 average neighbors per node). As the TTL is increased, however, a large jump occurs between 2 and 4 TTL while the gains taper off into steady-state. We measured the average path length to be approximately 6 hops and so our results are expected: when the control packet TTL is only set to 2, on average, announcement packet and RREQ packet intersections are rare leading to low reach. Increasing to 4 TTL

yields a maximum path length of 8 (4 hops from destination to rendezvous node and 4 hops from source to rendezvous node). While this ensures over 90% of paths found, there are also longer paths that are left out.

Total states maintained network-wide results are also consistent with the reach graph. As the TTL is increased, more states are maintained because announcement packets traverse more nodes. However, as the TTL grows past a certain point (12 hops), we notice that the total states remain steady. This is due to multiplier angle method (MAM) dropping packets after traversing along one or two lengths of the edges (see section 3.2.4.2) which is consistent with our results.



(a) (b)

Figure 3.21: Effect of number of interfaces on ORRP average end-to-end path stretch and latency for various number of interfaces. Higher latency with lower TTL is a result of RREQ retransmit timeout waits.

Figure 3.21 show the end-to-end average path stretch and latency. As we saw in section 3.4.1.1, as the number of interfaces increase, the path stretch increases. What was interesting is that as the control packet TTL increases, the path stretch increases as well. This is perhaps due smaller TTLs resulting in end-to-end paths that are closer to the source and therefore more optimal. As the TTL increases, RREP packets with longer paths might arrive at the source earlier than RREP packets with shorter end-to-end paths. This results in the initial packets buffered for a certain destination being sent along the longer paths until the 2nd RREP packet arrives with the shorter information. One method of dealing with this is to

keep packets buffered for a longer duration of time to allow time for the RREP with the shortest end-to-end path to arrive. This, however, incurs additional delays in latency.

Our latency results are also consistent with expectations. With shorter control packet TTL, there are many rendezvous that are not found. When no RREP is received after a certain timeout, another RREQ is sent. Because RREQs, like announcement packets, are sent to random neighbors in an interface direction, a new path can potentially be taken resulting in a path candidate found. The time between waiting for timeout and retransmitting RREQ is what causes higher end-to-end latency when the control packet TTLs are low.

3.4.1.4 State Information Maintained

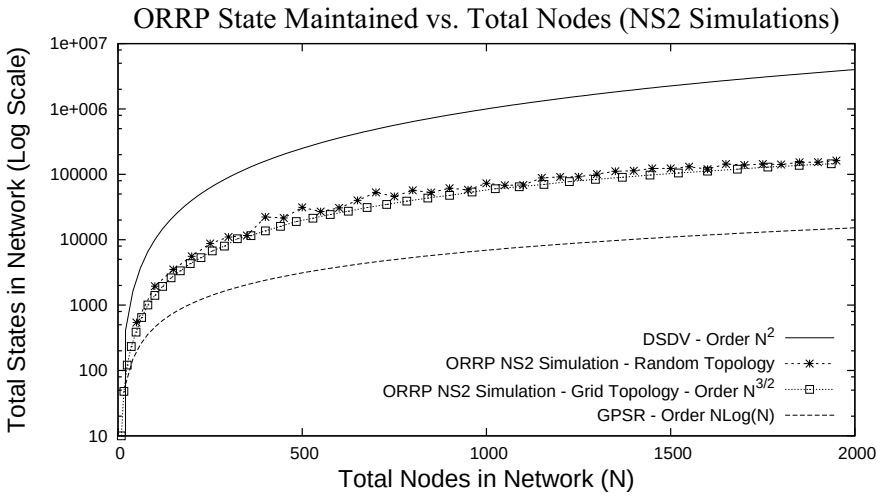
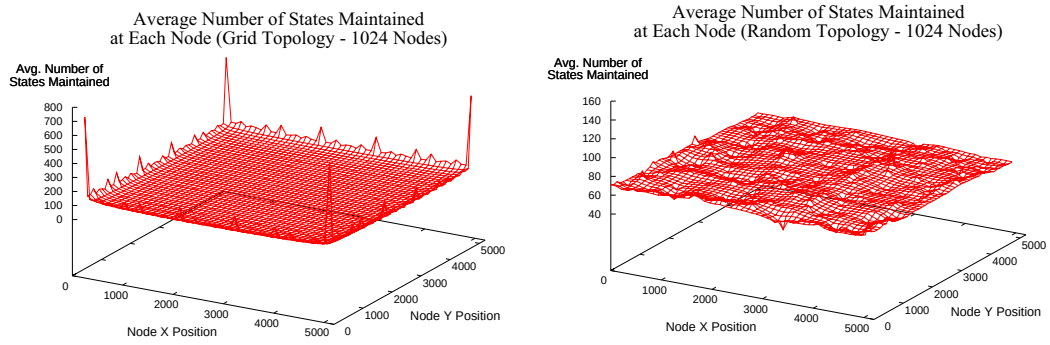


Figure 3.22: NS2: ORRP Total State Maintained vs. Total Nodes in Network

ORRP was run in with grid and random topologies for several numbers of nodes and the total state maintained throughout the network tracked. Figure 3.22 shows the total amount of states maintained vs the total number of nodes in both grid and random topologies. Lines fitted to both plots show an order $N^{3/2}$ maintenance of state at each node.

To understand the distribution of where on the topology nodes generally kept more state, a 1024 node scenario was run in grid and random topologies and the



(a) (b)

Figure 3.23: NS2: State Maintained in Network Topology. ORRP state is evenly distributed throughout the network.

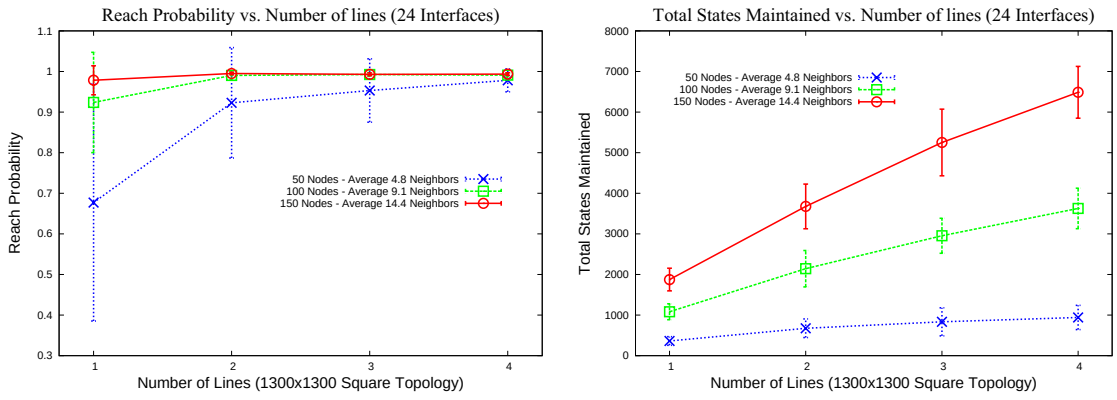
amount of state kept at each node was averaged over 10 trials. Figure 3.23 shows that edge nodes in both grid and random topologies maintained more state than usual. This is expected as perimeter nodes often bear the brunt of deviated routes. One interesting thing to note is that the amount of state information kept at each node is relatively consistent throughout the entire network. This finding is important because it shows that ORRP states are highly distributed and that no single point of failure will drastically affect the network.

3.4.1.5 Effect of Additional Lines on Various Topologies

Section 3.3 showed that under differing topologies without any angle correction, connectivity and path stretch is drastically affected by the number of lines used for transmissions. It is interesting, therefore, to see how the analysis matches up with packetized simulations *with* angle correction. We suspected that even with one line, MAM should be able to deal with the majority of perimeter nodes and therefore provide fairly high reachability in symmetric topologies. In asymmetric topologies, however, as the "incident angle" a packet hits a perimeter node becomes steeper, it becomes more difficult to do angle correction since we set a hard limiter to not forward more than 90° to avoid loops so we suspect in these topologies, additional lines will affect reach probability more drastically.

In the same way, because additional lines provide additional paths to choose from, we expect that as the number of lines increase, the average end-to-end path

stretch from source to destination will decrease. Default simulation and traffic patterns for the ORRP standalone simulations as listed in table 3.5 and 3.7 respectively, were used. The only change was the use of 24 interfaces instead of 12 because 24 is easily divisible by 1, 2, 3 and 4 lines. Section 3.4.1.1 showed that under the densities we're looking at, using more than 12 interfaces should show little difference in the majority of the metrics evaluated. Figures 3.24-3.27 show our results for square and rectangular topologies.

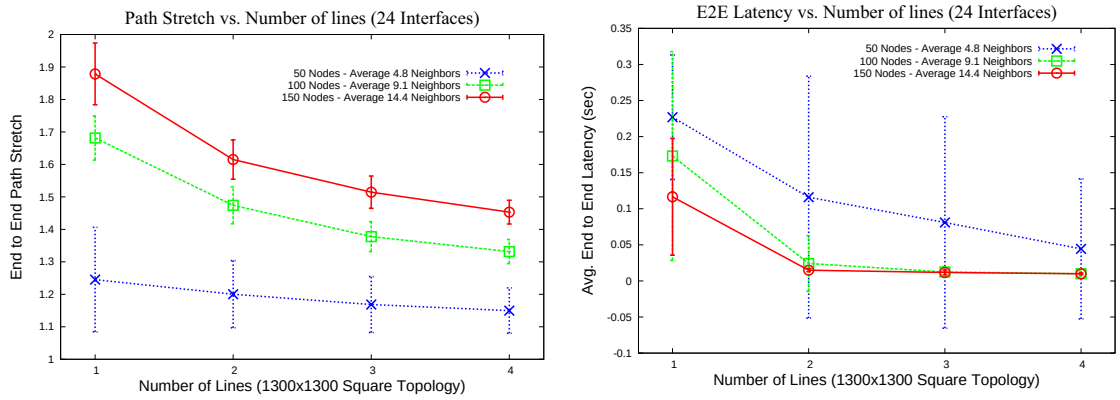


(a) (b)

Figure 3.24: Effect of number of lines on ORRP average end-to-end path stretch and latency for various square network densities. As expected, as number of lines increased, the reach probability and total states maintained increased.

Figure 3.24 show the reach probability and number of states maintained network-wide with ORRP sending packet along 1 to 4 lines. As our analysis in section 3.3 indicated, large increases in reach should occur from one to two interfaces but the gains significantly taper off after that. In sending along only one line, sparser environments revealed very low reach. This is due to fewer next-hop choices and one line limiting intersection opportunities. Our analysis in section 3.3 showed, however, that with one line, only 56.51% reach should be expected. We can easily see that with MAM, our reach increases to an average of 67.7% even in sparse network environments. In denser environments, reach even with one line with MAM yields over 97% reach. As expected, increasing the number of lines also increases the total states maintained network-wide as more announcement packets are being sent in

more directions periodically.

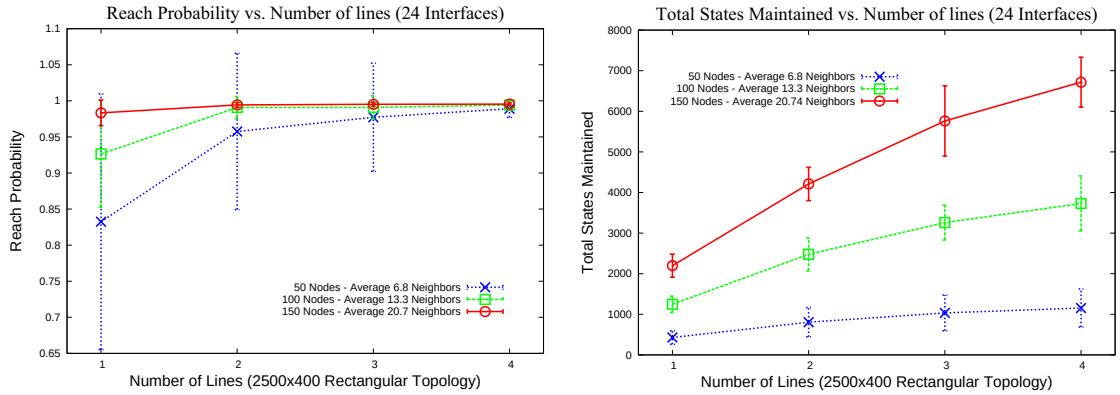


(a) (b)

Figure 3.25: Effect of number of lines on ORRP average end-to-end path stretch and latency for various square network densities. As number of lines increase, path stretch decreases (better paths found) and latency decreases.

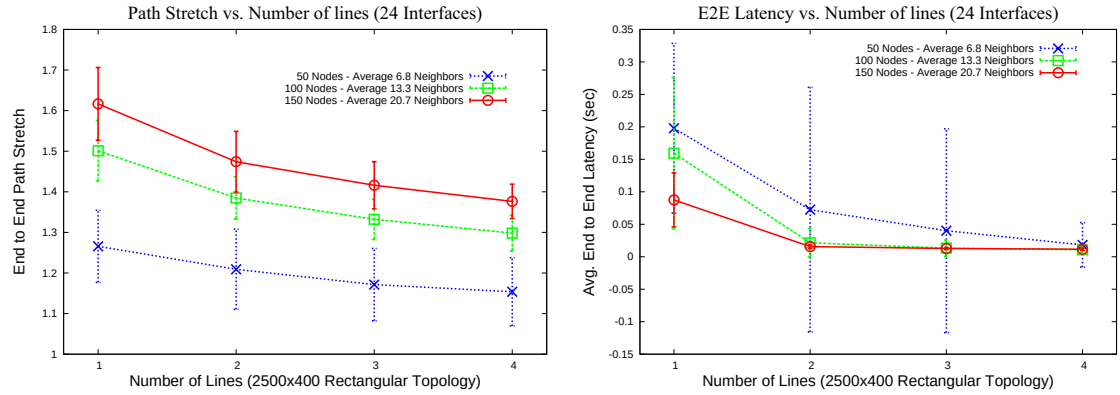
As illustrated in Figure 3.25, for square topologies, average end-to-end path stretch decreases with the increase in lines. This is expected as more lines yields more *correlated* intersections and potentially intersections with shorter path stretches. We explain why denser networks yield higher path stretches in section 3.4.1.1 and the data results are consistent. Average end-to-end latency results are also fairly consistent. With the better paths chosen, latency drops. There is a significant decrease in latency going from 1 to 2 lines because as mentioned in section 3.4.1.4, if paths are not found, ORRP resends RREQs that potentially travel on different paths. The period of waiting for a RREP timeout is what incurs the extra latency.

We saw very similar results for rectangular topologies except that the jump from two to three lines provided a larger jump in reach probability. Even with just one line, MAM was able to ensure roughly 83% packet delivery success as compared to the 34.55% shown in our analysis. By increasing the number of lines, additional paths were available despite the rather "thin" topology. Figure 3.27 shows that the average path stretch also decreases with the number of lines. This again, is expected due to better paths being chosen resulting concurrently with the decrease in average latency. This is therefore consistent with our hypothesis and as expected,



(a) (b)

Figure 3.26: Effect of number of lines on ORRP reachability and total states maintained for various rectangular network densities. Even with one line, MAM ensures high (∼83%) reach.



(a) (b)

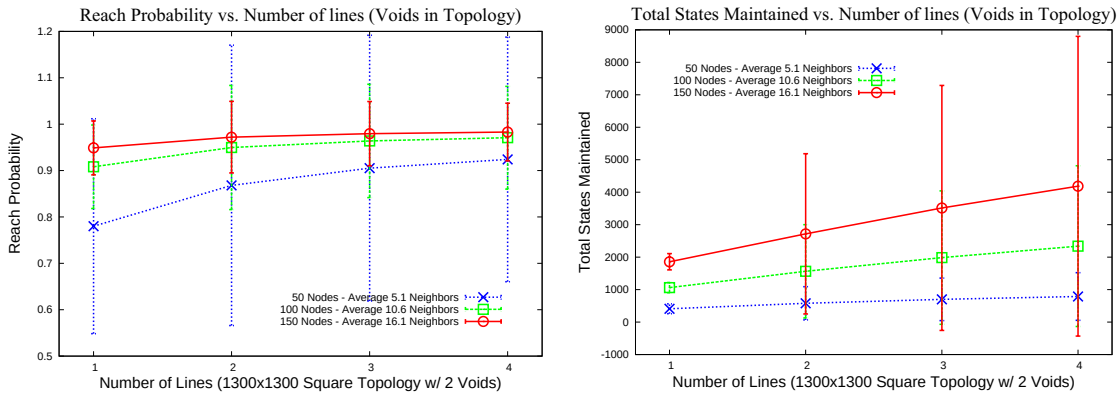
Figure 3.27: Effect of number of lines on ORRP average end-to-end path stretch and latency for various rectangular network densities. With increased lines, shorter paths are found and end-to-end latency drops.

total states maintained in the network grew fairly linearly with increased number of lines.

3.4.1.6 Effect of Number of Lines on Network Voids

It is interesting to see how the number of lines of transmission affect reachability and path length in networks with large voids. We hypothesize that while reach would increase with increased number of lines, average end-to-end path stretch and

latency would remain constant. This is due to few paths to choose from to navigate around voids and therefore, as long as there is a path, most likely, that path would be the one chosen. Default simulation and traffic patterns for the ORRP standalone simulations as listed in table 3.5 and 3.7 respectively, were used with the only change being that we used 24 interfaces instead of 12 because 24 is easily divisible by 1, 2, 3 and 4 lines.

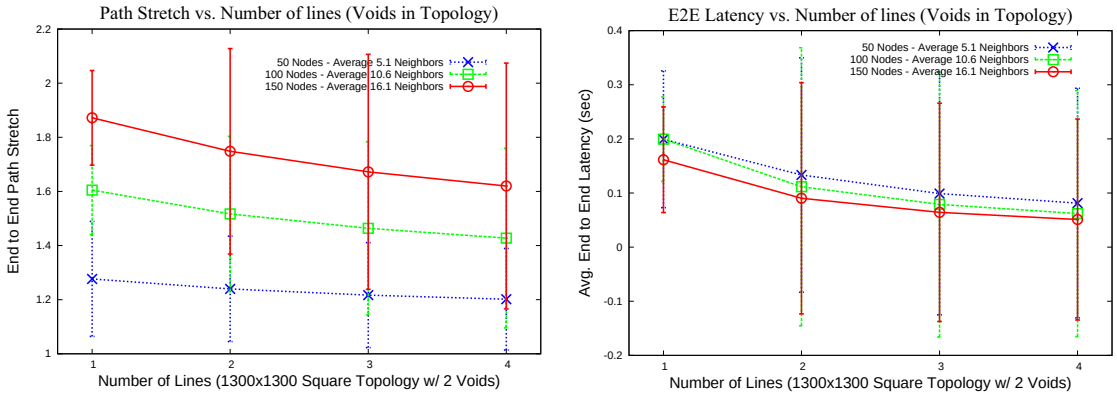


(a) (b)

Figure 3.28: Effect of number of lines on ORRP average end-to-end path stretch and latency for various square networks with two voids. As expected, as number of lines increased, the reach probability and total states maintained increased.

Figure 3.28 shows our results for reach probability and total states maintained for various lines on networks with voids. As expected, the increase from one to two lines yielded a fairly large connectivity gain. Voids in sparse networks are especially difficult for ORRP as even with 4 lines, there is only a 92% reach. This is primarily due to not having nodes to maintain a path around the void and yet keep the packet moving "forward". In sparse environments, the distribution of the reach varied greatly for each run suggesting even deeper issues for void traversals in sparse network environments with ORRP. Total states maintained network-wide, as expected, increased with the addition of lines.

Figure 3.29 give our results for various lines on networks with voids. Our data shows that there is actually a decrease in end-to-end path stretch and latency as the number of lines increased which was not originally hypothesized. This is because



(a) (b)

Figure 3.29: Effect of number of lines on ORRP average end-to-end path stretch and latency for various square networks with voids. As number of lines increase, path stretch decreases (better paths found) and latency decreases.

while packets that traverse voids are limited to one path, in areas where the voids do not affect path decisions as much, additional lines lead to better paths. This is consistent with our results in section 3.4.1.2. The slope of the drop in end-to-end path stretch and average latency is smaller than in figures 3.18 and 3.19 from section 3.4.1.2 for because of the fewer path choices around voids. The average end-to-end latency for each network density was fairly close. This again, is perhaps due to having fewer path choices to traverse voids.

3.4.1.7 Effect of Number of Lines on Varying Number of Interfaces

Adding more interfaces to a node increases the diversity of directions to send with the finer granularity of spread resulting in less neighbors associated with a single interface. Section 3.4.1.1 showed that there is an increase in reachability and average states maintained network-wide and decrease in average end-to-end path stretch and latency with the increase in the number of interfaces up until a point determined by network density. It is interesting, therefore, to understand how changing the number of lines affects networks with varying number of interfaces. As with the previous sections, the default simulation and traffic patterns for the ORRP standalone simulations as listed in table 3.5 and 3.7 respectively, were used.

We evaluated our simulations in a $1300 \times 1300m^2$ network with 50, 100 and 150 nodes and with 8, 12, 16, and 24 interfaces. Because it is important to transmit symmetrically (i.e. the angles between each transmission interface must be equal), certain number of interfaces can only transmit along 1, 2, 3 lines while others can only transmit along 1, 2, 4 lines. The N/A values in the tables represent the cases when transmission is not possible. Tables 3.8-3.9 give our results.

Table 3.8: ORRP Reach Probability vs. Number of Interfaces

	1 Line	**2 Lines**	**3 Lines**	**4 Lines**
8 Interfaces (Avg NBs: 4.8)	76.9%	93.6%	N/A	94.9%
8 Interfaces (Avg NBs: 9.1)	96.9%	98.9%	N/A	99.2%
8 Interfaces (Avg NBs: 14.4)	99.1%	99.3%	N/A	99.2%
12 Interfaces (Avg NBs: 4.8)	75.0%	93.4%	95.6%	N/A
12 Interfaces (Avg NBs: 9.1)	95.6%	99.1%	99.1%	N/A
12 Interfaces (Avg NBs: 14.4)	98.2%	99.5%	99.3%	N/A
16 Interfaces (Avg NBs: 4.8)	73.2%	91.8%	N/A	97.6%
16 Interfaces (Avg NBs: 9.1)	94.8%	99.1%	N/A	99.1%
16 Interfaces (Avg NBs: 14.4)	98.2%	99.5%	N/A	99.3%
24 Interfaces (Avg NBs: 4.8)	72.1%	90.3%	95.3%	97.2%
24 Interfaces (Avg NBs: 9.1)	92.8%	99.1%	99.1%	99.0%
24 Interfaces (Avg NBs: 14.4)	97.9%	99.6%	99.4%	99.4%

Table 3.8 shows the reach probability for varying number of interfaces and network densities with ORRP sending along 1, 2, 3 and 4 lines. It can be seen that in general, for sparse networks, when the number of lines increase from one to two, a large gain in reachability occurs. Afterwards, the gains taper off. It is interesting to note that a network density of 9.1 average neighbors per node equates to approximately 1 neighbor per interface. It makes sense that the affect on delivery success would be most affected by the network density as there is approximately one node per network interface. The lower the number of interfaces, the more neighbors are associated with a specific interface and therefore, there is higher risk of announcement and RREQ packets "missing" each other. Additionally, "matching" one neighbor to a specific interface allows MAM to operate to the best efficiency because it can be consistent when choosing random nodes to send to in a specific

direction.

Table 3.9: ORRP Average End-to-End Path Stretch vs. Number of Interfaces

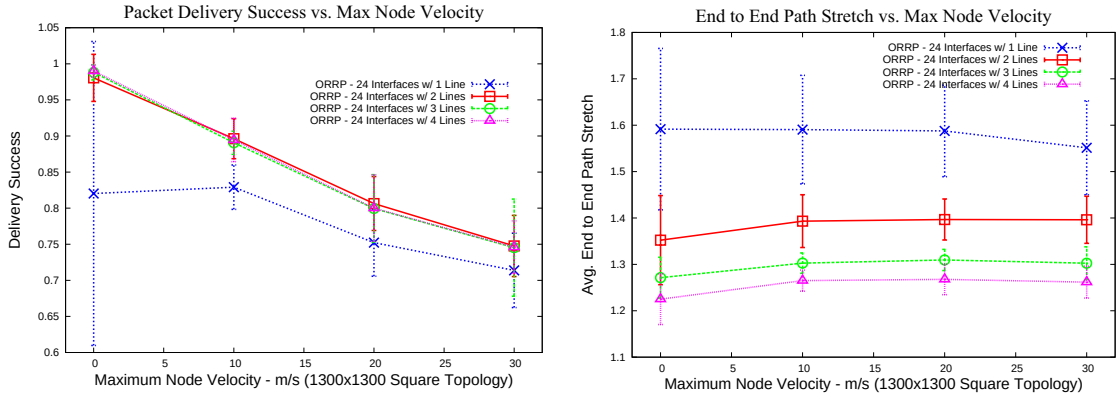
	1 Line	**2 Lines**	**3 Lines**	**4 Lines**
8 Interfaces (Avg NBs: 4.8)	1.28	1.20	*N/A*	1.16
8 Interfaces (Avg NBs: 9.1)	1.64	1.43	*N/A*	1.32
8 Interfaces (Avg NBs: 14.4)	1.79	1.54	*N/A*	1.42
12 Interfaces (Avg NBs: 4.8)	1.26	1.20	1.17	*N/A*
12 Interfaces (Avg NBs: 9.1)	1.68	1.46	1.37	*N/A*
12 Interfaces (Avg NBs: 14.4)	1.84	1.57	1.48	*N/A*
16 Interfaces (Avg NBs: 4.8)	1.27	1.20	*N/A*	1.15
16 Interfaces (Avg NBs: 9.1)	1.70	1.46	*N/A*	1.33
16 Interfaces (Avg NBs: 14.4)	1.86	1.59	*N/A*	1.44
24 Interfaces (Avg NBs: 4.8)	1.26	1.19	1.17	1.15
24 Interfaces (Avg NBs: 9.1)	1.68	1.47	1.39	1.33
24 Interfaces (Avg NBs: 14.4)	1.89	1.61	1.52	1.45

As can be seen from Table 3.9, as number of interfaces increase, the average end-to-end path stretch increases slightly with the exception of the 1 line case. We explained in section 3.4.1.1 that this is due to rigidity in next hop path choices and fewer states network-wide. Overall, these results are consistent with both our analysis and the case we observed with only 2 lines.

3.4.1.8 Effect of Number of Lines on Varying Network Mobility

Because ORRP was designed primarily for fixed wireless mesh networks, it is expected to fail under mobility because lines cannot be maintained in an efficient manner. Adding additional lines, however, could lead to better paths and increased delivery success even in mobile and/or disruption tolerant environments. In this section, we seek to understand whether addition of lines helps in a mobile environment. We suspect that the addition of lines should *not* affect reach probability much because all paths are moving. We simulate using the default simulation parameters listed in table 3.5 in a $1300 \times 1300m^2$ with 100 nodes each outfitted with 24 interfaces. We varied the maximum node mobility speed between 0m/s and 30m/s in

increments of 10m/s and random chose 1,000 source and destination pairs to send 512 byte CBR packets to for 10 seconds. Our results are show in the figures below.



(a) (b)

Figure 3.30: Effect of number of lines on ORRP packet delivery success and average end-to-end path stretch for various maximum node velocities. ORRP performs poorly with node mobility due to inability to maintain straight-line paths.

Our results in Figure 3.30 show that for a mobile network, directional routing protocols like ORRP have severe issues without decreasing the announcement interval and route timeout to ensure fresh routes. However, there seems to be a fairly large increase in reach probability as number of lines increased from 1 to 2 but the gains trail off afterwards. We attribute this increase to having additional and better paths to choose from which in-turn lead to less number of hops and less number of nodes that have moved away providing for a higher reach probability. In the same way, average path length, as expected, decreased with additional lines as better path options were available. It, therefore, a non-trivial problem to leverage directionality in highly mobile environments. Chapter 4 deals with issues with using directionality in *highly mobile* environments.

3.4.1.9 Summary of Standalone Evaluation Results

Below we summarize our findings in our standalone ORRP evaluations:

Reach Probability

- ORRP performs fairly poorly with only *4 interfaces* under *sparse network*

environments because each interface covers over 90° and with angle correction, even shifting transmission by one interface results in a major path deviation. In most cases, because of the sparsity of the network, packets are simply dropped.

- ORRP is not affected much by voids in the topology except in the sparse case with 4 interfaces. For 8 and above interfaces, the results are fairly consistent with the non-voids case.
- ORRP reach increases with TTL up to a certain point. The TTL required is determined by the network size and density.
- Increasing the number of lines yields big jumps in reach from 1 to 2 lines for both square and rectangular topologies but the gains taper off.
- Using MAM yields fairly high reach even with one line in rectangular environments.
- Reach probability drops significantly with high mobility and a mobile version of ORRP needs to be considered.

Total States Maintained Network-wide

- ORRP total states maintained network-wide drops until steady-state with the number of interfaces because everytime ORRP sends out announcement packets, it randomly chooses a neighbor in each interface direction to send. With fewer interfaces (and larger areas covered by each interface), there is a potential for different neighbors to receive state information everytime ORRP sends out announcements. This causes a larger number of states to be seeded network-wide as older states have not yet expired. With the finer granularity of interfaces, there is fewer choices in neighbors to send and therefore, states are fairly consistent network-wide. This is true for all topologies with and without voids
- ORRP states are fairly evenly distributed network-wide suggesting no single point of failure and grows on order $N^{3/2}$.

- ORRP states grow fairly linearly with the addition of lines.

Average End-to-End Path Stretch

- ORRP average end-to-end path stretch increases with node density for square and rectangular topologies with and without voids because ORRP has no notion of neighbor distances. With denser networks (more neighbor choices), nodes have a higher probability of choosing neighbors that are closer physically to itself and require more hops to the destination.
- Increasing the number of interfaces leads to a slight increase in end-to-end path stretch due to more states maintained network-wide for fewer interfaces above 8 interfaces. This results in better options for path selection.
- Increasing the number of lines leads to a decrease in path stretch because there are more rendezvous nodes and more choices for end-to-end paths to select from.

Average End-to-End Latency

- Average end-to-end latency remains fairly constant under conditions of increasing interfaces for square and rectangular topologies with and without voids except with 4 interfaces.
- Latency is higher for cases with 4 interfaces under sparse networks because there is low reach probably meaning that repeated RREQ packets needed to be sent. Since RREQ packets are retransmitted only after the RREQ timeout, the additional delay is incurred in the RREQ timeout.
- Latency drops with number of lines because better paths are chosen.
- In situations of voids, the latency difference between different density networks is smaller because there are fewer paths around voids.

3.4.2 Comparative Performance Evaluations

For our *comparative evaluations*, we choose to evaluate ORRP against three commonly used protocols: AODV [11], OLSR [34], and GPSR [5] with GLS [6] as the location service. AODV is a *reactive* routing protocol that generates routes "on-demand" by flooding the network with route request (RREQ) packets and building routing tables based on next hop. Reactive protocols trade-off flexibility in route discovery and better medium usage under lightly loaded situations for higher transmit latency as data packets need to wait for routes to be found before sending.

OLSR, on the other hand, is a *proactive* routing table which periodically floods the network with link state packets, building routing tables based on the whole network topology. While providing immediate access to optimal paths, proactive protocols put a heavy burden on the medium and much effort is spent on optimizing link state flooding. While designed primarily to limit and remove state information, *position-based* routing protocols like GPSR route using geographic positioning information. Protocols like GPSR rely on location services like GLS which map destination IP addresses to physical locations and also requires position systems such as GPS or node localization schemes. These overheads are rarely regarded as one complete entity. GLS periodically updates location servers and when a positioning information is needed, a request is sent to these location services. In many ways, it functions similarly to a hybrid routing protocol.

By contrast, ORRP is a hybrid proactive and reactive routing protocol that does not flood the network but forwards packets along lines. It is interesting, therefore, to see what kind of gains we get by utilizing directionality in a fixed meshed environment. Our *comparative evaluations* examine similar metrics as in the *standalone evaluations*, but under more practical environments.

- **Packet Delivery Success** - Whereas the standalone evaluations focused on reachability, our comparative evaluations focus more heavily on packet delivery success. Reachability does not deal with congestion and load and simply attempts to find paths from source to destination. Packet delivery success focuses on how well a protocol handles network load as more and more nodes attempt to communicate simultaneously. It is expected that using a directional

form of communication which by default frees the medium up for multiple simultaneous transmissions should lead to higher packet delivery success under the presence of high load. To measure packet delivery success, we simply create a set number of connections, send packets rapidly for several seconds, and measure the number of packets actually received vs. the total number sent. We will show that ORRP delivers a much higher number of packets under high load than AODV, OLSR or GPSR with GLS.

- **Control Packet Overhead (Bytes)** - The notion of *state* varies from protocol to protocol as some protocols maintain simply a destination-nexthop scheme while others maintain much more information. What is perhaps a more accurate assessment how much "work" is required to maintain routing paths is the amount of control packets that are sent by each protocol. The more information is sent network-wide, the more of the medium is used to maintain paths as opposed to sending data. We measure the sent control packet *bytes* as many protocols combine more state information into single packets than others. We show that ORRP sends considerably less control (announcement and RREQ) packets than proactive protocols like OLSR and reactive protocols like AODV under high number of connections and node densities.

- **Average End-to-End Path Stretch** - As mentioned previously, path stretch is defined as the path taken over the actual shortest path (as computed by Djikstra's algorithm). We measure path stretch by taking the actual number of hops traversed by each CBR packet and dividing it by the calculated shortest path and show that ORRP generally chooses better paths than AODV but not as optimal as OLSR. Because OLSR knows the whole network topology and builds routing tables based on link state information, it is expected to almost always come close to shortest path.

- **Average End-to-End Path Length** - Although path stretch gives gives a good picture of a normalized hop count, average path length measures hop count in a more absolute way. We seek to understand what kind of paths are being chosen given a set topology and measure this simply by counting the

number of hops a packet traverses from source to destination. We will show that ORRP chooses shorter paths than AODV but longer than OLSR.

- **Average End-to-End Latency** - Latency is the amount of time for a packet to travel from source to destination. This time includes the time for request and reply exchanges as data packets are buffered during this time. At times, multiple hops over better links are faster than long-haul hops. We evaluate latency by taking the difference between the received time and the send time of each CBR packet. We will show that ORRP has lower latency than AODV, OLSR, and GPSR with GLS even though OLSR chooses more optimal paths. This is due to the more efficient use of the medium.

- **Aggregate Network Goodput** - One of the more attractive things about using directional antennas is the potential for more efficient use of the medium as multiple nodes can send at the same time with lessened interference. This leads to potentially higher goodput. We measure aggregate network goodput by sending CBR packets from all nodes to all nodes simultaneously for 20 seconds, slowly increase the rate and summing the number of bits of data received network-wide. It's expected that as the capacity of the network is reached, more packets will be dropped and a "knee-like" affect will be seen. We show that ORRP utilizes the medium much more efficiently than AODV, OLSR and GPSR with GLS as all these protocols rely on omnidirectional antennas that hog up the medium.

In our comparative analysis, we evaluate each of the metrics above under varying conditions against proactive protocols like AODV, reactive protocols like OLSR, and position-based protocols like GPSR with GLS. These conditions and accompanying explanations of why they are important are listed below:

- **Varying Network Densities** - As the network density increases, scalability becomes a major issue because nodes must share the finite medium [18, 117]. Messaging overhead as well as medium usage negotiations all play a key role in data delivery success. It is therefore interesting to see how ORRP compares

to AODV, OLSR and GPSR with GLS under various conditions of network density from sparse to dense networks. To do this, we utilized the CMU scenario generator that is standard on NS2 [28] distributions to generate wireless topologies with an increasing number of nodes per square area. We used a $1300x1300m^2$ area and increased nodes from 50 to 300 at an interval of 50 nodes, generating five topologies for each density. When the scenarios were generated, the average neighbors came out to be 4.9, 9.1, 14,4, 19.9, 24.5, and 29.4 neighbors for each of the topologies generated with 50, 100, 150, 200, 250, and 300 nodes respectively. For each of the scenarios, we randomly choose 1000 source and destination pairs to send at constant bitrate of 2Kbps for 10 seconds each and measure the metrics described above. It is expected that as the node density increases, there will be much less successful packets delivered due to increased messaging overheads.

- **Varying Number of Connections** - Another way to test scalability is by increasing the number of simultaneous connections. Reactive protocols like AODV and DSR are very much affected by the number of connections because each connection has the potential to require a route lookup incurring large messaging overheads. Since ORRP is a hybrid proactive and reactive routing protocol, it becomes necessary to understand how it scales compared to AODV, OLSR and GPSR with GLS as the number of simultaneous connections increase. To measure this, we fix the number of nodes to 100 (average of 9.1 neighbors per node) and increase the number of 10 second, 2Kbps connections from 1000 to 10000 and capture the effects of this increase on the metrics listed above. It is expected that an increase in connections will have a huge impact on control packet overhead of AODV while not much effect on proactive protocols like OLSR. However, with the increase in the number of packets sent network-wide, flooding overheads will limit the number of successful packet deliveries especially with protocols that utilize omni-directional antennas.

- **Varying Rates of Transmissions** - Varying the rate of a fixed number of

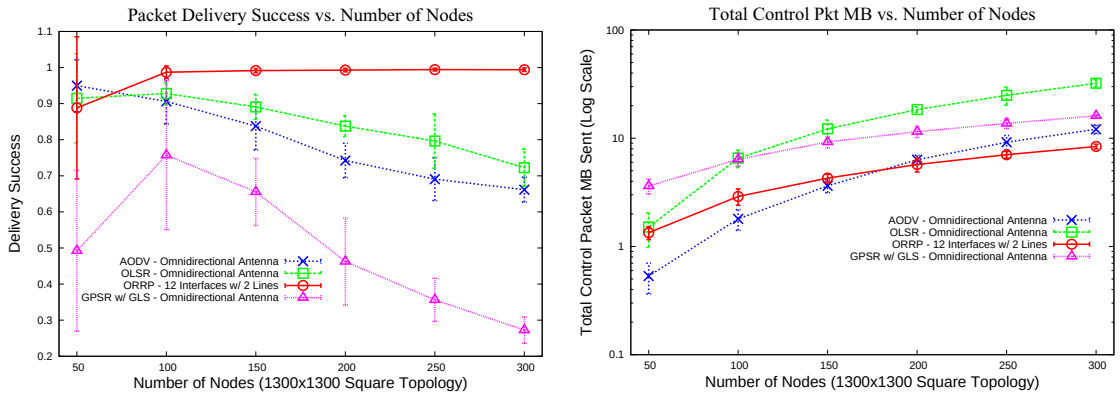
connections is another way to examine the capacity of a network and the capacity gain through using directional antennas in an intelligent way. To do this, we make connections from all nodes to all nodes at the same time and attempt to send data (512 byte packets) at an increasing (2Kb - 20Kb) bit rate for 20 seconds. We expect to see that by using directional antennas, a much higher goodput capacity can be achieved. The difference between varying the rate of transmissions and the number of connections is that it fixes the messaging overheads of reactive protocols like AODV and we can simply measure the overall network capacity.

In the following subsections, we will present our results and discussion of the comparative performance evaluations.

3.4.2.1 Network Density Evaluation vs. AODV, OLSR, and GPSR with GLS

It is interesting to understand how network density affects packet delivery success, average path length and path stretch, total control packets, average end-to-end latency and average goodput network-wide for ORRP compared to other routing protocols like AODV (reactive), OLSR (proactive) and GPSR with GLS (position-based). It is expected that with broadcast protocols that use omni-directional antennas such as AODV, OLSR and GPSR with GLS, as density increases, less packets will be delivered resulting in lower packet delivery success and goodput. Default simulation parameters found in table 3.5 and $1300 \times 1300m^2$ square topologies with no voids (specifics found in table 3.6) were used. For our comparisons, we set ORRP to use 12 interfaces and created 1000 random CBR connections (with a random start time between 10 and 60 seconds) for 10 seconds. Metrics described in 3.4.2 were evaluated and Figures 3.31 and 3.32 show our results.

As can be seen from Figure 3.31, as network density (number of nodes) is increased, ORRP maintains fairly consistent high delivery success while AODV, OLSR and GPSR with GLS steadily decline except for the case of sparse networks. ORRP fairs more poorly under sparse network conditions as seen from section 3.4.1.1 because there are fewer neighbor choices to send in a forward direction resulting in



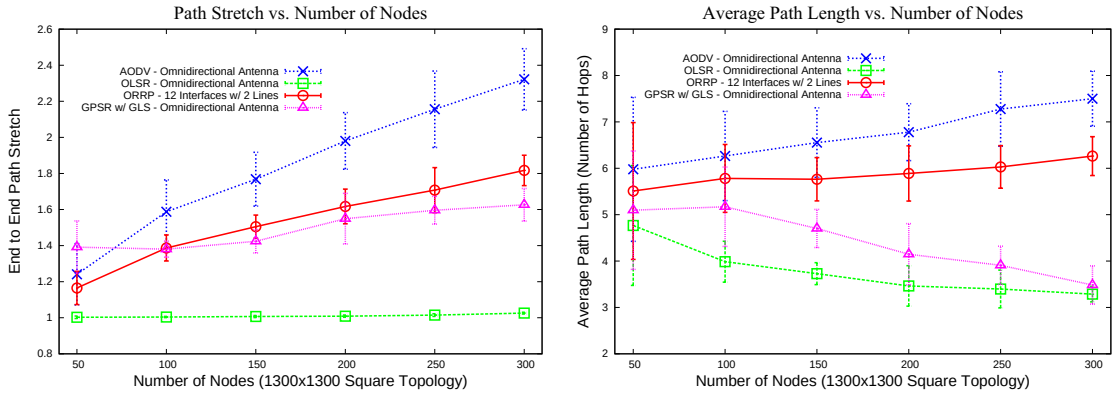
(a) (b)

Figure 3.31: Number of nodes in network vs. packet delivery success and total control packet bytes sent for various routing protocols. It can be seen that as the network becomes denser, ORRP continues to deliver more packets successfully and its control packets sent grows at a smaller rate compared to AODV and OLSR due to its use of directional antennas.

higher drops of announcement and RREQ packets. The same is true of GPSR with GLS as fewer neighbors in the "grid" make it difficult to perform location lookup. For AODV, OLSR and GPSR with GLS, their major limitation is the use of omnidirectional antennas which quickly saturate the medium. For example, Li et al. [6] show that GLS delivers 95% of the packets with 300 nodes in the network, but their traffic pattern is fairly light (only half the nodes sending small packets). By contrast, we overload the network with traffic. Under denser network environments and with growing number of nodes, protocols with omnidirectional antennas simply utilize the medium much less efficiently than protocols like ORRP which utilizes directional antennas.

From a control-packet perspective, both OLSR and AODV grow consistently with an increase in node density. For AODV, RREQ packets need to be propagated network-wide so more nodes equates more control packets. The same is true of OLSR except that OLSR periodically broadcasts link state information rather than sends requests for paths on demand. It makes sense, therefore, that OLSR sends much higher control packets than AODV. ORRP sits right in the middle of the two being a hybrid proactive and reactive protocol. AODV and OLSR by design

sends information to all their immediate neighbors whether in RREQ or link state dissemination phase. ORRP, on the other hand, only sends out certain interface directions and forwards along lines. This would explain why control packets for ORRP grows much slower than with OLSR and AODV with the increased number of network density (number of nodes). GPSR yields no overhead since it is a stateless routing protocol. However, it relies on GLS which has both a periodic update of location servers and a request for positioning information phase. The reason GPSR with GLS overhead is higher than ORRP is again due to the usage of the medium. Even with low overhead, GLS still floods certain nodes to maintain position to ID mappings.



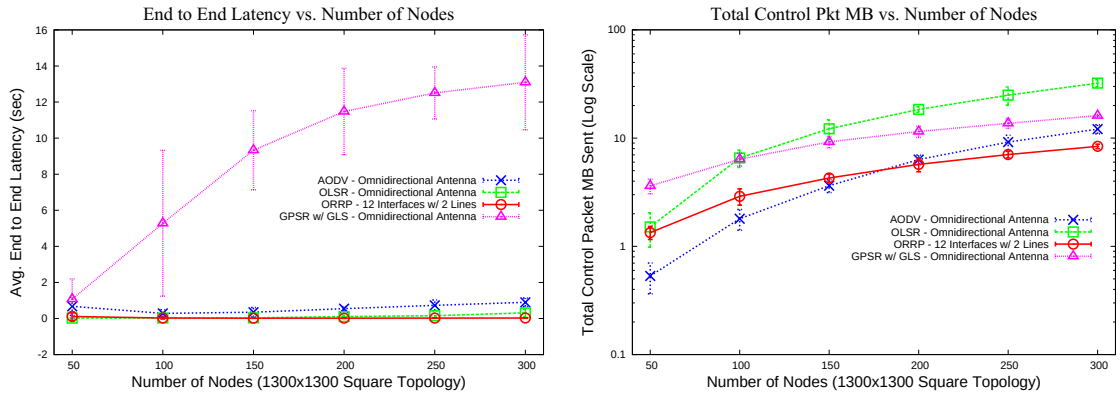
(a) (b)

Figure 3.32: Number of nodes in network vs. average end-to-end path stretch and path length for various routing protocols. It can be seen that ORRP path lengths and stretch sits between OLSR which generates almost optimal paths because of its full view of the topology, and AODV which finds paths on-demand. Although ORRP path stretch is on par with GPSR with GLS, the majority of packets delivered with GPSR fail to reach the destination resulting in only the paths with smallest stretch succeeding.

Figure 3.32 shows the average end-to-end path stretch and path length for OLSR, AODV, GPSR with GLS, and ORRP under various network densities. Because OLSR computes optimal paths based on link state information and has a full view of the network, it is expected (and shown) to have optimal path stretch. The reason why average path length decreases with increased density for OLSR

is because denser nodes tend to have next hops that are toward the edge of the transmission region resulting in longer "jumps". Because AODV generates paths on-demand by flooding the network, it starts forwarding packets once it receives the first RREP until one with fewer hops arrives. This results in non-optimal paths.

ORRP sits in between the two extremes due to its hybrid nature. Unlike AODV which receives multiple RREPs with different path choices, ORRP only receives from 2-4 paths. It can, therefore, make quicker decisions and choose better paths. Average path length for AODV and ORRP grows with the number of nodes simply because there are more next-hop choices to randomly choose from. Unlike OLSR which calculates optimal next-hop, ORRP simply chooses a random neighbor in a specific interface direction to send. GPSR with GLS shows a similar path stretch to to ORRP, however further examination of the average path length graph reveals that only shorter paths are successfully getting delivered with the increase in density. This results a false sense of path stretch since many of the packets fail to be successfully delivered.



(a) (b)

Figure 3.33: Number of nodes in network vs. average end-to-end latency and aggregate network goodput for various routing protocols. Latency is low for ORRP simply because the medium is not saturated with omnidirectional transmissions as is for OLSR, GPSR with GLS, and AODV. Because we are sending at constant rate, the aggregate network goodput graph is consistent with our packet delivery success graph.

Figure 3.33 shows that as the network density increases, GPSR with GLS end-

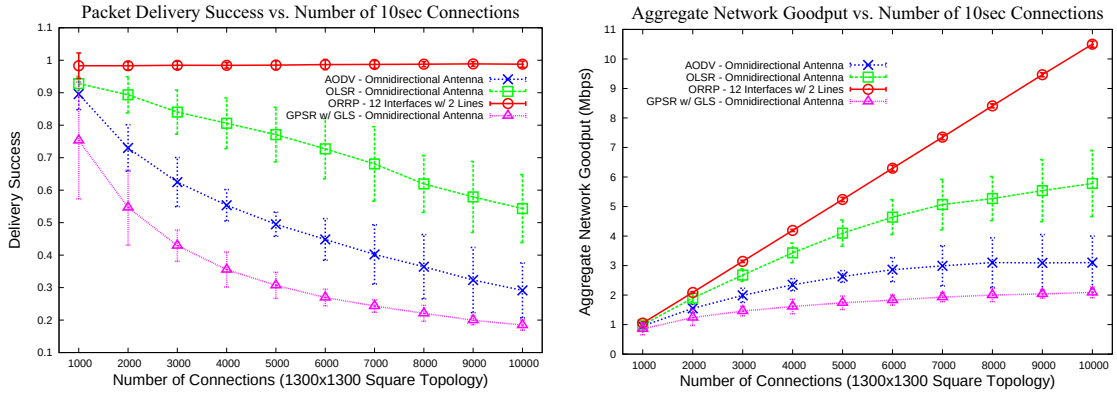
to-end packet latency increases dramatically while ORRP remains constantly low (and lower than AODV and OLSR). This is so because GLS has a difficult time mapping positions to IDs in dense environments because of network saturation. Omnidirectional transmissions performed by AODV, OLSR and GPSR with GLS also make it difficult to transmit simultaneously. If every node in the vicinity of a transmitting node needs to keep quiet to avoid collisions, data packets will remain in the queue longer waiting its turn to send. With ORRP, directional communications methods allow it to simultaneously send and receive from multiple interfaces leading to little delay in sending. The aggregate network goodput graph mimics our packet delivery success graph because we are only randomly selecting 1000 source and destination nodes and sending data at a constant bit rate. Because the number of connections and the constant bit rate does not change with the increase in network density, it is expected that goodput will be dependent on the percentage of packets received.

3.4.2.2 Number of Connections Evaluation vs. AODV, OLSR, and GPSR with GLS

It has been shown that network congestion can be controlled and limited by routing packets using two-phase routing algorithms [50] [49]. Current wireless networks measure route cost through hop count. In high-traffic networks, by choosing the shortest path, nodes with many connections will become saturated with packets. Busch et al. [50] has shown that by drawing a perpendicular bisector between source and destination and forwarding packets from source to a random point on the perpendicular bisector which in-turn forwards to destination when that point is reached, load can be balanced across the network. In much the same way, ORRP inherently implements a seemingly two-phase routing algorithm because it provides rendezvous abstractions whereby the source sends to the rendezvous node and the rendezvous node sends to the destination. In this section, we seek to understand how the number of connections affect the packet delivery success, average path length and stretch, control packets sent, average end-to-end latency, and aggregate goodput network-wide with ORRP, AODV (reactive), OLSR (proactive), and GPSR with

GLS (position-based).

Default simulation parameters found in table 3.5 and $1300 \times 1300m^2$ square topologies with no voids (specifics found in table 3.6) were used. For our comparisons, ORRP was set to use 12 interfaces and between 1,000 and 10,000 random CBR connections (with a random start time between 10 and 60 seconds) were made from random source and destination pairs for 10 seconds each. Metrics described in 3.4.2 were evaluated and Figures 3.34 and 3.35 show our results.

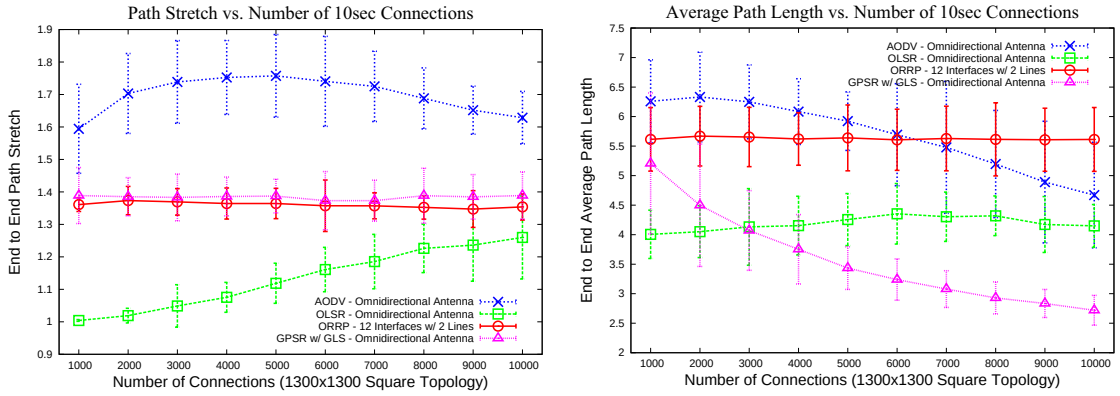


(a) (b)

Figure 3.34: Number of connections vs. packet delivery success and aggregate network goodput for various routing protocols. As connections increase, it can be seen that the network becomes saturated faster with AODV, OLSR, and GPSR with GLS. ORRP maintains high data delivery success and its goodput grows linearly with a linear growth of number of connections due to lack of network saturation.

As can be seen from Figure 3.34, ORRP delivers far more packets than AODV, OLSR, or GPSR with GLS and is fairly consistent in number of packets delivered despite increasing number of connections. This is due to more efficient medium usage by directional communications methods. AODV, OLSR, and GPSR with GLS suffer when the network becomes more saturated as number of connections increase. This is especially true for AODV and GPSR with GLS which performs much worse than both OLSR and ORRP because each connection results in AODV flooding the network with a route request. With GPSR with GLS, the consistent poor performance results from failed positioning requests.

The aggregate network goodput graph is also consistent with our delivery success data: as the number of connections are increased linearly, if the network does not become saturated, the aggregate network goodput should also increase linearly. This is observed for ORRP as ORRP maintains almost 100% data delivery success. With AODV, OLSR and GPSR with GLS, however, we can see that the goodput starts tapering off due to the medium become saturated with omnidirectional transmissions as well as control packets. We see in 3.36 that AODV and GPSR with GLS generates an increasing amount of control packets with an increase in connections (more route and positioning requests) resulting in even further poor medium reuse which explains why it performs much worse than OLSR in this case. We see that at 10,000 connections, ORRP acheives 240% *more* aggregate network goodput than AODV, 82% *more* aggregate network goodput than OLSR, and 400% *more* aggregate network goodput than GPSR with GLS.



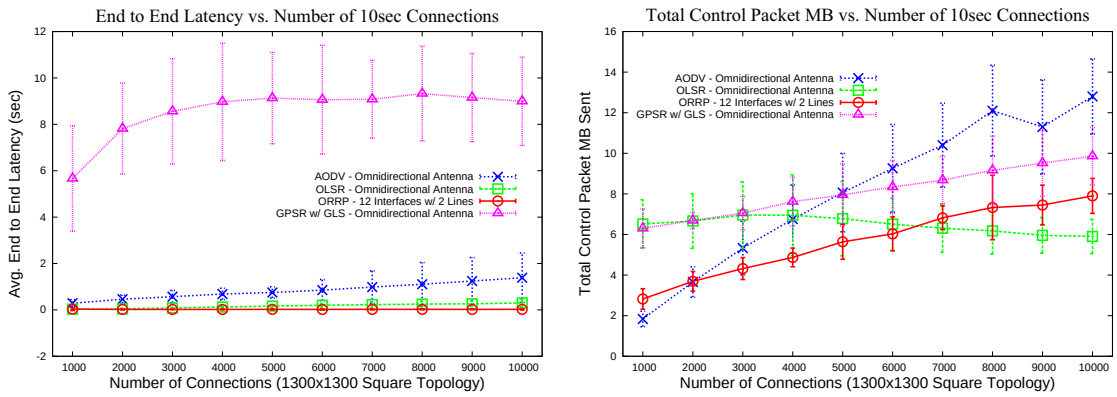
(a) (b)

Figure 3.35: Number of connections vs. average end-to-end path stretch and path length for various routing protocols. As connections increase, it can be seen that ORRP maintains a consistent average paths while OLSR paths become less optimal. AODV achieves better path stretch with increase in connections because closer nodes are the only destinations that are reachable. GPSR with GLS maintains a similar path stretch to ORRP but as can be seen with the average path stretch graph, when the network becomes more saturated, only paths with shorter hops reach. This distorts the stretch factor.

Figure 3.35 shows that ORRP maintains a fairly consistent average end-to-end path stretch and path length even with increased number of connections. This is because again, the network is not saturated leading to always choosing fairly consistent paths. With OLSR which finds optimal paths, when the number of connections increase and thereby saturating the network, suboptimal paths are chosen because link state information is not properly propagated network-wide. With AODV, the path stretch remains consistently high because of its on-demand nature. However, there seems to be a drop in path stretch with the increase in number of connections. To explain this, it is important to look at the average path length graph. As the medium becomes more and more saturated, RREP packets that have the highest chance of returning to the source are those that are closer (in hop count) to the source. Coincidentally, these paths are also easier to optimize. At 10,000 connections, *successful* data packets transmitted using AODV as the routing protocol have an average of 4.7 hops which seem to support our conclusions. Although GPSR with GLS has comparable path stretch to ORRP, we can see from the average path length graph that only destinations with very few hops are being successfully delivered resulting in a distorted path stretch.

Our data in figure 3.36 show that ORRP maintains fairly low packet latency with increase in number of connections. Again, this is expected as there is very little queuing backlog with multiple interfaces as different interfaces can transmit and receive simultaneously. GPSR with GLS latency is much higher than all the other protocols because it relies on GLS to perform positioning lookups to the location servers. If these lookups fail, additional wait time is required. AODV latency is higher than both ORRP and OLSR because AODV is an on-demand protocol and searches for paths only when there is data to send whereas OLSR is a proactive protocol that is periodically disseminating link state information to build optimal paths. The flooding of the medium also hinders latency.

The total number of control packet bytes sent is consistent with assumptions. OLSR, being a proactive routing protocol, periodically disseminates link state information and so even with the increase in number of transmissions, the control packets network-wide remains relatively constant. AODV, being a reactive protocol



(a) (b)

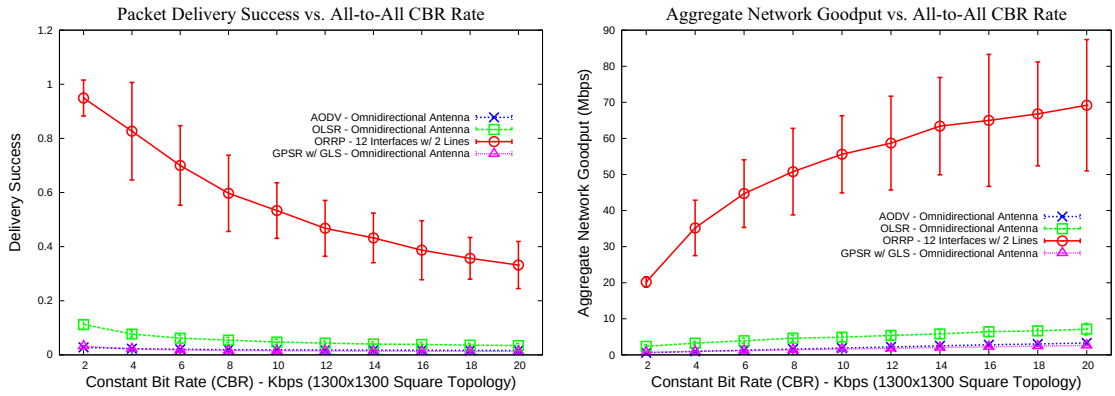
Figure 3.36: Number of connections vs. average end-to-end latency and total control packet bytes sent for various routing protocols. Latency is low for ORRP simply because the medium is not saturated with omnidirectional transmissions as is for OLSR, GPSR with GLS, and AODV. GPSR with GLS yields the highest end-to-end latency even with only short paths transmitting successfully.

that searches for paths on-demand, generates more and more control packets with the increase in number of connections. ORRP is a hybrid protocol that not only disseminates state information periodically and searches on demand, but also leverages directionality to not flood the network. It therefore requires much less control packet bytes to be sent than OLSR and AODV at larger number of connections, while at the same time grows in total control packet bytes sent with the number of connections. GPSR with GLS relies on GLS to find node positioning information. Since GLS has both a proactive element and a reactive position request element, it is expected to grow but not quite the same rate as AODV.

3.4.2.3 Transmission Rate Evaluation vs. AODV, OLSR, and GPSR with GLS

In this subsection, we compare the packet delivery success ratio, aggregate network goodput, average end-to-end latency and total control packet overhead under ORRP, AODV, OLSR, and GPSR with GLS. Our goal is to saturate the network with data packets to see what kind of aggregate network goodput we can

expect with using directional vs. omnidirectional antennas. Because ORRP takes advantage of directionality for medium reuse, it was expected that more packets would be delivered and a higher aggregate network goodput would result. To do this, we send 512 byte CBR packets from all nodes to all nodes simultaneously starting 10 seconds into the simulation and keep it up for 20 seconds. For each run of our simulations, we increase the send rate from 2Kb to 20Kb in increments of 2Kb and measure the metrics listed above. Like before, default simulation parameters found in table 3.5 and $1300 \times 1300m^2$ square topologies with 100 nodes were used.



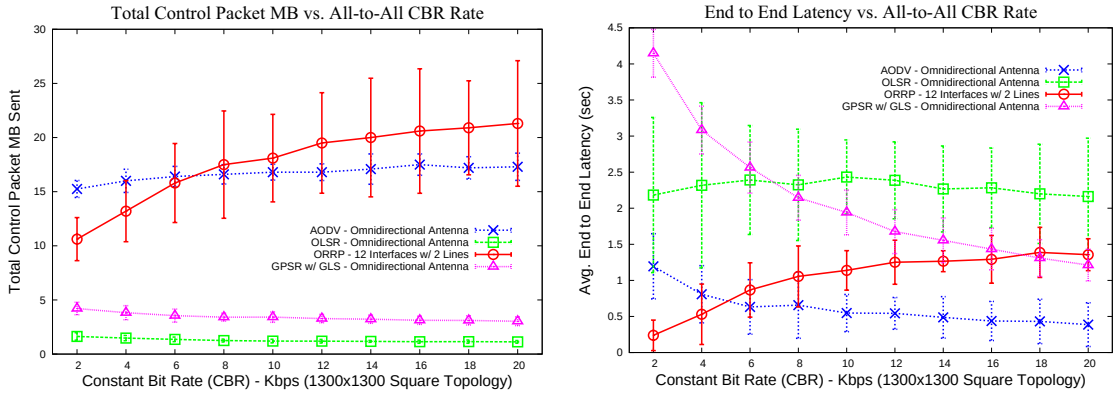
(a) (b)

Figure 3.37: CBR transmission rate vs. packet delivery success and aggregate network goodput for various routing protocols. ORRP achieves about 30x more goodput than AODV, 10x more goodput than OLSR and 35x more goodput than GPSR with GLS.

Our results from figure 3.37 show that even with a very small rate of CBR packets, all-to-all connections flood the network and AODV, OLSR, and GPSR with GLS are simply unable to deliver most of the packets sent (delivering only 2.7% of the packets at best). This result is translated over to the aggregate network goodput graph as we see ORRP achieving over 30x the goodput compared to AODV, 10x the goodput compared to OLSR, and 35x the goodput compared to GPSR with GLS. These gains come from more efficient usage of the medium due to directional antennas. The reason OLSR performs much better than AODV is because for every packet that's sent, AODV must send out RREQ and await for RREP packets. If the network is saturated, these packets have a hard time finding paths and so data

packets without paths are dropped. OLSR periodically signals and builds link state routing tables and so even if updates are not received, it has a route to send packets.

GPSR with GLS performs consistently poor because when a node needs to send a packet, it issues a location request packet. Under conditions of all-to-all flows, every node in the network issues location discovery packets to every other node in the network resulting in a huge overhead and causing many request packets to be dropped. As a result, the source cannot learn the location of the destination and after several unanswered requests, it assumes the destination is unreachable and drops the packet. In our simulations, we noticed that most of the dropped packets don't even leave the source node.



(a) (b)

Figure 3.38: CBR transmission rate vs. total control packet bytes sent and average end-to-end latency for various routing protocols. ORRP end-to-end latency grows because more packets with longer distances to travel get through. The other protocols suffer with saturated network.

It is interesting to note that AODV, OLSR, and GPSR with GLS control packet overhead remain constant network-wide (figure 3.38(a)), decreasing only slightly. For OLSR, it is easy to understand as it is a proactive routing protocol and sends out link state announcements periodically. Even with increased rate of sending, the amount of control packets disseminated periodically does not change. We only notice a very slight decrease as the rate increases and this is perhaps due to the medium being saturated. With AODV, however, paths are found "on-demand". Because the all-to-all connections were performed at relatively the same time, AODV

would have obtained all the data it needed to forward from source to destination for every node in the network right in the beginning. AODV has a complicated caching, route repair mechanism, and RREQ suppression which explains the little need to update their cache in a non-mobile environment. GPSR with GLS too, sends requests for node positioning information and with the saturation of the network, these requests fail.

ORRP, however, employs a simple route expiry mechanism and as such, must re-send route requests when the state information in the routing table goes stale. In our simulations, we set this to 5.0 seconds. Because the CBR duration was set to 20.0 seconds, it would require us to send out 4 RREQs for the total 20.0 seconds of transmission. As the bit rate increases, RREQs will be sent out at a quicker interval before our RREQ suppression techniques can be employed. This results in more control packets sent network-wide with the increase in send rate.

The average latency graph shows that initially, data sent using AODV and GPSR with GLS have very high latency. This is expected because even at a low rate, AODV is flooding the network while GLS is being used to successfully query for positioning. Latency gets better with increased CBR because only successful packet transmissions are measured and with AODV and GPSR with GLS, very few packets are getting through. With ORRP, latency is initially very good because the network is not very saturated. As the network becomes more saturated, however, delivery latency increases. OLSR has high average end-to-end latency because like AODV and GPSR with GLS, it is being limited by the network capacity, but unlike AODV and GPSR with GLS, more packets (with longer latencies) are getting through resulting in much delayed packets being counted toward latency.

3.4.2.4 Summary of Comparative Evaluation Results

Below we summarize our findings in comparing ORRP with AODV, OLSR, and GPSR with GLS:

- **Packet Delivery Success** - ORRP consistently delivered almost 100% packets even with increase in network density and number of connections. This is due to the more efficient usage of the medium vs. AODV, OLSR, and GPSR

with GLS which rely on omnidirectional antennas.

- **Control Packet Bytes Sent** - Because ORRP is a hybrid proactive and reactive protocol, the number of control packets sent grows with increased number of connections. It, however, grows at a much lesser rate than AODV with increased connections and network density.

- **Average End-to-End Stretch and Path Length** - ORRP chooses non-optimal paths but we show that the path selection is much better than reactive protocols like AODV. OLSR maintains optimal paths through periodic expensive link state broadcasts. GPSR with GLS paths stretch is similar to ORRP, but only because most of the medium to longer paths fail resulting in a distorted path stretch.

- **Average End-to-End Packet Latency** - Data packets being routed via ORRP maintain very low end-to-end packet latency. This is due to the efficient reuse of the medium as protocols that utilize omnidirectional antennas cannot send to multiple neighbors simultaneously.

- **Aggregate Network Goodput** - ORRP achieves about 30x the aggregate network goodput compared to AODV, 10x the aggregate network goodput compared to OLSR, and 35x the aggregate network goodput compared to GPSR with GLS. AODV fails under heavy network saturation and its RREQ packets often are not answered. The same is true of GPSR with GLS as GLS position requests are unsuccessful. OLSR fails due primarily to inefficient usage of the medium due to omnidirectional antennas.

CHAPTER 4
Mobile Orthogonal Rendezvous Routing Protocol

4.1 Introduction

Mobile Ad-hoc Networks (MANETs) and more recently, Disruption Tolerant Networks (DTNs) have attracted a high degree of interest in recent years due to its fluid and flexible nature. To facilitate connectivity in MANETs and DTNs, routing protocols have had to grapple with the twin requirements of connectivity and scalability in increasingly mobile environments. The problem of scalability in MANETs and DTNs is two fold: (1) as mobility increases, routing information becomes stale quicker resulting in the need for more route refreshing and (2) as network size and density increases, maintaining routes from every source to every destination becomes increasingly costly. Researchers have tackled this issue through novel methods such as maintaining hierarchies where link/node information of closer neighbors are maintained and routing beyond groups done between backbone nodes [13], propagating link state information to farther nodes at decreased intervals of time [14], among others. While effective in their own ways, with high mobility, many of these schemes rely on increased rate of route dissemination that ultimately leads to decreased scalability.

A recent trend in wireless communications has been the desire to leverage directional forms of communications (e.g. directional smart antennas [17], Free-Space-Optical transceivers [23], and sector antennas) for more efficient medium reuse, increased scalability, enhanced security and potential for higher achievable bandwidth. Previous work in directional antennas focused heavily on measuring network capacity and medium reuse [17] [18]. In these works, it was shown that with proper tuning, capacity *improvements* using directional over omnidirectional antennas are dramatic - even just 8 directional interfaces results in a theoretical capacity gain of 50X.

Additionally, there has been a large push in the free space optical (FSO) community to use FSO to compliment traditional RF methods [53]. FSO has sev-

eral attractive characteristics like *(i) dense spatial reuse*, *(ii) low power usage*, *(iii) license-free band of operation*, and *(iv) relatively high bandwidth compared to RF* but suffers from *(i) the need for line of sight (LOS) alignment* and *(ii) reduced transmission quality in adverse weather conditions*. Yuksel et al. [23] proposed several ways to mitigate these issues by tessellating low cost FSO transceivers in a spherical fashion and replacing long-haul point-to-point links with short, multi-hop transmissions.

Given the seemingly large increases in medium reuse (and thus, scalability) and potential for higher bandwidth in directional forms of communications, it becomes interesting to investigate how *directionality* can be used to complement and even enhance wireless networks in all layers of the stack. There are several challenges associated with using directionality in mobile networks. Unlike omnidirectional antennas where neighbor reach depends almost exclusively on range, nodes using directional antennas need also take into account the neighbor's direction and map it to a specific interface in that direction. The problem is complicated even further as nodes closer to a source seemingly incur more dynamism (even small movements can affect perceived direction dramatically) while nodes farther away incur less change. In this chapter, we address these issues and propose utilizing directionality for a novel purpose: to facilitate layer 3 routing in *highly mobile environments* without the need for flooding either in the route dissemination or discovery phase. Most prior work on leveraging directional antennas in the routing layer focus on adapting routing protocols to simply *utilize* directional communications [131][41]. Our work is novel in that we utilize *local* directionality as a property *to* route packets itself.

Our protocol, Mobile Orthogonal Rendezvous Routing Protocol (MORRP) heavily extends Orthogonal Rendezvous Routing Protocol (ORRP) introduced in chapter 3 to the MANET context. Like ORRP, MORRP is based on two fundamental primitives: a) local directionality is sufficient to maintain forwarding of a packet on a straight line, and b) two sets of orthogonal lines in a plane intersect with high probability even in sparse, bounded networks. We showed showed that in *static* wireless mesh networks, by forwarding packets to nodes intersected by a pair of orthogonal lines originating from a source and destination, one can success-

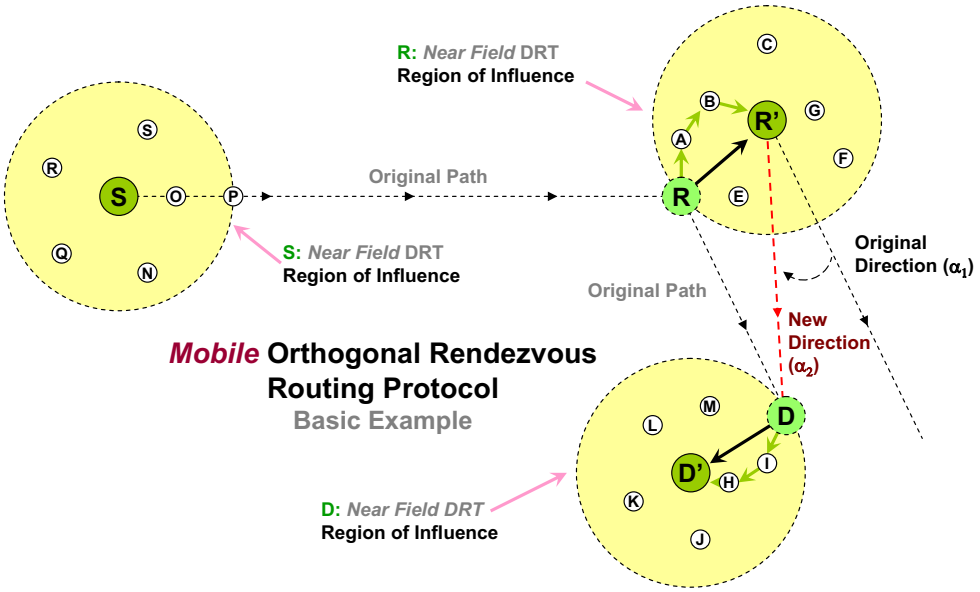


Figure 4.1: Basic MORRP Example: Source sends to rendezvous node R found in region illustrated which in turn sends to destination D found in the DRT region given.

fully route packets to a high degree of connectivity (∼98%) without the need for coordinate space [1]. Furthermore, it was shown that forwarding using this method state-scales to $O(N^{3/2})$ with the states spread evenly throughout the network, while incurring a path stretch vs. shortest path of only 1.2.

While highly effective and scalable for fixed, unstructured networks, ORRP fails under mobility because straight line paths are difficult to maintain due to reliance on individual nodes/points to provide routing information (see chapter 3). MORRP mitigates this issue and facilitates high mobility by abstracting the concept of *rendezvous nodes/points* even further and introducing *rendezvous regions*. In short, each node maintains some location information about all nodes in its neighborhood (note this is different than 1 hop neighbor information) with the amount of information *diffused* as physical distance from the source increases. As long as data packets traverse the original *direction* setup by the initial RREQ and announcement packets, there is a high probability that it will intersect the *rendezvous region* which will then direct the packet toward the rendezvous point. At the rendezvous node,

the process is repeated with the *destination region* being the goal. MORRP uses *directionality* to tackle the issues created by mobility by routing packets *probabilistically* based on what *antenna sector* of transmission a neighbor resides in and a node's *local* velocity to make adjustments. Essentially, each node forwards packets in the general direction of the intended receiver and over time, this information shifts directions accordingly to a node's *local* velocity. For example, if a source node is moving north, a node originally *east* of the source will *seem* to be moving *south*.

Figure 4.1 illustrates a basic example. Suppose source S wants to send packets to destination D and nodes within R and D's "region of influence" have some information about each node respectively (we will explain how this information is disseminated later). Suppose also that by announcement and RREQ/RREP packets, the path "Original Path" is established between S and D with node R as the rendezvous point. With *infrequent updates* in a mobile environment, node R wishes to maintain a general direction to node D based solely on *local* information (its own mobility pattern) and adjusts its direction of sending from angle α_1 to α_2. After some time, if S sends to D, the data packets traversing the original path will "gravitate" toward R' once it hits R's region of influence, and then gets sent in the direction of D until it hits D's region of influence which will then forward packets to the new position of D'. The positioning probability is tracked with directional forms of communications in a novel replacement to routing tables we formulate called the *directional routing table* (DRT).

Figure 4.1 illustrates a basic example. Suppose source S wants to send packets to destination D and through announcement and route request (RREQ) packets, the path "Original Path" is established between S and D with node R as the rendezvous node. After some time, node R has moved to R' and node D has moved to D'. With *infrequent updates* in a mobile environment, node R wishes to maintain a general direction to node D based solely on *local* information (its own mobility pattern) and adjusts its direction of sending to D from angle α_1 to α_2. All nodes maintain a "field of influence" where each node knows the relative direction to all nodes in its region. The data packets S sends to D will traverse the original path, "gravitating" toward R' once it hits R's field of influence. Then, it will be sent in the modified direction

of D until it hits D's field of influence and "gravitates' toward the destination.

MORRP routes packets using directionality in highly mobile environments by 1) shifting destination node directions based on a node's *local* velocity and 2) increasing probability of finding nodes by introducing "fields of influence". All of this is done through a novel replacement to routing tables we formulate called the *directional routing table* (DRT).

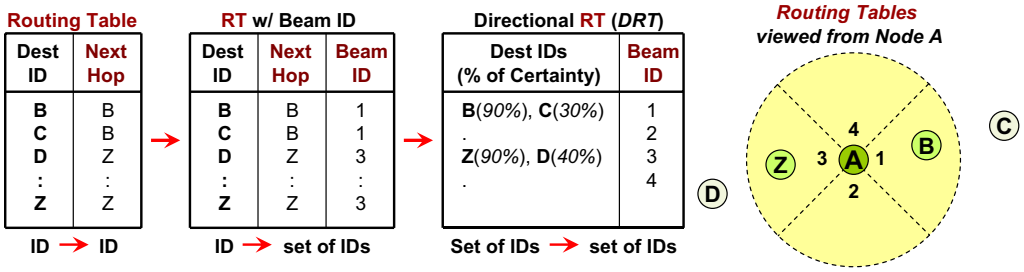


Figure 4.2: Directional Routing Tables (DRTs) map a direction to a set-of-IDs stored in bloom filters

The concept behind DRTs is simple: instead of maintaining *destination IDs* to *next-hop IDs*, we map a *probabilistic set-of-IDs* to each *interface direction* as shown in figure 4.2. The set-of-IDs are stored in bloom filters that are aggregated and sent to neighbors who merge them with the set-of-IDs associated with the interface of receipt. The information in the filter becomes less useful as we progress in time and space and thus we decay (remove bits) from each bloom filter before sending it to its neighbors to capture this effect. Closer nodes have "more" information because the rate at which they are being updated by the source node is higher.

Because DRTs only maintain information on each interface rather than on specific routes in a network, it adds more robustness to mobility as it provides several alternative paths for reaching a destination. In short, any next-hop in a particular direction can take the packet forward. Naturally, the closer a packet gets to a destination node, the information intermediate nodes have about the location of the destination increases. For destinations too far for the source to have any information about the location, MORRP, much like ORRP, relies on route request (RREQ) packets sent in orthogonal directions to *rendezvous* with state information

maintained by each node along an announcement path also disseminated in orthogonal directions. This lightweight method of information dissemination ensures low control overhead from being flooded network-wide.

Key contributions of MORRP include:

- **Using directionality to solve the issues caused by high mobility in MANETs** - By dividing communication regions to sectors and assigning probabilistic position of neighbor nodes to each sector, MORRP is able to shift probabilities of finding a specific node in a specific sector depending on its own velocity. Notice this is done with only *local* information.

- **The Directional Routing Table** - A replacement for traditional routing tables based on purely *probabilistic routing*. DRTs map a specific *direction* to a *set-of-IDs* which eliminates the need to maintain exact routing information about nodes in a network while lessening the frequency of route dissemination.

- **Routing Based on Probabilistic Hints** - Traditional routing protocols have a hard limit on route expiration. With probabilistic routing, routing information is decayed with time and becomes less and less accurate. Below a certain threshold, the information becomes insignificant.

In comparing with several proactive, reactive, and position-based routing protocols, MORRP shows high data delivery (∼93%+), low packet overhead, and over 6X goodput gains vs. traditional routing protocols and 2X goodput gains vs. traditional routing protocols modified with multiple directional interfaces in highly mobile (30m/s) environments. These gains come from many *key design factors*:

- **Weak state information and probabilistic routing** - MORRP does not maintain complete paths and is thus more flexible to forward packets in mobile environments.

- **Local update of weak state information** - Adjusting the "general direction" of a destination node based on one's local velocity "takes a packet forward" even with infrequent location updates.

- **Field of influence** - Enlarging the intersection area results in a greater probability of finding a path in highly mobile environments even with infrequent updates.

- **Leveraging local direction information** - Limited flooding is curtailed by using local directionality to forward in straight lines and rely on intersections of announcement and route request packets to "find" potential paths. This results in "freeing up" the medium for data. This is especially important because MORRP is a hybrid proactive/reactive protocol.

The rest of this chapter is organized as follows: Section 4.2 and 4.3 outline the concept of MORRP including a detailed explanation of DRTs and several decaying strategies as well as how route information would be disseminated and maintained. Section 4.4 gives a basic numerical analysis on path intersection probability while section 4.5 gives some simulation performance evaluations.

4.2 The Directional Routing Table

One of the underlying mechanisms behind MORRP's *probabilistic* forwarding strategy is the directional routing table (DRT), a simplified method of storing route information by leveraging directional communications methods. Unlike traditional routing tables which map *destination-IDs* to *next hop IDs*, DRTs map a *set of IDs* to a specific interface direction. In other words, all the nodes covered by the transmission sector of a specific antenna are included in the entry for that interface in the DRT. The number of entries in the DRT remains constant based on the number of interfaces and does not grow even as the number of nodes in the network grows. This is done through bloom filters.

The concept of using *bloom filters* in probabilistic routing schemes is not new. Acer et al. [132] and Kumar et al. [52] have suggested novel, decentralized, and scalable ways on how information can be disseminated in various types of networks using bloom filters. Bloom filters are space efficient probabilistic data structures that are used to test whether an element is a member of a set. Given an array of bits A (the bloom filter) initialized to all 0 and a fixed number (k) of hash functions

$(h_1(.), ...h_k(.))$, elements (x) are inserted into the bloom filter by evaluating the element in each hash function and mapping the resultant locations in the array to one $(h_i(x) = 1, i = 1, 2, ..., k)$. Lookups are done in the same way in that if the positions in the bit array corresponding to the hashes of an element all equal 1, then the element is a member of the set.

Kumar et al. [52] introduced exponential decay bloom filters (EDBF), a data structure based on the traditional bloom filter concept. Instead of testing whether an element is part of a set or not (absolute information), EDBFs *count* the number of 1's in the bit array corresponding to the element hash in lookup $(\theta_x = |\{i|A[h_i(x)] = 1, i = 1, 2, ..., k\}|)$. The fraction of bits set to 1 over the number of hash functions can be used to interpret the certainty of an element being in the set. Bits are "dropped" (decayed) using various strategies. In this paper, we apply the concept of EDBFs to store a probabilistic set-of-IDs corresponding to neighbor nodes a sector antenna covers in a MANET. We generalize the term to decaying bloom filter (DBF) as there are many ways to decay bloom filters.

Figure 4.2 outlines the structure for the DRT. In short, a *set-of-IDs* stored in a decaying bloom filter is mapped to each specific *interface direction*. To find the certainty of reaching a node by sending out a specific interface, the DBF associated with the interface is selected and the destination node ID is sent through each hash function. By counting the number of bits set to "1" in the locations where the hashes land, the level of certainty of reaching a destination node by sending out that interface is obtained. Because some IDs can potentially be hashed to the same position in the bit array in DBFs, removing IDs from the DBF can potentially affect other IDs. Therefore, strictly "timing-out" neighbors is impossible.

As time goes on and without frequent updates, the level of certainty decreases. To facilitate this idea, we decrease the level of certainty by "decaying" bits in the bloom filter (i.e. changing bits in the DBF from 1 to 0). Decaying methods can be broken up into two main thrusts: *intra-node decay* which handles how bits are removed using only local information, and *inter-node decay* which dictate how bits are removed as information is passed from node to node. In the following subsections, we detail each method. An in-depth exploration of EDBFs and optimal hash sizes

are beyond the scope of the thesis.

4.2.1 Intra-Node Decay

4.2.1.1 Time Decay

Current routing strategies employ hard timeouts for routing entries, updating routing entries periodically through route dissemination or route discovery. While effective for low mobility situations, high mobility situations can cause routes to become stale quickly if the interval between route updates remain constant. As a result, maintaining accurate routing entries network-wide poses a huge problem as it incurs a much higher overhead. MORRP attempts to mitigate this issue by decaying the likelihood a neighbor or destination is in the direction covered by a specific interface as time moves on. In stationary environments, the probability of a neighbor being in a specific region decays at a constant rate (bits from the bloom filter are removed randomly at a constant rate).

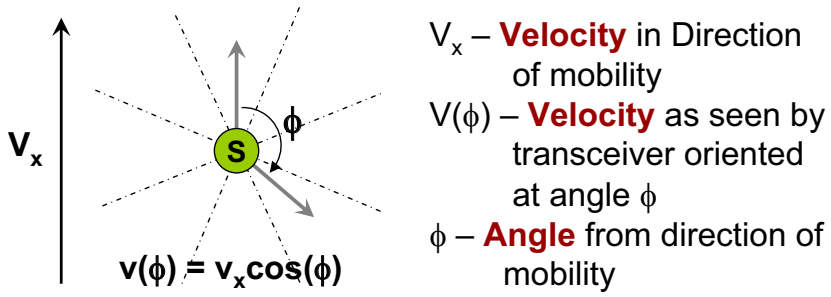


Figure 4.3: Each interface has a different relative notion of how fast a specific node is traveling.

In mobile environments, we employ a different strategy to decay neighbor location probabilities. Figure 4.3 illustrates the basis for our formulation of a simple *time decay* heuristic in mobile scenarios. Assuming all things constant, as a node moves away from its original position, the probability of neighbors in the direction of movement should decay slower than the nodes directly opposite of the direction of movement. In short, the velocity with which each interface perceives itself to be moving at is dependent on the angle the transceiver is from the direction of

movement. As we wish to split the *intra*-node decay between *time decay* and *spread decay*, we will only use half the bits in each bloom filter in our calculations.

We formulate our *time decay* heuristic as follows:

Step 1: Suppose v_x is the velocity a node is moving and ϕ is the angle a specific interface is from the direction of movement. We define the velocity as seen by a specific transceiver v_ϕ as:

$$v(\phi) = v_x \cos(\phi) \tag{4.1}$$

Step 2: If we let R be the range of a transceiver, a node traveling directly away from a specific direction at velocity v_x would be disassociated with a specific transmission region in $\frac{R}{v_x}$ seconds. Taking that into account in our formulation of *time decay*, we specify that all bits of the bloom filter in a specific interface direction must be decayed in $\frac{R}{v(\phi)}$ seconds.

Step 3: Assuming there are k bits of ones in the bloom filter for a specific interface and half of those k bits ($\frac{k}{2}$) are reserved for *time decay*, we wish to linearly decay the number of bits in each bloom filter for each interface with respect to time and velocity. The number of bits to remove per time interval (δ_t) is therefore:

$$\begin{aligned} \delta_t &= \frac{ktD_{tc}}{2} - \frac{ktv(\phi)}{2R} \\ \delta_t &= \frac{kt}{2}\left(D_{tc} - \frac{v_x \cos(\phi)}{R}\right) \end{aligned} \tag{4.2}$$

Where $k/2$ is the number of bits reserved for *time decay* ($\frac{1}{2}$ the total bits set to 1 in the bloom filter), t is the time, D_{tc} is the time decay factor in the stationary case (D_{tc} fraction of bits removed per second in the stationary case), R is the transceiver range, v_x is the velocity in a specific direction, and ϕ is the angle from the current interface to the direction of movement. These bits are removed and discarded.

4.2.1.2 Spread Decay

In a mobile environment with directional communications, the probability a neighbor will be in a certain transmission region/sector is stretched over time. In short, as time progresses, the area a neighbor is possibly located, increases. Figure

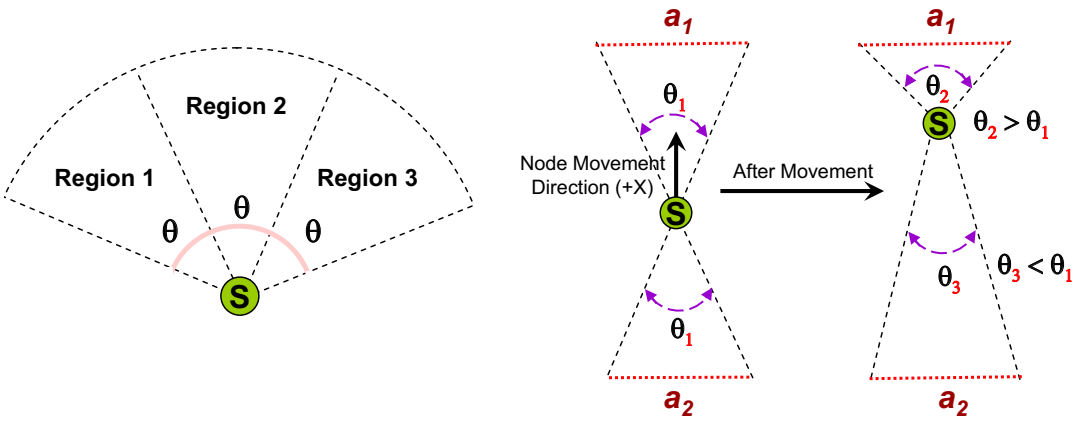


Figure 4.4: Each interface/transceiver has a specific coverage region. As a node moves in one direction, the spread overflows to regions covered by neighboring interfaces.

4.4a illustrates this concept. Suppose a neighbor announces its position to be within region 2. Without knowing what direction and velocity the neighbor is traveling at, as time progresses, there is a greater possibility that the neighbor will be in region 1 and region 3 and a lessened probability that the neighbor will be in region 2. We say that as time goes on, the "spread" for the area the neighbor is in, is increased.

In much the same way, a mobile node traversing in a certain direction will need a greater spread to cover the same area in the direction it is traveling in. Figure 4.4b illustrates this. As a node trying to cover range θ_1 moves in the "+x" direction, it will need a greater spread, θ_2 to cover the same transmission region *in* the direction it is traveling while at the same time, a smaller spread, θ_3 to cover the same region in the direction *away* from the direction it is traveling. Each direction other than the direction the node is traveling in and the direction directly opposite has varied stretch in between these two extremes based on the angle from the direction the node is traveling.

Unlike in our *time decay* heuristic formulation, bits removed from the bloom filter are not discarded but instead, *relocated* to the surrounding directions. The inherent nature of bloom filters allows us to move bits in the DBF associated with a specific interface, to surrounding DBFs, keeping the bits set to 1 *in the same*

hash locations. Due to space constraints, we do not go into details regarding spread strategies. For our simulations, we assume a simple heuristic that whatever bits were affected by the $\frac{v_x \cos(\phi)}{R}$ term in equation 4.2 are affected in the opposite way for spread decay (ie: if the bits were removed, they are spread). When there is no mobility, there is no spread decay. It is important to note the *duality* of *time* and *spread* decay: A neighbor in the direction of travel will incur *less* time decay but at the same time, *more* spread decay.

4.2.2 Inter-Node Decay

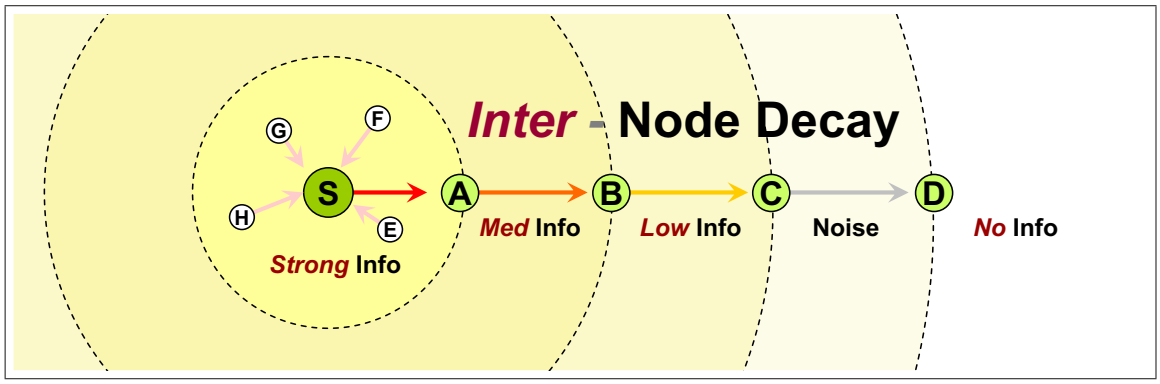


Figure 4.5: Neighbor information is decayed going farther from the source.

The general idea behind decaying the information transferred *between* nodes is that nodes "closer" to a specific source will most likely have more accurate information about the location of the source and nodes "farther" away will have a lessened amount of information. Nodes that are much farther away from the source will have so little information on the source that it will be indistinguishable from "noise". Figure 4.5 illustrates this principle: Node A is a 1-hop neighbor of Node S. Node S aggregates its information about all its neighbors (E, F, G, H, A) and decays this information before sending it to node A. Node A does the same thing with all its neighbors and what results is less and less accurate information about any node in a network depending on the distance that node is from the source.

4.2.2.1 Exponential Distance Decay

Updates are easily created by aggregating the DBFs associated with each interface in the DRT. We follow much of the same aggregation techniques presented in [52] in decaying bits exponentially with number of hops. Exploration of various distance decay methods are beyond the scope of this thesis. Algorithm 2 gives the algorithm for creating updates.

Algorithm 2 DRT Updates

```
1: // Create Local DBF (given local ID x)
2: for all i ∈ {1, ..., k} do
3:    Set bits A[h_i(x)] to 1.
4: end for
5: // Create Update (w/ decay function f_b(n))
6: // Copy all the bits from the local DBF A into update U
7: U ← A;
8: // Decay info received from neighbors stored in DRT
9: for all i ∈ Interface Direction do
10:   for all r ∈ {1, ..., m} do
11:      if A_i[r] == 1 then
12:         U[r] ← 1, with prob. f_b(n + 1)/f_b(n);
13:      end if
14:   end for
15: end for
16: Return U;
```

As algorithm 2 shows, the local node ID is first hashed into a DBF U. Then, each of the DBFs associated with each interface is bitwise decayed according the decaying function $f_b(r)$ and bitwise OR-ed with U. U is then compressed using bloom filter compression [129] and broadcasted out all interfaces to all neighbors.

Upon receipt of the aggregated and decayed DBF from a neighbor, a node will take the max number of bits counted for each entry and bit-wise OR the received DBF with the DBF associated with the interface it receive the packet for the max number of bits for that entry. The reason we cannot simply bit-wise OR the entry with the received DBF is because with increased number of hash functions, probabilities will be biased toward directions with more neighbors since there is higher probability that even if neighbors have same amount of information about a specific node, the bits associated with that information will be more spread out.

Dissemination, which occurs periodically, only takes place between 1-hop neighbor nodes and requires no route/path maintenance. A common assumption in wireless routing protocols is neighbor discovery (each node knows its 1 hop neighbors) and this is usually achieved through periodically broadcasting *hello* packets to all nodes within transmission range. By piggy-backing dissemination information on these *hello* packets, we can therefore disseminate DRT information to our 1 hop neighbors without additional overhead.

4.2.3 Design Variables and Considerations

There are several factors to consider in designing routing algorithms based on DRTs. Table 4.1 lists several parameters that affect successful packet delivery using DRTs. Exploration of all the variables is beyond the scope of this thesis, however, in section 4.5, we examine how varying some of the constraints affect routing in MORRP.

Table 4.1: MORRP Parameters Affecting Successful Packet Delivery

Network Density	Average number of neighbors
Num of Interfaces (ϕ)	The number of interfaces per node
Time Decay Factor (D_t)	Fraction of bits in bloom filter dropped per second per time interval D_i)
Time Decay Interval (D_i)	The time interval to do decaying
Dist. Decay Factor (D_d)	Fraction of bits in bloom filter to drop per hop
Near/Far-Field Threshold ($thresh$ / ff_thresh)	The number of bits found for it to be considered a positive result in searching in NF/FF DRT
Spread Ratio (s_ratio)	The ratio between bits used for spread decay and bits used for time decay
Bloom Filter Size (m)	The number of bits in each bloom filter
# of Hash Funcs (k)	The number of hash functions

4.3 MORRP: Basic Scheme and Design Parameters

MORRP relies heavily on DRTs to provide probabilistic routes from source to destination. Because information about nodes farther away tend not to need to be

refreshed as often as nodes closer to a source [14], MORRP is broken into two major arenas of operation, each with a separate DRT updated at different intervals: *near field* and *far field*. The near field handles direction changes and information about 2-3 hop "neighbors" while the far field handles everything beyond the near field's "region of influence". Near field operation including information dissemination is fairly straight forward and follows what is described in section 4.2.1. A separate DRT is used to keep track of the near and far field respectively. The following subsections will talk about some basic assumptions of MORRP and detail each type of operation.

4.3.1 Assumptions

One of the major aspects of MORRP is that it relaxes many of the assumptions made by position-based routing protocols while still providing connectivity even in *highly mobile* environments. MORRP makes no assumptions on location discovery and uses packets forwarded in orthogonal directions to find paths to the destination from a given source. To do so, MORRP assumes 3 givens:

- *Neighbor to Direction Assignment* - Any given node will know (i) its 1-hop neighbors and (ii) the given direction/interface to send packets to reach this neighbor. This is a fair assumption as the link layer is constantly exchanging ARP, RTS/CTS, among other requests. Removing this assumption requires a "hello" protocol implementation which is fairly standard and trivial.

- *Local Sense of Direction* - Each node must have its own local perception of direction with antennas/transceivers oriented in such a way as to be able to consistently send out orthogonal directions. This can easily be done by selecting any of the transceivers as the "local North" and assigning angles to the others based on that selected transceiver. Nodes must also be capable of communicating directionally over their transceivers. This can be done by various hardware including directional and smart antennas [15], and FSO transceivers [23]. FSO transceivers are a particular interest due to their fine-grained transmit angle and ability for several dozen to be tesselated together oriented in several directions on a single node [23].

- *RREQ and RREP send/receive time is negligible* - We assume that the time required to send a RREQ and receive a RREP (if one is found) is negligible compared to node movement. In otherwords, if a path exists, a node receiving a RREQ should be able to simply record its "previous hop" and "source" so that RREP packets can retrace the route back to the source easily.

4.3.2 Near Field Operation

Nodes within two or three hops (depending on distance decay factor) of a specific source are considered "near-field" nodes because they have *some* information about the position of the source relative to itself. Near-field DRTs are maintained periodically as described in section 4.2.1 and nodes close to a specific source should have adequate information about the position of a destination in the near-field even if they're not an immediate neighbor. Sending to a node in the near-field involves querying each entry in the DRT to return the number of bits in the DBF associated with a specific node ID. The node is said to "have information" about a specific node if the maximum returned bits is greater than a set threshold ($thresh$). The interface with the maximum number of bits associated with a destination node ID and above the threshold bits is then selected as the interface to send the packet and a random neighbor in that direction is chosen to be the forwarder. If there is a tie in the number of bits found for a specific node ID, one is randomly chosen. The process is repeated until the destination is reached.

Additionally, because one of the basic assumptions of MORRP is neighbor discovery in which each node knows its 1 hop neighbors and the interface associated with that interface, if a source wishes to send to its neighbor, it can do so by merely selecting the interface the neighbor resides in and send out that interface. Sending to nodes not within 1 hop from the source but within near-field operation requires querying the *near-field DRT* for a specific destination.

4.3.3 Far Field Operation

Because *near-field* DRTs are decayed between nodes at a substantial decay rate, in general, nodes past three hops from a specific source will have little to no information about the source. To forward packets to nodes where there is little to

no information about position (Far-field operation), MORRP sends route request (RREQ) packets in orthogonal directions (randomly choosing a neighbor in each orthogonal direction) and when one of these RREQ packets intercepts the path of the destination's announcement packets (also sent in orthogonal directions at periodic intervals), a RREP packet is sent back to the source. MORRP stores only *weak-state*[132] at each hop and because of infrequent updates, the far-field DRT is decayed at a slower rate than the near-field DRT. The protocol itself consists of both a *proactive* and *reactive* element and the next sections will detail each element and explain the tradeoffs and design considerations associated with each part.

4.3.3.1 Proactive Element

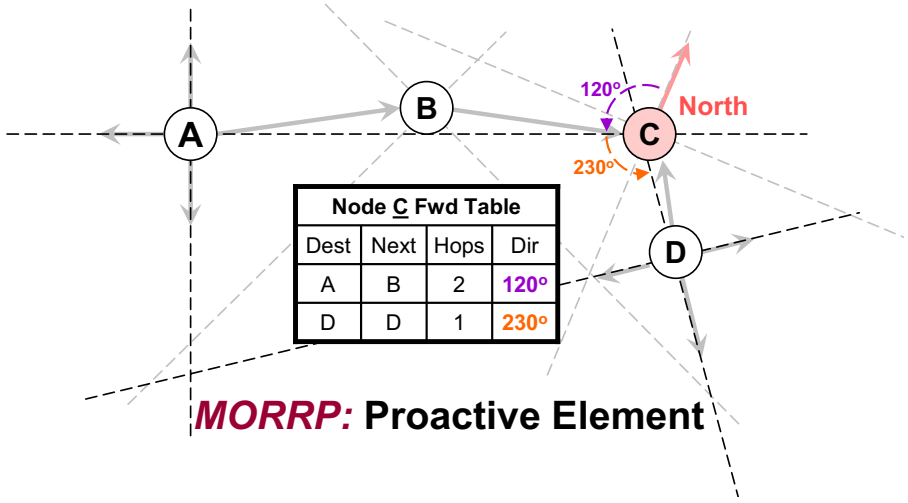


Figure 4.6: MORRP Proactive Element - Announcements used to generate rendezvous node-to-destination paths

In order for a source and destination to agree upon a rendezvous node, pre-established "routes" from the rendezvous node to the destination must be in place. Because each node has merely a local sense of direction, making no assumption on position and orientation of other nodes in the network, it can only make forwarding decisions based on its own neighbor list. As mobility is increased however, routes become stale more quickly. Upon a set interval, each node sends MORRP announcement packets to its neighbors in orthogonal directions as shown in Fig-

ure 4.6. When those neighbors receive these MORRP announcement packets, it hashes the ID of the source of the packet into the far-field DRT entry corresponding to the interface/direction it received the announcement packet and stores/updates the shortest number of hops associated with this announcement sequence number to the announcement source in a "hop count" table if the sequence number of the packet is greater than that recorded in the table or if the table sequence number is the same and the hop count is less (better path). Notice that this "hop count" table is not maintained in any traditional sense and only updated once we have routes. The packet is then forwarded out the interface exactly opposite in direction from the interface it received the packet. If no neighbor is found in the opposite interface to send the MORRP announcement, ORRP's multiplier angle method (MAM) is employed to attempt to maintain straight paths as much as possible. Discussion of MAM is beyond the scope of this chapter (see chapter 3).

Algorithms 3 and 4 detail the basic procedure for sending, forwarding, and receiving MORRP announcements.

The entries in the far-field DRT are decayed in the same way as the near-field DRT with *intra-node* decay methods described in section 4.2 used. In this way, even if nodes are moving, they can maintain a general sense of direction for any source they receive an announcement packet from. Time decaying methods ensure that positioning of nodes become less and less accurate with time and eventually, the information a specific node has about another node becomes negligible if not updated. Unlike the near-field DRT, however, far-field DRT is *not* shared with neighbors so *inter-node* decay is not used.

4.3.3.2 Reactive Element

In order to build the path from source to rendezvous node, an on-demand, reactive element to MORRP is necessary. When a node wishes to send packets to an destination that is not within its immediate neighbor table or near-field DRT, it creates an entry in a simple *destination-rendezvous node* table and sends out a route request packet (RREQ) in all four of its orthogonal directions. Due to the fact that far-field DRTs only track nodes that send MORRP announcements or

Algorithm 3 Send/Forward MORRP Announcement

```
ForwardAnnouncementPacket(p)
 1: // Check if we are the source - forward opposite if not
 2: if p → Src = ID then
 3:    // We are the source, forward orthogonally
 4:    // Get interface ID of local north
 5:    j ← GetLocalNorthIntID
 6:    α ← NumInterfaces
 7:    // Send out orthogonal directions
 8:    for i = 1, i ≤ 4, i++ do
 9:       Φ ←GetRandomNeighbor(j)
10:       // Send to neighbor
11:       send(Φ)
12:       j ← ((j + α/4)%α)
13:    end for
14: else
15:    // We are forwarding - only forward opposite
16:    // Get received interface ID
17:    j ← (p → Recv_Int_Id)
18:    // Get opposite interface j ← ((j + α/2)%α)
19:    Φ ←GetRandomNeighbor(j)
20:    // Send to Neighbor
21:    send(Φ)
22: end if
```

RREQ packets along the line, the *destination-rendezvous* table keeps track of which rendezvous nodes to forward to for a specific destination. Until a RREP is found, this entry is considered unusable. Algorithm 5 outlines how MORRP RREQ packets are sent and forwarded.

When a neighbor node receives this RREQ packet, it hashes the node ID of the source into its far-field DRT and forwards the packet in the opposite direction utilizing MAM. Because one of the assumptions we made is that RREQ and RREP send and receive times are negligible compared to node movement, we need to add a short-timeout reverse path to the source so RREP packets can be sent back quickly. A simple *destination-nexthop* routing table with fast entry expiry times is used for this reverse-route back to the source. Algorithm 6 shows how MORRP RREQ packets are processed upon receipt.

In a 2-D Euclidian plane, by sending a RREQ packet in all 4 of its orthogonal

Algorithm 4 Receive MORRP Announcement

`RecvAnnouncementPacket(`p`)`

1: $p_{src} \leftarrow (p \rightarrow Src)$
2: $p_{int} \leftarrow (p \rightarrow Recv_Int_Id)$
3: $dbf \leftarrow$ `GetDBFfromFarFieldDRTInterface(`p_{int}`)`
4: // Hash Announcement source (p_{src}) into Far-Field DRT associated with received interface
5: **for all** $i \in \{1, ..., k\}$ **do**
6: Set bits $dbf[h_i(p_{src})]$ to 1.
7: **end for**
8: // Get entry from hop-count table, if missing, create one
9: $hc \leftarrow$ `GetHCEntry(`p_{src}`)`
10: **if** $hc = null$ **then**
11: // There's no entry back to announcement source, create one
12: $hc \leftarrow$ `CreateHCEntry(`p_{src}`)`
13: **end if**
14: // Update hop count entry if its a new announcement or if hop count smaller
15: **if** ($hc_{seqnum} < p_{seqnum}$) OR ($hc_{seqnum} = p_{seqnum}$ AND $hc_{hops} < p_{hops}$) **then**
16: $hc \leftarrow$ `UpdateHCEntry(`p`)`
17: **end if**
18: **if** $p_{hops} \geq TTL$ **then**
19: `drop(`p`)`
20: **else**
21: `ForwardAnnouncementPacket(`p`)`
22: **end if**

directions, it is highly likely to encounter a node that has a path to the destination (see section 3.3). When a node with a path to the destination (destination is either in neighbor table or destination ID is above threshold in near or far-field DRTs) receives the RREQ, it sends a RREP packet back the way the RREQ came. Because each node along the path stored a reverse route to the source and we assume that nodes have not moved much in the process of the RREQ being sent, it is able to forward the RREP back efficiently. At each hop, nodes hash the rendezvous node's ID into its far-field DRT so as to provide probabilistic routing options. Finally, when the source receives the RREP, it hashes the rendezvous node's ID into its far-field DRT and updates the *destination-rendezvous* table with the rendezvous node for a specific destination and "activates" that entry.

Algorithms 8 and 9 detail the send, forward, and receive process for MORRP

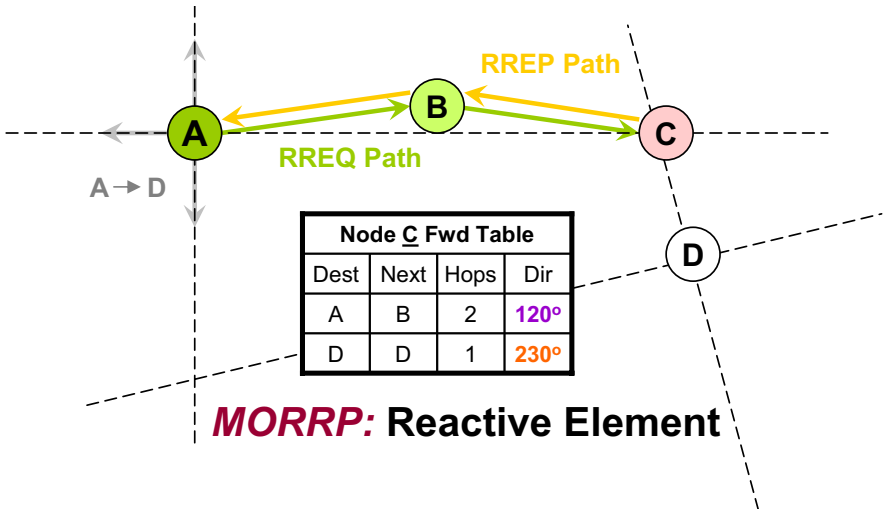


Figure 4.7: MORRP Reactive Element - RREQ and RREP Packets to generate source-to-rendezvous node paths

RREP packets.

4.3.3.3 Data Delivery

For data delivery, if the packet is at the source, first the neighbor list and near-field DRT is queried for the destination. If destination is not found in these two tables, then the far-field DRT is checked to see if the number of bits associated with the destination hash is above the threshold. If destination is still not found in the far-field DRT, then the destination-rendezvous table is queried to see if there is a rendezvous node we need to send to. If it is found, then the far-field DRT is queried for the rendezvous node ID. If after all these steps the destination is unreachable, then a RREQ is sent out in orthogonal directions.

For forwarding packets, a similar approach is taken in that first the neighbor list and near-field DRT is checked for the rendezvous node if its present in the packet header and if not, the destination node. If it is not found in either, the far-field DRT is checked. If it is not found in any of the tables, the packet is simply forwarded to the opposite direction of receipt (the antenna exactly 180° from the receiving antenna). Algorithm 10 shows the basic forwarding steps, filtering out whether the packet is at the source or merely at an intermediate node. It then forwards the

Algorithm 5 Send/Forward MORRP Route Request

`ForwardRREQPacket(`p`)`

```
1: // Check if we are the source - forward opposite if not
2: if p → Src = ID then
3:    // We are the source, forward orthogonally
4:    // Get interface ID of local north
5:    j ← GetLocalNorthIntID
6:    α ← NumInterfaces
7:    // Send out orthogonal directions
8:    for i = 1, i ≤ 4, i++ do
9:       Φ ←GetRandomNeighbor(j)
10:      // Send to neighbor
11:      send(Φ)
12:      j ← ((j + α/4)%α)
13:   end for
14:   // Create an entry in the Destination-Rendezvous table
15:   dre ← CreateDREntry(p_src)
16: else
17:   // We are forwarding - only forward opposite
18:   // Get received interface ID
19:   j ← (p → Recv_Int_Id)
20:   // Get opposite interface j ← ((j + α/2)%α)
21:   Φ ←GetRandomNeighbor(j)
22:   // Send to Neighbor
23:   send(Φ)
24: end if
```

packet using algorithm 11 if it is the packet source, algorithm 12 if it is a forwarding node and the packet has its rendezvous flag set, and algorithm 13 if no rendezvous is set in the packet header.

4.4 Numerical Analysis

In ORRP, a path is established when a RREQ and announcement packet intersect at a rendezvous node. The probably of intersection depends on a point and determines reachability. With MORRP, because nodes are constantly moving, the probability that a RREQ will intercept a node that originally contained announcement information about a destination becomes increasingly slim with time. We say that the information is "diffused" with time. It is therefore interesting to gain in-

Algorithm 6 Receive MORRP Route Request

`RecvRREQPacket(`p`)`

1: $p_{src} \leftarrow (p \rightarrow Src)$
2: $p_{search_id} \leftarrow (p \rightarrow Search_ID)$
3: $p_{int} \leftarrow (p \rightarrow Recv_Int_Id)$
4: $dbf \leftarrow$ `GetDBFfromFarFieldDRTInterface(`p_{int}`)`
5: // Hash RREQ source (p_{src}) into Far-Field DRT associated with received interface
6: **for all** $i \in \{1, ..., k\}$ **do**
7: Set bits $dbf[h_i(p_{src})]$ to 1.
8: **end for**
9: // Create an entry for the reverse route for RREP
10: $rt \leftarrow$ `GetRTEntry(`p_{src}`)`
11: **if** $rt = null$ **then**
12: // There's no entry back to RREQ source, create one
13: $rt \leftarrow$ `CreateRTEntry(`p_{src}`)`
14: **end if**
15: // Update reverser route entry if its a new RREQ or if hop count smaller
16: **if** $(rt_{seqnum} < p_{seqnum})$ OR $(rt_{seqnum} = p_{seqnum}$ AND $rt_{hops} < p_{hops})$ **then**
17: $rt \leftarrow$ `UpdateRTEntry(`p`)`
18: **end if**
19: // Get entry from hop-count table, if missing, create one
20: $hc \leftarrow$ `GetHCEntry(`p_{src}`)`
21: **if** $hc = null$ **then**
22: // There's no entry back to RREQ source, create one
23: $hc \leftarrow$ `CreateHCEntry(`p_{src}`)`
24: **end if**
25: // Update hop count entry if its a new RREQ or if hop count smaller
26: **if** $(hc_{seqnum} < p_{seqnum})$ OR $(hc_{seqnum} = p_{seqnum}$ AND $hc_{hops} < p_{hops})$ **then**
27: $hc \leftarrow$ `UpdateHCEntry(`p`)`
28: **end if**
29: // Look to see if we can send out a Route Reply
30: `RREQCheckRoute(`p`)`

sight on the probability of even finding a rendezvous node in a mobile environment by sending out RREQ and announcement packets in orthogonal directions.

Figure 4.9 gives an illustration of our analysis. Details are left out due to space constraints. In short, assuming a source S wanting to send to a destination D, if the transmission radiuses of the nodes are the green/smaller bands, we say that with a set mobility speed, the maximum an announcement packet path can

Algorithm 7 Search Destination Route Check

`RREQCheckRoute(`p`)`

1: **if** $p_{search_id} = ID$ **then**
2: // I'm what the source is looking for
3: $num_hops \leftarrow 0$
4: `SendRREPPacket(`p, num_hops`)`
5: **else if** $p_{search_id} \in NeighborList$ **then**
6: // RREQ search ID is my 1 hop neighbor:
7: $num_hops \leftarrow 1$
8: `SendRREPPacket(`p, num_hops`)`
9: **else if** $p_{search_id} \in NearFieldDRT$ **then**
10: // RREQ search ID is my Near Field DRT:
11: $num_hops \leftarrow 2$
12: `SendRREPPacket(`p, num_hops`)`
13: **else if** $p_{search_id} \in FarFieldDRT$ **then**
14: // RREQ search ID is my Far Field DRT:
15: $hc \leftarrow$ `GetHCEntry(`p_{src}`)`
16: `SendRREPPacket(`p, hc_{hops}`)`
17: **else**
18: **if** $p_{hops} \geq TTL$ **then**
19: `drop(`p`)`
20: **else**
21: `ForwardRREQPacket(`p`)`
22: **end if**
23: **end if**

Algorithm 8 Send/Forward MORRP Route Reply

`ForwardRREPPacket(`p`)`

1: // Search for path back to RREQ source
2: $rt \leftarrow$`GetRTEntry(`p_{dest}`)`
3: // Send to next hop
4: `send(`$rt_{nexthop}$`)`

deviate from the line is represented by the grey/larger bands. The intersection formed by the smaller green bands (area B) represent the area of nodes that would have have received the RREQ *and* the announcement packets. Additionally, the intersection formed by the larger grey band and green band originating from the source represents the area of nodes where nodes originally along the announcement path would have traveled (area A).

Using Matlab, we iterate through all possible nodes in a network and all possi-

Algorithm 9 Receive MORRP Route Reply

1: // Take care of reverse route to rendezvous node
2: $p_{src} \leftarrow (p \rightarrow Src)$
3: $p_{search_id} \leftarrow (p \rightarrow Search_ID)$
4: $p_{int} \leftarrow (p \rightarrow Recv_Int_Id)$
5: $dbf \leftarrow$ `GetDBFfromFarFieldDRTInterface(`p_{int}`)`
6: // Hash RREQ source (p_{src}) into Far-Field DRT associated with received interface
7: **for all** $i \in \{1, ..., k\}$ **do**
8: Set bits $dbf[h_i(p_{src})]$ to 1.
9: **end for**
10: // Create an entry for the reverse route to rendezvous node
11: $rt \leftarrow$ `GetRTEntry(`p_{src}`)`
12: **if** $rt = null$ **then**
13: // There's no entry back to RREQ source, create one
14: $rt \leftarrow$ `CreateRTEntry(`p_{src}`)`
15: **end if**
16: // Update reverse route entry if its a new RREP or if hop count smaller
17: **if** $(rt_{seqnum} < p_{seqnum})$ OR $(rt_{seqnum} = p_{seqnum}$ AND $rt_{hops} < p_{hops})$ **then**
18: $rt \leftarrow$ `UpdateRTEntry(`p`)`
19: **end if**
20: // Get entry from hop-count table, if missing, create one
21: $hc \leftarrow$ `GetHCEntry(`p_{src}`)`
22: **if** $hc = null$ **then**
23: // There's no entry back to RREQ source, create one
24: $hc \leftarrow$ `CreateHCEntry(`p_{src}`)`
25: **end if**
26: // Update hop count entry if its a new RREQ or if hop count smaller
27: **if** $(hc_{seqnum} < p_{seqnum})$ OR $(hc_{seqnum} = p_{seqnum}$ AND $hc_{hops} < p_{hops})$ **then**
28: $hc \leftarrow$ `UpdateHCEntry(`p`)`
29: **end if**
30: // Process RREP Packet
31: **if** $p_{dest} = ID$ **then**
32: // We are the source of the RREQ/Dest of RREP
33: // Find rendezvous table entry
34: $rne \leftarrow$ `GetRTEntry(`p_{src}`)`
35: $rne_{rend_node} \leftarrow p_{src}$
36: // Send all buffered packets for this destination
37: `SendBufferedData()`
38: **else**
39: // We are NOT the source of the RREQ/Dest of RREP
40: `ForwardRREPPacket(`p`)`
41: **end if**
42: `SendBufferedData()`

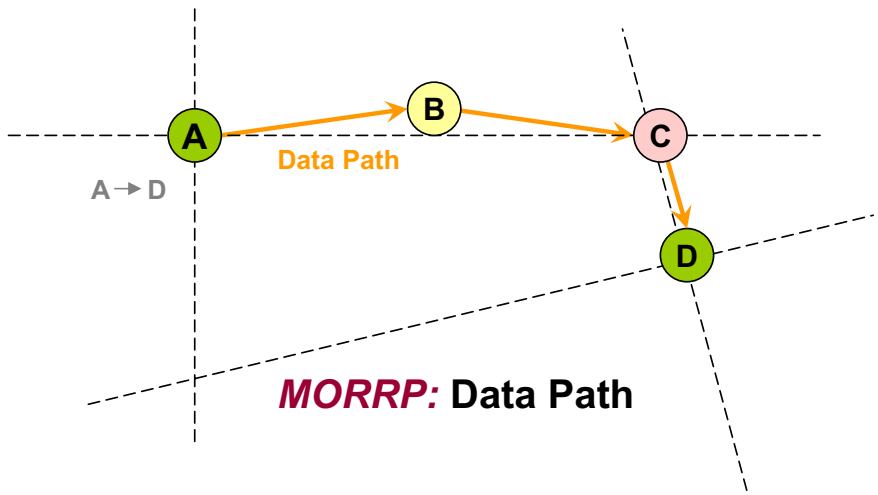


Figure 4.8: MORRP Data Delivery - Data path after route generation

Table 4.2: MORRP Comparison of Probability of Rendezvous vs. Velocity

Mobility Speeds:	10 m/s	20 m/s	30 m/s
After 1 sec	98.5%	96.9%	94.3%
After 4 sec	91.9%	81.9%	74.6%

ble source and destination orientations and generate the intersection parallelograms. Then, we count the number of nodes within area A and divide by area B to get the probability that sending a RREQ will intercept an announcement given a set mobility speed, transmission radius, and time after announcement sent. In our numerical analysis, we tried to mimic our NS simulations so we normalized the 250m transmission radius to 1m and corresponding mobility speeds. For our paper, we only considered square topologies.

Table 4.2 shows our results for probability of announcement/RREQ rendezvous for various mobility speeds after waiting 1 and 4 seconds after announcement packets were sent. As expected, the results showed decreasing, yet high, intersect probability with higher mobility and longer wait time. This is because information becomes more dispersed over time and higher mobility. Our analysis gives only a partial view of reach probability as actual data will still need to hit the rendezvous region

Algorithm 10 MORRP Data Delivery

```
ForwardData(p)
 1: if p_dest ∈ NeighborList then
 2:   // Destination is my 1 hop neighbor:
 3:   SendData(p, next_hop)
 4: else if p_dest ∈ NearFieldDRT then
 5:   // Destination is my Near Field DRT:
 6:   j ←GetInterfaceIDfromNFDRT()
 7:   Φ ←GetRandomNeighbor(j)
 8:   p_rend_node ← null
 9:   SendData(p, Φ)
10: else
11:   if p_src = ID then
12:     // I'm the source
13:     ForwardDataSrc(p)
14:   else
15:     // I'm just a forwarder
16:     if p_rend_node ≠ NULL then
17:       // data packet has rendezvous node state set in packet header
18:       ForwardDataWithRendezvous(p)
19:     else
20:       // data packet has no rendezvous node
21:       ForwardDataNoRendezvous(p)
22:     end if
23:   end if
24: end if
```

and destination region for successful packet delivery in mobile environments. And although not complete in describing the whole protocol, it gives a high-order view of the overall intersect behavior and shows that even with high mobility, the probability of finding a rendezvous point is relatively high. In the actual protocol, not all nodes require this far-field operation because some are close enough to the source to utilize the near-field DRT. Additionally, RREQs are sent upon need and can be anywhere between the announcement interval and node mobility velocity is not constant throughout the network. All these factors merit additional simulations to fully understand the inner-workings of the protocol which we describe in the following section.

Algorithm 11 Data Delivery - Packet Source

`ForwardDataSrc(`p`)`

1: **if** $p_{dest} \in FarFieldDRT$ **then**
2: // Destination is my Far Field DRT:
3: $j \leftarrow$`GetInterfaceIDfromFFDRT()`
4: $\Phi \leftarrow$`GetRandomNeighbor(`j`)`
5: $p_{rend_node} \leftarrow null$
6: `SendData(`p, Φ`)`
7: **else if** $p_{dest} \ni DestRendTable$ **then**
8: // Destination not in Dest-Rendezvous Table
9: `BufferData(p)`
10: `ForwardRREQPacket(`p`)`
11: **else**
12: $dre_{rend_node} \leftarrow$ `GetRendNode(`p_{dest}`)`
13: **if** $dre_{rend_node} \neq null$ **then**
14: **if** $dre_{rend_node} \in NeighborList$ **then**
15: // Rendezvous node is my 1 hop neighbor:
16: `SendData(`p, dre_{rend_node}`)`
17: **else if** $dre_{rend_node} \in NearFieldDRT$ **then**
18: // Rendezvous node is my Near Field DRT:
19: $j \leftarrow$`GetInterfaceIDfromNFDRT()`
20: $\Phi \leftarrow$`GetRandomNeighbor(`j`)`
21: $p_{rend_node} \leftarrow dre_{rend_node}$
22: `SendData(`p, Φ`)`
23: **else if** $dre_{rend_node} \in FarFieldDRT$ **then**
24: // Rendezvous node is my Far Field DRT:
25: $j \leftarrow$`GetInterfaceIDfromFFDRT()`
26: $\Phi \leftarrow$`GetRandomNeighbor(`j`)`
27: $p_{rend_node} \leftarrow dre_{rend_node}$
28: `SendData(`p, Φ`)`
29: **else**
30: // Stale route
31: `BufferData(p)`
32: `ForwardRREQPacket(`p`)`
33: **end if**
34: **else**
35: // Destination and Rendezvous definitely not known
36: `BufferData(p)`
37: `ForwardRREQPacket(`p`)`
38: **end if**
39: **end if**

Algorithm 12 Data Delivery - Forward with Rendezvous Node

`ForwardDataWithRendezvous(`p`)`

1: **if** $p_{rend_node} = ID$ **then**
2: // We are the rendezvous node. Check if dest is in far field DRT
3: // (Whether dest is in neighbor list and near field DRT already checked)
4: $p_{rend_node} \leftarrow null$
5: **if** $p_{dest} \in FarFieldDRT$ **then**
6: $j \leftarrow$`GetInterfaceIDfromFFDRT()`
7: $\Phi \leftarrow$`GetRandomNeighbor(`j`)`
8: `SendData(`p, Φ`)`
9: **end if**
10: **else**
11: **if** $p_{dest} \in FarFieldDRT$ **then**
12: // Destination in far field DRT
13: $j \leftarrow$`GetInterfaceIDfromFFDRT()`
14: $\Phi \leftarrow$`GetRandomNeighbor(`j`)`
15: `SendData(`p, Φ`)`
16: **else if** $p_{rend_node} \in NeighborList$ **then**
17: // Rendezvous node is my 1 hop neighbor:
18: `SendData(`$p, next_hop$`)`
19: **else if** $p_{rend_node} \in NearFieldDRT$ **then**
20: // Rendezvous node is my Near Field DRT:
21: $j \leftarrow$`GetInterfaceIDfromNFDRT()`
22: $\Phi \leftarrow$`GetRandomNeighbor(`j`)`
23: `SendData(`p, Φ`)`
24: **else if** $p_{rend_node} \in FarFieldDRT$ **then**
25: // Destination is my Far Field DRT:
26: $j \leftarrow$`GetInterfaceIDfromFFDRT()`
27: $\Phi \leftarrow$`GetRandomNeighbor(`j`)`
28: `SendData(`p, Φ`)`
29: **else**
30: // Just keep forward in opposite direction
31: $\alpha \leftarrow$ `NumInterfaces`
32: $j \leftarrow (p \rightarrow Recv_Int_Id)\ j \leftarrow ((j + \alpha/2)\%\alpha)$
33: $\Phi \leftarrow$`GetRandomNeighbor(`j`)`
34: `SendData(`p, Φ`)`
35: **end if**
36: **end if**

Algorithm 13 Data Delivery - Forward without Rendezvous Node

```
ForwardDataNoRendezvous(p)
1: // (Whether dest is in neighbor list and near field DRT already checked)
2: if p_dest ∈ FarFieldDRT then
3:    // Destination in far field DRT
4:    j ←GetInterfaceIDfromFFDRT()
5:    Φ ←GetRandomNeighbor(j)
6:    SendData(p, Φ)
7: else
8:    // Just keep forward in opposite direction
9:    α ← NumInterfaces
10:   j ← (p → Recv_Int_Id) j ← ((j + α/2)%α)
11:   Φ ←GetRandomNeighbor(j)
12:   SendData(p, Φ)
13: end if
```

4.5 Performance Evaluation

In this section, we provide performance evaluations of MORRP under various parameters and against several proactive, reactive, and position-based routing protocols with one omni-directional interface and several directional interfaces. The simulations were performed using NS2 [28], with nodes using the standard IEEE 802.11 MAC and a 250m antenna range (NS2 default). RTS/CTS is turned off because this is standard practice in actual deployment. Each node moves using the random waypoint mobility model with a node pause time of 5 seconds in $1300 \times 1300m^2$ and $2000 \times 2000m^2$ areas. All simulations were averaged over 2 runs of 5 different randomly generated flat topologies (total 10 trials) and the 95% confidence intervals of the runs plotted. Table 4.3 gives our default simulation parameters

MORRP and ORRP were configured using n interfaces (divisible by 4) with each interface having a beam-width of $360/n$ degrees and announcement and RREQ packet TTL set to 10 hops. Announcement packets were sent every 4 seconds. We choose 30 hash functions and a bloom filter size of 16000 bits for simulations with MORRP to ensure minimum overlap of bits with 100 or so nodes and employ no bloom filter compression. The exploration of optimal hash function sizes to ensure minimal bit collisions are beyond the scope of the paper and more information can be found in [129].

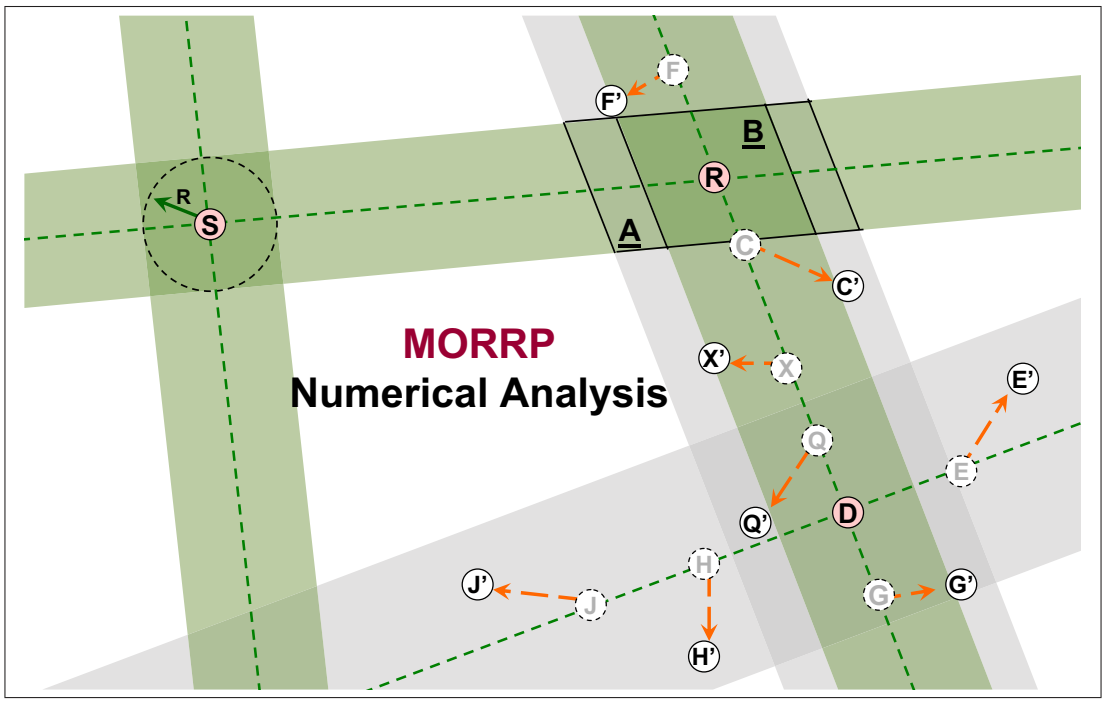


Figure 4.9: MORRP Reachability Numerical Analysis Calculation

For *reactive* routing protocols like DSR and AODV which require no periodic updates and position-based protocols like GPSR with GLS as the location service, the standard NS2 defaults were used. For OLSR, we set the TC update interval to 4 seconds to match ORRP and MORRP announcement intervals. For all simulations, a hello interval of one second was used and MAC layer feedback employed for all the routing protocols. A potential future extension is MORRP with routing metrics. Traffic patterns varied for each test and are described in each subsection. Implementations and defaults for GPSR/GLS and OLSR can be found at [137] and [138] respectively.

In order to explore whether MORRP gains were merely from capacity gains with multiple directional antennas or *actual* design improvements, we modified AODV and OLSR implementations to support multiple directional interfaces in the same way as MORRP and ORRP. Since AODV and OLSR rely on omnidirectional broadcast to disseminate information, sending out all interfaces simulates the behavior of AODV and OLSR broadcasts. Transmitting data packets, however, require only one interface to be active at a time freeing the medium and other interfaces for

Table 4.3: MORRP Default Simulation Parameters

Parameter	Values
Trans. Radius / # Interfaces	250m / 8 Directional Interfaces
Topology Boundaries	1300m x 1300m, 2000m x 2000m
Queue Length / Simulation Time	250 / 70s
Announcement Interval / Mobility (m/s)	4.0s / RWP 0m/s - 30m/s
DRT Update Interval	2.0s
Distance Decay Factor (D_d)	0.7 (fraction of bits dropped per hop)
Time Decay Factor (D_t)	0.3 (fraction of bits dropped per sec)
Time Decay Interval (D_i)	0.5s
# of BF Hash Funcs / BF Size	30 / 16000 bits
NF Threshold / FF Threshold	6 bits / 6 bits
Spread Decay Ratio (s_{ratio})	0.5
CBR Packet Size / Send Rate	512 bytes / 2Kbps

other nodes to use.

Additionally, in all the simulations, all routing protocols utilized a perfect ARP where ARP requests were suppressed and hardware addresses to next hop nodes automatically mapped. It was necessary to make these changes because ARP is not optimized for high mobility with multiple directional interfaces. These inefficiencies provided extra bottlenecks whereby the actual ability to transmit packets was limited not by the routing protocol but ARP request flooding.

The performance evaluations for MORRP are broken up into two major sections: *standalone evaluations* and *comparison evaluations*. Standalone evaluations deal purely with MORRP and adjust several knobs to understand how each plays a part in the protocol. Comparison evaluations take MORRP and evaluate it against several proactive, reactive, and position-based routing protocols. In the following subsections, we will provide metrics and evaluation conditions for both the standalone and comparative evaluations as well as the results from our simulations with respect to each condition.

4.5.1 Standalone Performance Evaluations

Standalone evaluations deal solely with MORRP under various conditions. In this section, we outline the metrics and the purpose and specifics of each conditions evaluated. The metrics used for the *standalone evaluations* are as follows:

- **Reach Probability** - In MANETs, reachability is the critical metric to evaluate routing protocols. This is simply because latency, goodput, and other traditional routing metrics are void if nodes are unreachable. MORRP relies on paths formed by the the intersection of announcement and route request paths. Unfortunately, in highly mobile environments, these paths are not easily maintained. MORRP attempts to address this issue by expanding the notion of rendezvous points to rendezvous regions and shift directions to forward packets based on local movement information. It is therefore interesting to see whether this technique achieves higher reach probability. We evaluate reach probability by sending only a few (around 1 or 2) packets from all nodes to all nodes ensuring little to no congestion drops and collecting the number of CBR packets received as compared to the number sent. It is important to understand that this metric is different than *data delivery success* which is utilized and described in the *comparison evaluations*. We will show that MORRP has fairly high (∼93+%) reach even in highly mobile environments (nodes moving at a maximum speed of 30m/s). When the network size becomes larger and sparser, MORRP's reach drops to 83% which still represents fairly high reach. Under larger and medium-dense networks, MORRP reach is approximately 89%.

- **Far Field and Near Field DRT Usage** - MORRP relies on near-field DRTs to route packets to neighbors within a few hops away. It does this by maintaining a "region/field of influence" where all nodes within the region have *some* information about the position of the source. This information becomes less and less accurate as the number of hops from the source increases until it is indistinguishable from "noise". Nodes that are beyond this distance threshold rely on the far-field DRT to build source to rendezvous and rendezvous to destination paths. It becomes interesting, therefore, to examine how much of the

data transmissions rely on the near-field and how much relies on the far-field. Network density, size, and velocity all have important roles in determining this usage. To evaluate near and far-field DRT usage, we simply track how many of the packets find neighbors within their near field and their far field.

To evaluate the metrics listed above for our standalone analysis, we utilized a simple *traffic pattern*: First, we allowed MORRP to perform its messaging for about 10 seconds to ensure states are properly seeded network-wide. Then, we send 512 byte CBR packets from all nodes to all nodes at a rate of 2Kbps for 2 seconds with thee start times of each of these connections varied between 10 seconds after the start of the simulations and 60 seconds. Since the simulations went on only for 70 seconds, it provided a good 10 second buffer for rogue packets to either be dropped due to TTL or reach the destination. We limited the amount of data sent to ensure no packet is dropped due to medium saturation or excessive collisions. Table 4.4 lists the default traffic pattern used in our standalone simulations of MORRP.

Table 4.4: MORRP Standalone Sims - Default Traffic Pattern Information

Parameter	Values
Number of CBR Connections	All-to-all
CBR Packet Size	512 Bytes
CBR Transmission Rate	2Kbps
CBR Transmission Duration	2.0 seconds
CBR Start Time Range	10.0 - 60.0 seconds randomly generated

In the standalone analysis, we evaluate each of the metrics above under varying conditions. These conditions and accompanying explanations of why they are important are listed below:

- **Varying Time Decay Factor (D_t)** - In MORRP, the probability of a node being in a specific direction from the source becomes less certain with time. The time decay factor determines how many bits of the decaying bloom filter are to be dropped per time decay interval. The positioning of nodes with

respect to the source are refreshed everytime a DRT update is received. With little to no bits decayed with respect to time (low D_t), updates might overlap in bits with previous entries resulting in false-positives in determining which direction to send packets to reach a node. On the otherhand, with too many bits decayed with respect to time and infrequent updates (high D_t), the effect of the DRT will be limited as only 1-hop nodes are utilized. In our simulations, we vary the time decay factor from 0.1 to 1, resulting in a drop of 10% to 100% of the bits every time decay interval.

- **Varying Distance Decay Factor (D_d)** - The distance decay factor determines the range of the "region/field of influence". Larger distance decay factors essentially mean that more bits are dropped per hop from the source and as a result, only next hop information is used. A smaller distance decay factor means that information about a node's position will be transmitted further (less bits dropped per hop). Although this can be desirable in low mobility situations, under high mobility, this can result in confusing directions as these bits are not dropped quickly enough. In our simulations, we vary the distance decay factor from 0.1 to 1, resulting in a drop of 10% to 100% of the bits every hop from the source.

- **Varying Near and Far Field Threshold Bits ($thresh/ff_thresh$)** - Because bits from one ID hashed into a bloom filter might overlap with bits from other IDs, it becomes necessary to distinguish actual information about the direction of a node vs. "leftover"

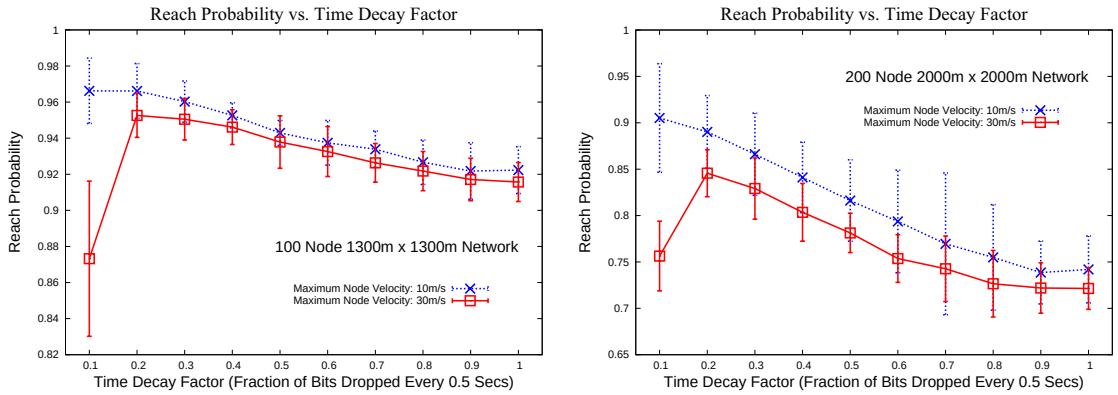
- **Varying Spread Ratio (s_ratio)** - Spread ratio determines how much of the bits are decayed using spread decay and how much are dropped using time decay. A low spread ratio means that all the bits that are decayed are dropped via time decay. This indicates that there is no "shifting" of send directions based on *local* node movement. A high spread ratio means that all the bits selected per time decay interval are spread to other interfaces. A spread ratio of 1.0 indicates that no bits are dropped and can cause confusion in terms of which direction to send the data to reach a destination.

In the following subsections, we will present and discuss our results from the standalone performance evaluations.

4.5.1.1 Effect of Time Decay Factor in Mobile Environments

As section 4.2.1 mentions, knowing how many bits of the bloom filter to "decay" per time interval will affect reachability in highly mobile situations. The smaller the time decay factor (fraction of bits dropped per time interval), the *less* bits are decayed per interface resulting in bit accumulation and misinformation in mobile situations. In otherwords, nodes will think another node is in a specific direction when it has long since veered to a different path. It is therefore important to explore how the time decay factor affects reachability and percentage of packets routed through the near-field DRT vs. the far-field DRT. In our simulations, we fixed the default values given in Table 4.3 while varying the time decay factor from 0.1 to 1.0. With our default time decay interval set to 0.5 seconds, a time decay factor of 0.1 means that 10% of the bits are allocated for spread and time decay every 0.5 seconds. Thus, if the spread decay factor is 0.5, then 5% of the bits will be dropped and 5% of the bits will be "spread" to other interfaces. A time decay factor of 1.0 means that 100% of the bits are allocated for intra-node decay (spread and time decay). Again, if the spread decay is set to 0.5 as it is in the default simulation parameters, half of the 100% of the bits will be used for time decay and dropped (changed to 0) and half of the bits (50%) will be used for spread decay and shifted to other interfaces every 0.5 seconds. Figures 4.10 and 4.11 shows our results for a 100 node ($1300 \times 1300m^2$) and 200 node ($2000 \times 2000m^2$) network.

As expected, figure 4.10 shows that for high mobility, decreasing the time decay factor (dropping *less* bits per time interval) results in lower reach probability due to misinformation and bit accumulation in both medium-sized topologies ($1300 \times 1300m^2$) and large-scale topologies ($2000 \times 2000m^2$). On the opposite spectrum, having too high of a time decay factor, thereby dropping a high number of bits per time interval also leads to less reach probability. This can be explained by the *far-field* DRT usage graph (figure 4.11). As the time decay factor is low, the majority of data packets will be utilizing the far-field DRT to find a path because the near-field



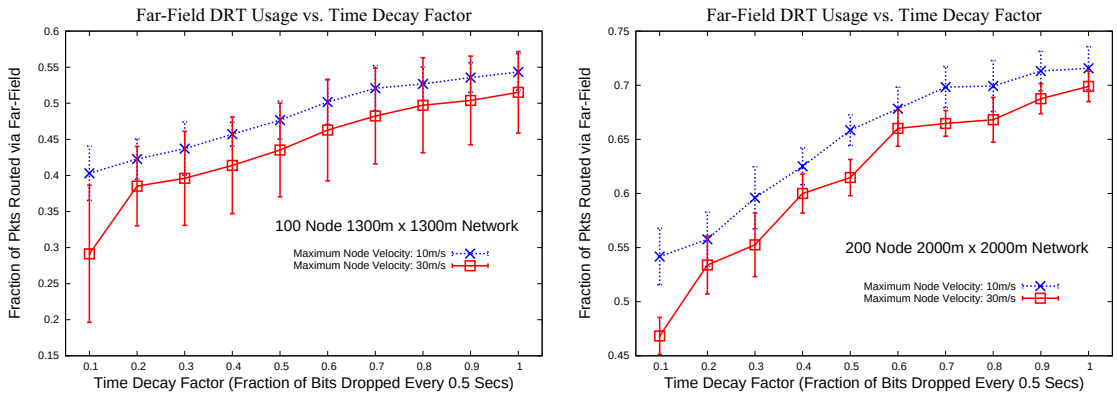
(a) (b)

Figure 4.10: Effect of time decay factor on MORRP reachability with various maximum mobility speeds on various topology sizes. For higher mobilities, an optimal time decay factor is necessary to ensure high reach. Decaying too much forces nodes to rely too heavily on far-field which is updated less frequently.

DRT coverage region will decay rather quickly. Because far-field DRT information is updated *less* frequently, too much reliance on it can yield inaccurate results. An optimal decay factor must be selected, therefore, to ensure high delivery success and a fair usage of both far-field and near-field DRTs. One interesting note is with the nodes moving at a maximum velocity of 10m/s case. Because our bloom filter union technique performs unions only on "unique" bits, repeated updates without decay has no affect on delivery success as long as nodes remain in same general direction. With low mobility, nodes do not move far off its intended direction quickly so DRT updates can quickly correct for changes in trajectory.

4.5.1.2 Effect of Distance Decay Factor in Mobile Environments

A key element in determining how much position information a node has about its neighbors beyond the next-hop neighbors is the *distance decay factor*. As the distance decay factor (the fraction of bits dropped per hop) increases, each node will have *less* information about nodes beyond their next hop neighbors. Therefore, while a source might have *some* information about a node three hops away with a low distance decay factor, by increasing this value, the radius of information decreases

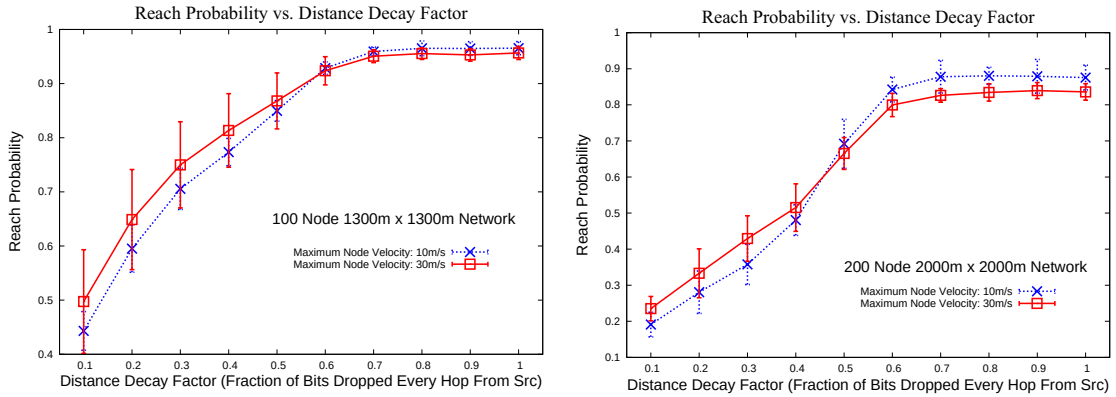


(a) (b)

Figure 4.11: Effect of time decay factor on MORRP far-field usage with various maximum mobility speeds on various topology sizes. As time decay increases, near-field DRTs decay information quickly thereby relying on far-field information.

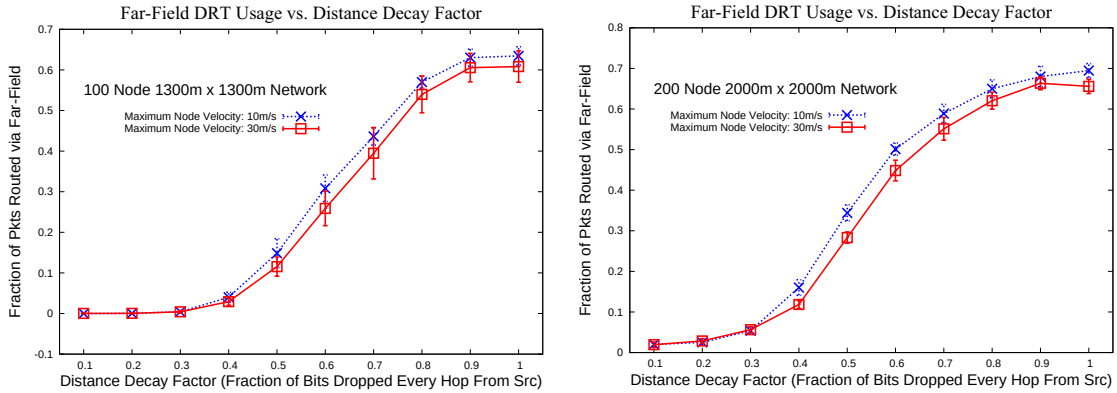
drastically. At the same time, having a low distance decay factor might give false impressions of neighbor locations in highly mobile networks. We will use the same metrics as the previous section: evaluating reach probability and far-field vs. near-field DRT usage to route packets under various mobility conditions and distance decay factors. In our simulations, we fixed the default values given in Table 4.3 while varying the distance decay factor from 0.1 to 1.0. A distance decay factor of 0.1 corresponds to 10% of the bits dropped per hop beyond the first hop while a distance decay factor of 1.0 corresponds to 100% of the bits dropped beyond the first hop.

Our results in figures 4.12 and 4.13 show that there is a gradual increase in reach probability when the distance decay factor goes from 0.1 to 0.6 and then the reach plateaus out. The low reach probability when the distance decay is lower results from saturation of bits to multiple interfaces resulting in confusing paths chosen. While the reach probability plateau's at a distance decay factor of 0.6, the far-field dependence graph in figure 4.13 shows that there is still a gradual shift from using near-field DRT to route information at the source to far-field DRT dependence. Although the far-field DRT is updated less frequently, continual dependence on the far-field means that nodes are constantly requesting for new routes. Coupled with



(a) (b)

Figure 4.12: Effect of distance decay factor on MORRP reachability with various maximum mobility speeds on various topology sizes.



(a) (b)

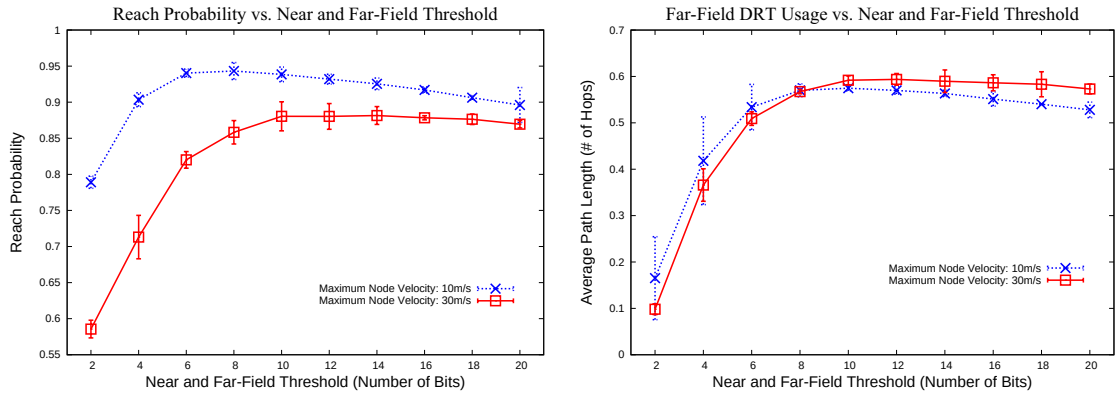
Figure 4.13: Effect of distance decay factor on MORRP far-field usage with various maximum mobility speeds on various topology sizes.

the direction correction and a good time decay, a high degree of reach even in highly mobile situations is expected.

4.5.1.3 Effect of Threshold in Mobile Environments

In this section, we adjust the knob of near-field and far-field bit threshold. As the threshold increases, the size of the "fields/regions of influence" for each decreases because there needs to be more bits in the reverse hash match for IDs in order to constitute a positive match. It is expected that as the threshold for the near-field

increases to greater than or equal to the number of hash functions, each node will only have information about itself (since the number of hash functions constitute "full information" about a neighbor). Figures 4.14(a) and 4.14(b) shows the results of changing threshold for both near-field and far-field DRTs. In our simulations, we assumed that the threshold for near-field and far-field are equal and adjusting both knobs independently are beyond the scope of this thesis.



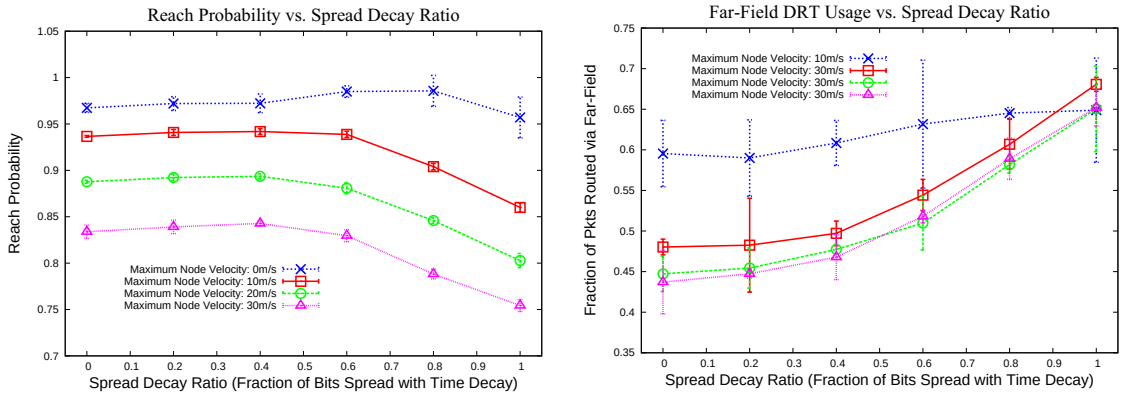
(a) (b)

Figure 4.14: Effect of near and far-field threshold on MORRP reachability and far-field DRT usage with various maximum mobility speeds on various topology sizes.

As can be seen in figure 4.14, when the threshold is low, reach probability is low. This is due to a lot more confusion about path selection because of bit collisions and slow time decay. In short, bit collisions cause confusing "false positives" and results in poor path choices, low reachability, and greater dependence on the far-field DRT. As threshold increases, it approaches a point where each node has information about itself and its 1 hop neighbors (since 1 hop neighbors do not decay their node own ID hash when sending their DRT) resulting in high reachability. This point occurs at different times depending on the maximum node velocity. For example, it was seen that reach probability peaks at roughly 6 bits which is roughly 20% of the number of hash functions used for nodes moving at a maximum velocity of 10m/s while it peaks at 10 bits which is roughly 33% of the number of hash functions for networks with nodes moving at a maximum velocity of 30m/s.

4.5.1.4 Effect of Spread Ratio in Mobile Environments

Although section 4.2.1 illustrate a heuristic for dropping bits and spreading bits over time, it was assumed that an equal number of bits were allocated to have the *possibility* of dropping and spreading (note this is *different* than allocating equal bits to drop and spread). It is interesting, therefore, to see what happens when we vary the ratio between bits allocated for the *possibility* of being dropped and spread. In this section, we vary the spread ratio (s_ratio) between 0.0 and 1.0 while keeping all other factors constant. Figures 4.15 shows our plot of when spread decay is applied to *both* near-field and far-field DRTs



(a) (b)

Figure 4.15: Effect of far-field spread ratio on MORRP reachability and far-field DRT usage with various maximum mobility speeds on various maximum node speeds. Spread decay is onyl used for far-field DRTs. Under average network density, spread decay helps reach probability by roughly 1-2% when applied at a ratio of 0.6.

Because near-field DRT is updated rather frequently compared to the far-field DRT (1:4 ratio), it was suspected that the far-field DRT would benefit much more from spread decaying. Figure 4.15 shows the reach probability and far-field DRT dependence vs. spread decay ratio for various mobility under a node density of about 10 neighbors per node with spread decay removed from the near-field DRT. Our results show that spread decay increases reach probability by 1-2% when applied at a ratio of about 0.6. This makes sense as additional increase of spread decay would lead to less bits being dropped yielding in additional confusing routes from

the far-field DRT.

4.5.1.5 Summary of Standalone Evaluation Results

Below we summarize our findings in our standalone MORRP evaluations:

- For highly mobile situations, the time decay factor must be increased such that more nodes are dropped per time interval. The optimal time decay factor should be a factor of the node mobility as networks with low mobility show that low time decay factors are beneficial.
- Under all mobility cases, a distance decay equal of 0.6 or 0.7 (dropping about 60% - 70% of bits per hop past the next hop) is optimal.
- Optimal near and far-field threshold is about 20% to 33% of the number of hash functions depending on mobility speed. Higher mobility requires higher threshold for positive match.
- Optimizing spread decay yields only about 1.5% - 2% gain in reach and more exploration into different heuristics to do spread decay is needed.

4.5.2 Comparative Performance Evaluations

For our *comparative evaluations*, we choose to evaluate MORRP against several proactive, reactive, hybrid, and position-based protocols: AODV [11], OLSR [34], GPSR with GLS [5, 6] and ORRP [1] under varying conditions of mobility. As described earlier, AODV, like all *reactive* topology-based routing protocols, finds routes "on-demand" and trades smaller overhead with few connections for higher end-to-end latency. OLSR, much like other proactive topology-based routing protocols, periodically share routing information with neighbors trades optimized paths for additive bandwidth consumption. GPSR, like all position-based routing protocols, boasts seemingly low control overhead, but relies on location services such as GLS to provide end-to-end position information. These location services incur additional overhead and still do not take into consideration position information usually garnered through localization techniques or devices such as GPS.

ORRP is a hybrid proactive and reactive protocol that removes the need for positioning information to route packets geographically using directional communications methods. We saw in chapter 3 that ORRP utilizes the medium much more efficiently while providing high connectivity and low end-to-end path stretch and latency. Unfortunately, ORRP fails under conditions of high mobility because lines are difficult to maintain. MORRP, by contrast, takes the concept of using directionality to find path intersections, but expands the notion of rendezvous points to rendezvous regions. It also shifts directions of send based on its own local movement information. It is interesting, therefore, to see what kind of gains we get by utilizing directionality in a mobile adhoc environment.

What is even more interesting, however, is understanding whether the gains that come from MORRP come simply as gains from using directional antennas or by an improved protocol design. To test this effect, we modified OLSR and AODV to support multiple directional antennas in much the same way as ORRP and MORRP and gave each protocol the same assumption we used for MORRP (i.e. each node knows its 1 hop neighbors and the interface mapped to each neighbor). Since OLSR and AODV rely on omnidirectional transmissions to disseminate link state information or perform route requests, everytime link state packets or RREQ packets are sent, each of the interfaces send out a copy of the packet to its neighbors. The only gains that theoretically come from OLSR and AODV using directional antennas are from actual data transmission and RREP packets which only utilize one interface at a time. Our *comparative evaluations* examine similar metrics as in the *standalone evaluations*, but under more practical environments and against a wide variety of proactive, reactive, and position-based protocols.

- **Packet Delivery Success** - Packet delivery success focuses on how well a protocol handles network load as more and more nodes attempt to communicate simultaneously. It is the *key* metric in MANETs because if packets are not reaching intended destinations, then all other metrics are mute. With high mobility, it becomes increasingly difficult to ensure end-to-end delivery success. The issue is complicated further by efficient medium usage with increased need to send control packets to maintain routes. It is expected that

using a directional form of communication which by default frees the medium up for multiple simultaneous transmissions should lead to higher packet delivery success under the presence of high load. Furthermore, because MORRP only selects 4 interface directions and only chooses a single next-hop in each of those directions to send for both dissemination and route request, it is expected (and we will show) that MORRP will utilize the medium even more efficiently than omnidirectional transmissions-designed protocols modified for use with directional antennas.

- **Control Packet Overhead (Bytes)** - With high mobility, state information becomes stale quickly and often needs to be refreshed at a shortened interval. With the increased dissemination of state information, either proactively or reactively, the network becomes quickly saturated with control packets, preventing data from being transmitted. In short, messaging overhead costs become significant. It is interesting, therefore, to measure how much of the medium is being used network-wide for control overheads. We record the sent control packet *bytes* as many protocols combine state information into single packets and show that MORRP sends much fewer control packets than all the other protocols we compared against.

- **Average End-to-End Path Stretch** - Path stretch becomes increasingly difficult to optimize with high mobility because links change quickly. As the medium becomes saturated with control messaging overheads, however, optimal paths become harder and harder to calculate even for proactive routing protocols like OLSR. It is interesting to see how MORRP fares in terms of end-to-end path stretch in highly mobile environments. We will show that MORRP performs better than AODV, GPSR/GLS, and ORRP but lags behind OLSR. We measure path stretch by taking the actual number of hops traversed by each CBR packet and dividing it by the calculated shortest path.

- **Average End-to-End Latency** - Latency is the amount of time for a packet to travel from source to destination. In highly mobile environments, latency is important because routes become stale quickly. The longer it takes for data to

travel, the higher possibility the next-hop paths specified will be inaccurate. If routes need to be repaired because of link breakages due to mobility, latency is also increased. We evaluate latency by taking the difference between the received time and the send time of each CBR packet.

- **Aggregate Network Goodput** - With the more efficient usage of the medium and limiting of control packet flood even with using directional forms of communications, it is fundamental to understand what kind of gain in goodput we can achieve. Goodput gain in highly mobile environments come from successfully packet delivery and efficient usage of the medium. In our simulations, we measure aggregate network goodput by sending CBR packets from all nodes to all nodes simultaneously for 20 seconds, slowly increase the rate and suming the number of bits of data received network-wide. The reason why we saturate the medium with increased CBR packet transmit rate is to ensure we are looking at medium usage instead of simply the percentage of successful packets delivered (as MORRP is expected to have higher goodput because it delivers more packets successfully). We will show that MORRP achieves higher aggregate network goodput than the other protocols.

In our comparative analysis, we evaluate each of the metrics above under varying conditions against traditional routing protocols like AODV, OLSR, GPSR with GLS with omnidirectional antennas as well as AODV and OLSR modified with directional antennas similar to ORRP and MORRP. The conditions for our experiments and accompanying explanations of why they are important are listed below:

- **Varying Maximum Node Mobility Speeds** - In highly mobile networks, links and end-to-end paths are hard to maintain. We are interested in seeing how robust MORRP is compared to other state-of-the-art routing protocols under various maximum node mobility speeds moving using random waypoint mobility (RWP) model. The scenarios were generated with the CMU scenario generator with maximum node mobilities between 10m/s and 30m/s in increments of 10m/s and a node pause-time of 5.0 seconds. We generated 5

scenarios under 2 different topology areas ($1300 \times 1300m^2$ and $2000 \times 2000m^2$) with both scenarios yielding an average neighbor density of 8-9 neighbors per node. It was important to see how MORRP and other protocols performed under smaller topologies and medium-sized topologies. Larger topologies were harder to simulate with NS2 because to keep the same node density (as the previous chapter showed that ORRP performs fairly weakly under sparse network environments), more nodes needed to be added. Computing resources became the limiting factor. For each simulation, 1000 random CBR sources and destinations pairs were selected and 512 byte packets transmitted for 5 seconds. It is expected that with the increase in network area size and mobility speed, less packets will be successfully delivered due to constant link changes and longer end-to-end paths. MORRP uses DRTs, weak-state, and probabilistic routing to forward packets and as such, it is expected to perform fairly well even in highly mobile situations. We show that MORRP still delivers relatively high packets even in the presence of high mobility.

- **Varying Network Densities** - Increasing network density becomes an interesting scenario to evaluate in MANETs because with the high mobility, traditional routing protocols will struggle to send more control packets to keep link information up-to-date. With increased density, more nodes are expected to share the medium and take turns transmitting which results in greater network congestion. We seek to understand how MORRP compares with the other protocols in managing the increase in network density. Of particular interest is seeing delivery success and goodput when protocols like AODV and OLSR are outfitted with directional antennas. To measure these results, we utilized the CMU scenario generator that is standard on NS2 [28] distributions to generate wireless topologies with a $1300 \times 1300m^2$ area with increasing node density. Each node was set to travel at a maximum velocity of 30m/s with a pause-time of 5.0 seconds. We then randomly choose 1000 source and destination pairs to send at constant bitrate of 2Kbps for 10 seconds each and measure the metrics described above. It is expected that as the node density increases, there will be much less successful packets delivered due to increased

messaging overheads.

- **Varying Rates of Transmissions** - To examine the capacity of the network even under high mobility, we slowly increase the rate of a fixed number of CBR transmissions until a "knee" in the goodput is observed. Although goodput in MANETs is often determined by delivery success, by flooding the network until capacity, we can adequately see whether the gains come simply by using directional antennas or by protocol design. In our simulations, we make connections from all nodes to all nodes at the same time and attempt to send data (512 byte packets) at an increasing (2Kb - 20Kb) bit rate for 20 seconds. We expect to see that by using directional antennas, a much higher goodput capacity can be achieved. Furthermore, we expect higher goodput to be achieved with MORRP because it uses directional antennas intelligently even in route request and state dissemination phases.

In the following subsections, we will present our results and discussion of the comparative performance evaluations. Default parameters for MORRP given in table 4.3 were used in all scenarios unless otherwise stated. We focus heavily on *reachability/delivery success* in all these scenarios because in mobile adhoc networks, reachability comes primary over throughput, latency, etc. The reason is because our results show that for high mobility and high load, even limited-flooding protocols like AODV, OLSR, and GPSR with GLS simply cannot deliver the majority of the packets (low reachability). GPSR with GLS performs well under low load, boasting high reachability, but our tests show very high end-to-end latency. Additionally, protocols like GPSR require special equipment such as GPS receivers for node localization which are often unavailable due to lack of "sky" access or power limitations.

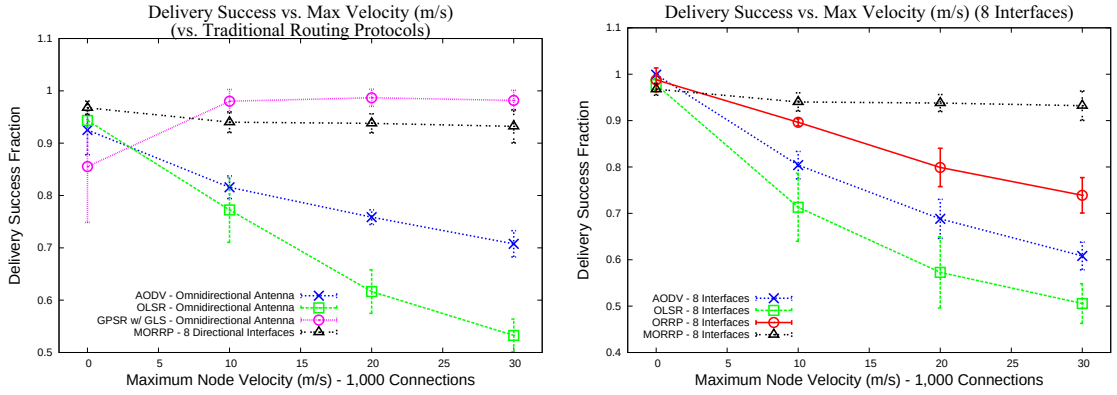
4.5.2.1 Effect of Increased Velocity

In this subsection, we evaluate the effect of increasing velocity on traditional routing protocols like AODV, OLSR, GPSR with GLS, and OLSR and compare it to MORRP, ORRP, and multi-interfaced versions of AODV and OLSR. To do this, we utilize various CBR traffic patterns under various node densities and network sizes and measure reachability, average path length, and average end-to-end latency for

each of the protocols. We shall see that under light load, position-based protocols like GPSR with GLS perform well. However, under load, reach drops considerably. Even with the modified interfaces, however, OLSR and AODV perform worse than MORRP in terms of reach for increased velocities.

In our initial simulations, we seek to understand how each protocol performs under light load with varying network sizes (both node density and topology area). For our simulations, default simulation parameters described in Table 4.3 were used for a 100 node $1300 \times 1300m^2$ and a 200 and 300 node $2000 \times 2000m^2$ topology. Because we did not want to load the network, only 1000 512 byte CBR packets from random source and destination pairs were sent at a rate of 2Kb for 5 seconds each. Each of the connections were started at a random time between 10 and 60 seconds. Figures 4.16 to 4.21 show our results for MORRP vs. traditional routing protocols with omnidirectional antennas and vs. traditional routing protocols modified with 8 directional interfaces.

4.5.2.2 Effect of Velocity on Medium-Sized, Lightly Loaded Networks



(a) (b)

Figure 4.16: Effect of velocity on packet delivery success for multiple routing protocols for 1,000 connections in a 100 node 1300m x 1300m network. MORRP maintains about 93% reach even in highly mobile situations.

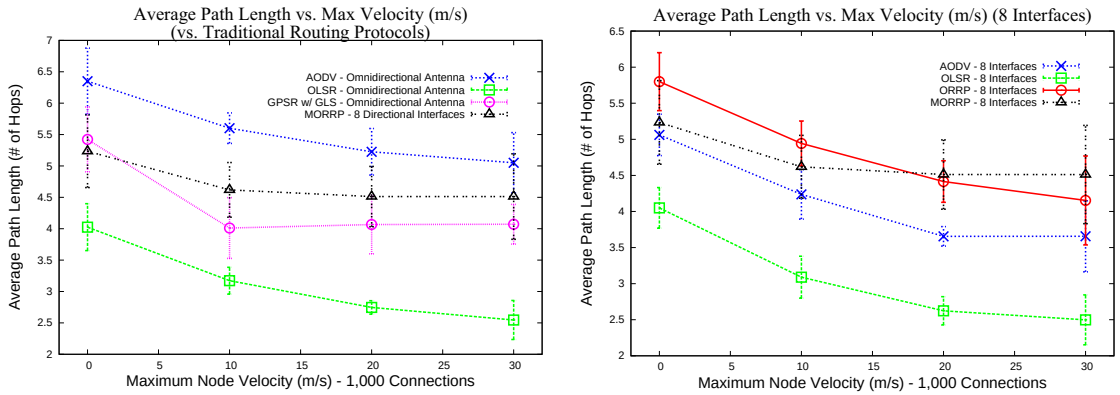
Figure 4.16 shows our results for a 100 node $1300 \times 1300m^2$ network with traditional routing protocols and traditional routing protocols modified with 8 directional

antennas. It can be seen that even under light load, as mobility increases, AODV and OLSR reach probability drops considerably. GPSR with GLS, however, retains high reach under light load. In fact, GPSR with GLS boasts higher reach with mobility than in the static fixed case. This is because although GLS periodically sends beacon information to neighbors, it does not send location updates periodically. The only time GLS sends location update packets is at the simulation startup or when a node moves some distance (in our case, the standard 250m corresponding to the antenna range) after the last update packet.

Under normal operation, GLS broadcasts location packets upon startup at relatively the same time. Because of contention, some of these packets do not arrive at their neighbors and because they are control/broadcast packets, they are not resent. Thus, if the location servers do not receive the location information on the first trial, they will never receive them again. This is by design in GLS. In non-mobile situations, as seen in our simulations, location information might not be successfully sent leading to low reach when GPSR queries for it. There are several reasons why we would be hesitant to choose GPSR with GLS as our routing strategy (or even a modified GPSR/GLS with directional antennas) despite its high reach probability in high mobility. These will be explained in section 4.5.2.4.

MORRP performs consistently well even under high mobility in this scenario, successfully delivering over 93% of the packets. AODV and OLSR modified with directional interfaces performs slightly worse than their omnidirectional counterparts most likely because of inefficiencies in the MAC layer with negotiating multiple interface collisions. The same drop-off occurs in both the omnidirectional case and the directional case. ORRP, as expected, drops in reachability because lines are hard to maintain.

Figure 4.17 describes our results under the same scenarios for average path length. It is interesting to note that AODV still chooses the worst paths even under light load. This is because AODV path selection depends on route replies which arrive first. If AODV RREQ packets rendezvous a node with a cached route for a destination, a reply will be sent quicker than a node that might find a shortest path. Until the 2nd RREP is received, AODV will forward using the first informa-



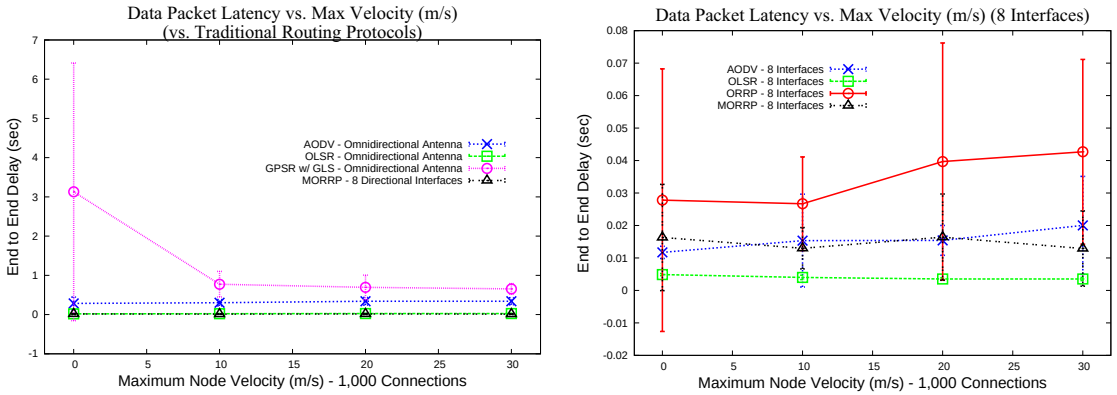
(a) (b)

Figure 4.17: Effect of velocity on end-to-end path length for multiple routing protocols for 1,000 connections in a 100 node 1300m x 1300m network.

tion found. This explains why chosen paths are often suboptimal. OLSR, being a link state protocol calculates perceived optimal paths resulting in low average path length. Its interesting to note that path lengths drop for both OLSR and AODV with increased mobility not because better paths are being chosen, but because the protocol is failing to reconcile longer paths. This is evident through the low reachability in figure 4.16(a).

AODV with multiple interfaces chooses better paths than AODV with an omnidirectional interface. This is presumably because multiple interfaces allows simultaneous transmissions in different directions resulting in quicker response times for RREP packets to arrive. ORRP paths chosen are the worst because of the two phased routing nature. MORRP performs better than ORRP in same delivery success scenarios because nodes that are near a destination help to "gravitate" the packet toward the destination.

Its interesting to note that out of all the routing protocols, GPSR with GLS incurs the most end-to-end delay. This is due to waiting on positioning information with each data packet. Although AODV (and its multi-interface counterparts) incur similar delay as MORRP, it is important to note that AODV drops over 40% of the packets in highly mobile situations. These packets are usually packets from longer paths as figure 4.17 shows and thus incur higher latency. Thus, our latency graph



(a) (b)

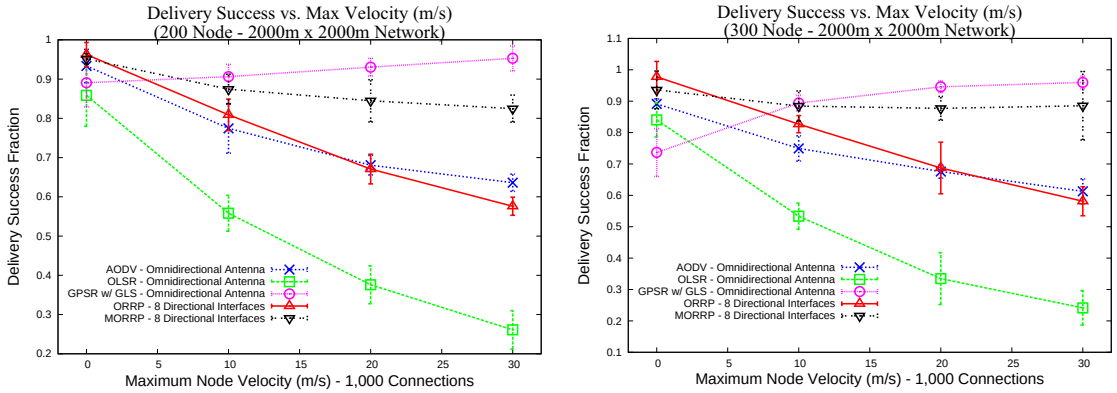
Figure 4.18: Effect of velocity on end-to-end latency for multiple routing protocols for 1,000 connections in a 100 node 1300m x 1300m network.

for AODV gives a false sense of AODV latency simply because longer latency paths are not counted due to unreach.

4.5.2.3 Effect of Velocity on Large-Sized, Lightly Loaded Networks

The previous subsection showed that AODV and OLSR fail under longer paths. Even for probabilistic routing protocols like MORRP, longer paths mean more predicting. It is interesting to see, therefore, how MORRP fairs with other routing protocols in the presence of longer paths on larger networks. Figures 4.19 to 4.21 show our results for larger $2000 \times 2000m^2$ networks with 200 and 300 nodes moving using random waypoint mobility. Because of the lightly-loaded network, we only test MORRP against the omnidirectional versions of AODV and OLSR since they showed higher reach than the multiple interfaced versions.

In both the 200 node and 300 node networks, we see a considerable drop in reach for OLSR in reach as the maximum mobility speed increases. This is due to requiring more hops to maintain link state information on the whole network. Farther nodes are updated less frequently due to lack of reach of link state packets and therefore routing is less accurate with longer paths. The same is also true of AODV but to a lesser degree because AODV is reactive and only requests routes on-demand. It is interesting to note that GPSR with GLS actually increases in reach



(a) (b)

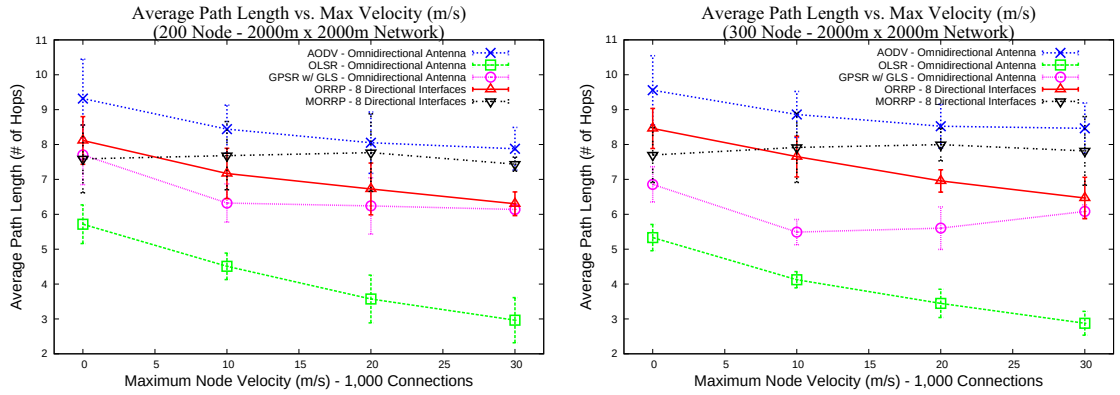
Figure 4.19: Effect of velocity on packet delivery success for multiple routing protocols for 1,000 connections in 200 node and 300 node 2000m x 2000m networks. MORRP provides about 83% reach for large-sized networks with medium sparsity and about 89% reach for large-sized networks that are denser.

with increased velocity. This is due to more frequent updates of location information as nodes move in and out of their transmission ranges.

ORRP again, performs even more poorly in the larger network than in the medium-sized network because longer path lines are harder to maintain in the presence of infrequent updates. MORRP drops in reach probability going from a medium-sized network to a larger-sized one (93% reach to 83% reach for 30m/s). This can be due to network density as the reach jumps up to close to 89% for the 300 node network vs. the 200 node network, but also because longer paths are harder to predict with a probabilistic routing protocol like MORRP. It could also indicate that our spread and decay factors are not being decayed properly to account for the extra distance. Further investigation into decay heuristics are required.

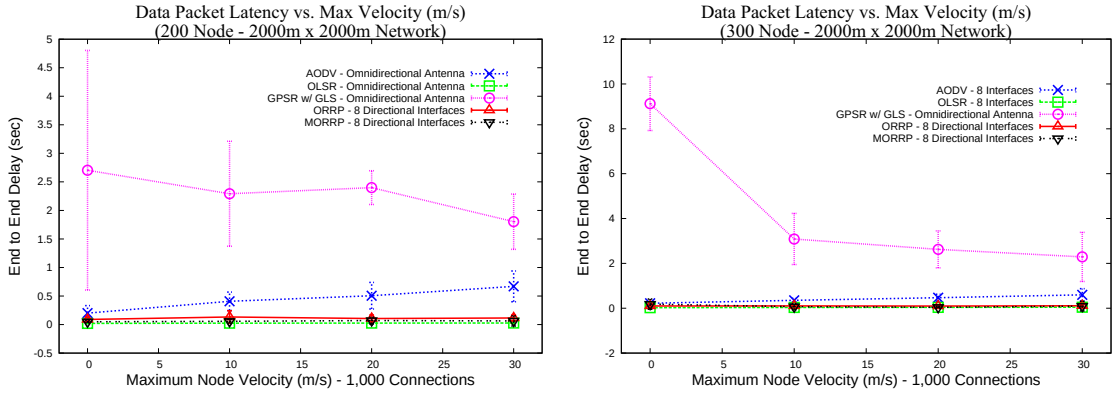
We see a similar trend in average path length compared to the medium-sized network and will defer to figure 4.20 above and explanations found in the previous section.

Our latency graphs in figure 4.21 paint an interesting story for GPSR with GLS. In both cases, we see an average of 40x the latency using GPSR with GLS than with MORRP. Figure 4.21(b) shows almost on average, 4 seconds to send a



(a) (b)

Figure 4.20: Effect of velocity on end-to-end path length for multiple routing protocols for 1,000 connections in 200 node and 300 node 2000m x 2000m networks.



(a) (b)

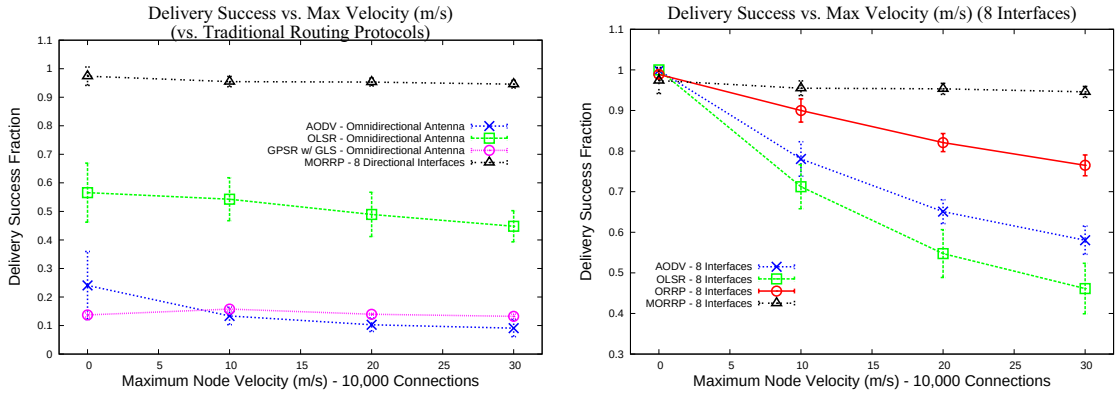
Figure 4.21: Effect of velocity on end-to-end latency for multiple routing protocols for 1,000 connections in 200 node and 300 node 2000m x 2000m networks. MORRP provides about 40x less latency than GPSR with GLS.

single packet. The measured latency comes from position querying and beaconing and position update intervals. Because positions are only updated when a node moves out of range, this can cause extra delays.

4.5.2.4 Effect of Velocity on Heavily Loaded Networks

In this subsection, we seek to understand how increased velocity affects each protocol under heavily loaded networks. To simulate the heavy load, we sent all-

to-all 512 byte CBR packets from random source and destination pairs were sent at a rate of 2Kb for 5 seconds each (approximately 10,000 connections). Each of the connections were started at a random time between 10 and 60 seconds. All simulations were performed under a 100 node, $1300 \times 1300 m^2$ network and figure 4.22 shows our results.



(a) (b)

Figure 4.22: Effect of velocity on packet delivery success for multiple routing protocols for 10,000 connections in a 100 node 1300m x 1300m network. Under load, all protocols using omnidirectional interfaces show signs of huge strain and fail to deliver most packets. Even with multiple interfaces, only MORRP successfully delivers the majority of packets, reporting 93% data delivery success.

It is clear that in conditions of high mobility and high load, MORRP with atleast 8 interfaces yields the highest reach probability (93%+) even under conditions of infrequent announcements sent (4 second intervals). As maximum velocity increases, AODV and OLSR fail because of stale routes and need to perform limited flood to find new paths leading to medium saturation. OLSR performance degrades fairly linearly under high mobility because unlike AODV, OLSR periodically broadcasts control packets rather than flooding "on-demand". Without increasing the topology control interval, however, routes become stale under infrequent updates.

GPSR with GLS performs consistently poor because when a node needs to send a packet, it issues a location request packet. Under conditions of all-to-all flows, every node in the network issues location discovery packets to every other

node in the network resulting in a huge overhead and causing many request packets to be dropped. As a result, the source cannot learn the location of the destination and after several unanswered requests, it assumes the destination is unreachable and drops the packet. In our simulations, we noticed that most of the dropped packets don't even leave the source node.

It is important to understand how much of the gains from figure 4.22(a) results from more efficient medium reuse due to directional interfaces vs. the gains coming from MORRP protocol design itself. We compare modified versions of AODV and OLSR to support multiple directional antennas with ORRP and MORRP with the results shown in figure 4.22(b). The modified versions of AODV and OLSR still broadcast (ie: send out all interfaces) when performing route requests or dissemination due to the protocol design and as such, we expect to see better performance with MORRP.

AODV with 8 directional interfaces shows significant improvement in delivery success vs. the traditional AODV due to the directional interfaces causing less interference in data delivery. The primary gains come from utilizing multiple interfaces as AODV is a reactive protocol and sends out route request packets on-demand. OLSR, on the other hand, saw only gains vs. the single interface OLSR in the stationary and low mobility case. This is because under high mobility, the limiting factor was not the medium but simply because the protocol could not find routes easily under periodic broadcasts. Being a proactive routing protocol, the only way to ensure accurate routes is to lower the hello and TC dissemination interval which cause more problems in heavily loaded environments. ORRP delivery success drops with increased mobility because it cannot maintain straight line next hop paths without constant updates. MORRP performs consistently well, delivering over 93% of the packets even in highly mobile environments.

We asked the question earlier why not simply modify GPSR with GLS to support multiple interfaces as figures 4.16(a), 4.19(a), and 4.19 all seemed to indicate high reach with high mobility. There are several reasons for this:

- Although GPSR with GLS displays high reach, the end-to-end packet latency is terribly high even under lightly loaded situations. At times, the latency can

be close to 3 to 4 seconds per packet. For many applications, this latency is simply not acceptable.

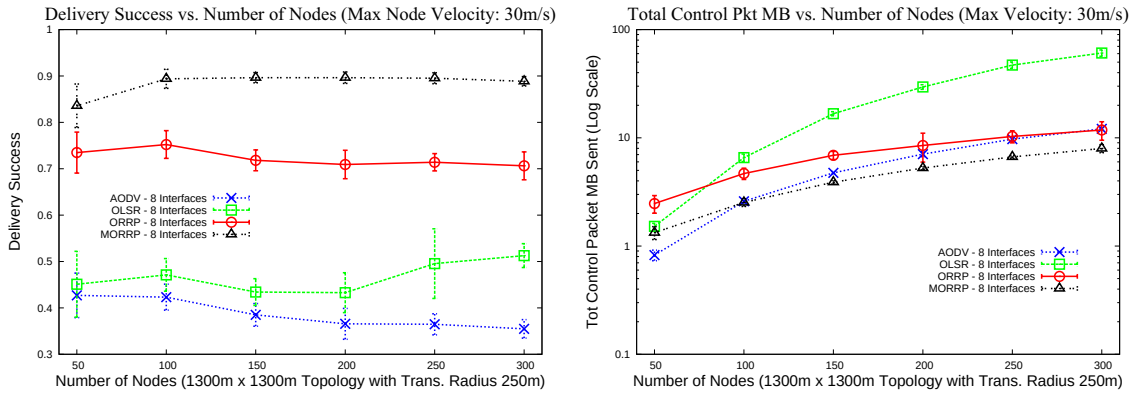
- GPSR with GLS requires node localization which we (and the protocol itself) assume to be a given. Position information is usually garnered through node localization protocols such as AOA [133], APIT [135], and Cricket [134] which often rely on triangulation. Triangulation is difficult in sparser and highly mobile situations because node locations change rapidly. These techniques incur additional overhead in obtaining node localization and even recent attempts to use directional communications for triangulation have limitations [136]. Another method of obtaining node localization is through use of GPS receivers. However, GPS receivers often do not work without "sky access" and can incur overhead in power usage. Essentially, additional hardware would be required.

For the reasons of latency and additional requirement of hardware or overhead associated with node localization, it becomes increasingly difficult to make a case for GPSR with GLS in highly mobile situations.

4.5.2.5 Effect of Increased Network Density

In this subsection, we evaluate how increasing network density affects *reachability* and *amount of control packet bytes* networkwide. The reason why we focus on control packet bytes rather than control packets is simply because MORRP sends bloom filters to its immediate neighbors. Although bloom filters are relatively small in size, the incurred overhead is larger than traditional packets. Figures 4.23(a) and 4.23(b) shows our results varying number of nodes from 50 to 300 with each node having a maximum velocity of 30m/s. 2500 random source and destination pairs are chosen and 512KB CBR packets sent for 20 seconds at a rate of 2Kbps. For fair comparison, we only evaluate MORRP against ORRP and the modified versions of AODV and OLSR to support multiple directional interfaces.

It can be seen that as the number of nodes increases, AODV with multiple interfaces start dropping in reach due to its broadcast nature. OLSR fails because of stale routes due to high mobility. As the density increases, however, OLSR performs seemingly better because closer nodes are more within better reach. ORRP fails to



(a) (b)

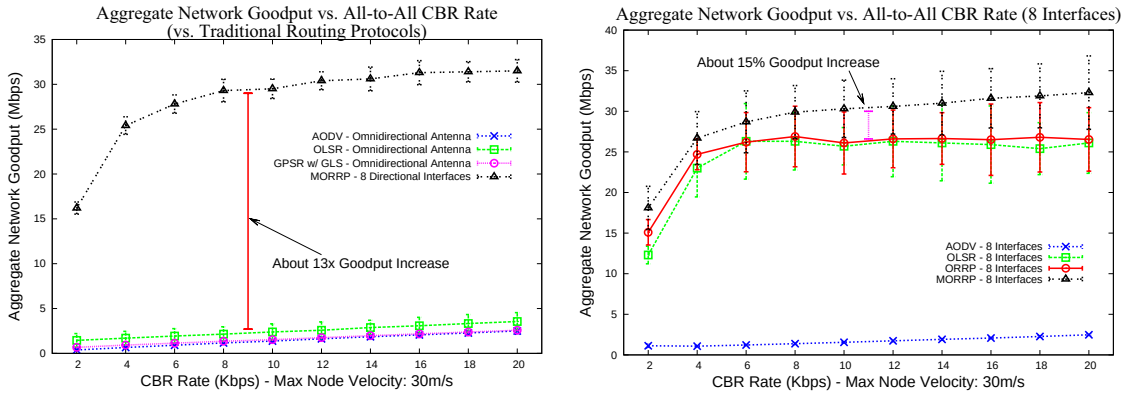
Figure 4.23: Effect of network density on packet delivery success and total control packet bytes sent for multiple routing protocols with multiple directional antennas.

deliver packets because in highly mobile environments, straight line paths are hard to maintain. MORRP delivers roughly 90% of the packets successfully.

It is interesting to note that MORRP seems to send out *less* control packets than ORRP despite it needing to periodically send DRT update messages to all neighbors. The reason for this is simple: In ORRP, RREQ packets travel in a line and a RREP is generated only when this packet intersects with a path generated by an ORRP announcement packet. With MORRP, however, RREQ packets stop being forwarded once it intersects with a destination's "field". Because these "fields" are two or three hops large, MORRP RREQ packets traverse less hops than ORRP RREQ packets. OLSR grows rapidly with network size because more nodes are periodically sending out link-state information. AODV grows despite the constant number of connections due to more nodes in the network forwarding RREQ and RREP packets.

4.5.2.6 Effect of Increased Data Transmission Rate

Although in mobile environments, high reachability naturally leads to high aggregate network goodput, it is important to quantify these gains. In this subsection, we evaluate the effect of increased data rate on network goodput. To do so, we make all-to-all connections simultaneously network-wide and send packets at a set



(a) (b)

Figure 4.24: CBR transmission rate vs. aggregate network goodput for various routing protocols with omnidirectional and directional antennas. Routing with MORRP provides about 10-14X more goodput than traditional routing protocols with omnidirectional antennas and 15-20% better good put than traditional routing protocols modified for directional interfaces.

data rate for 20 seconds. By slowly increasing the rate, we can measure the amount of data that actually gets sent. We expect the capacity constraints will be mostly dependent on medium usage. All nodes are moving in a $1300 \times 1300m^2$ network at a uniformly distributed velocity with a maximum of 30m/s.

We first compare MORRP to AODV, OLSR, and GPSR/GLS to highlight the gains from simply moving from omnidirectional antennas to directional antennas. Figure 4.24 shows our results. As expected, MORRP with 8 interfaces achieves much higher goodput than all the other protocols (roughly 10-14X more than OLSR the closest competitor).

Comparing to AODV and OLSR with 8 directional interfaces and ORRP, MORRP still performs almost 15-20% better than OLSR and ORRP. Again, ORRP fails because it was never designed for mobility and maintenance of straight-line paths becomes difficult in highly mobile environments. The gains from MORRP come from protocol design. Much like the majority of previous work in using directional interfaces in layer 3 routing [41][17], the modified versions of OLSR and AODV simply *adapt* the protocol to support directionality rather than leveraging

the *inherent properties* of directionality to route. Whereas OLSR and AODV even with multiple directional interfaces simply "broadcast" out all intervals for topology control dissemination or route discovery, MORRP utilizes local directionality to disseminate packets along lines to limit flooding. Therefore, it is understandable to see large gains with MORRP over OLSR and AODV with multiple interfaces.

4.5.2.7 Summary of Performance Evaluation

Below we summarize our findings in evaluating MORRP:

- MORRP yields above 93% reachability even in highly mobile environments for medium-sized networks and 89% reach for large-sized networks with medium density.
- Routing using MORRP accounts for an almost 10-14x higher aggregate goodput compared to AODV, OLSR and GPSR/GLS. These gains come primarily through more efficient reuse of the medium under heavy load.
- MORRP yields 15-20% higher aggregate goodput compared to modified versions of AODV and OLSR for 8 directional interfaces and also ORRP. These gains come by using directionality constructively and scalably to overcome problems inherent with directionality.
- End to end packet latency is very low under MORRP compared to AODV, OLSR, and GPSR/GLS because of more efficient medium reuse.
- As node density increases, AODV, OLSR and GPSR/GLS data delivery success drops significantly due to network saturation but does not affect MORRP much.
- MORRP sends *less* control packets than ORRP and much less than AODV, and OLSR in highly mobile situations.

CHAPTER 5
Virtual Direction Routing Protocol

In recent years there's been an explosion of peer-to-peer (P2P) systems for distribution of content. Peer-to-peer systems are attractive for several reasons including 1) its distributed nature, 2) shared overhead, 3) relatively quick response to dynamic network changes, and 4) ease of joins and leaves. One of the biggest challenges in peer to peer systems is information replication/dissemination and discovery in environments of high dynamism. In order to locate where items resolved in a network of peers, various strategies for query propagation and information location need to be implemented.

Peer-to-peer networks are broadly characterized into two major types based on whether or not strict overlay topologies are enforced: *unstructured* and *structured*. Unstructured P2P systems make little or no requirement on how overlay topologies are established and are often easy to build and maintain while being robust to churn [118]. Unfortunately, they tend to have difficulty in finding rare objects and because overlay topologies tend to move toward a power-law distribution when it comes to node degrees, high load is often placed on high degree nodes. Early unstructured systems like Gnutella [125] queried for objects by simply flooding the network with search queries until an item was found. Flooding and even limited flooding techniques (e.g. normalized flooding [122]), tend to be prohibitive in large-scale networks as limited available bandwidth and the large number of nodes limit scalability.

Because of the inherent lack of scalability in flood-based schemes, researchers have looked at several hierarchical and structured based approaches [123, 124]. Hierarchical approaches like Kazaa [126] relied on certain nodes to house more information and coordinate data for a specific subset. Although effective in their own right, hierarchical approaches require reorganization in the event of node failure of local leader nodes. Recently, researchers have utilized novel distributed hash table (DHT) techniques to build virtual structures on overlay networks by mapping nodes

to a specific structure be it a CHORD [123] or a coordinate space [124]. In these self-organizing overlay networks, neighborhood relations are more strictly controlled than in unstructured networks and search queries are propagated along the structure until a match is found. Maintaining the structure, however, makes DHT approaches brittle to attacks and churn.

In recent years, we've seen a large move yet again from hierarchical and structured systems to unstructured, flat, yet scalable techniques to perform search [122, 118]. In this paper, we present Virtual Direction Routing (VDR), a light-weight information replication and location routing technique in unstructured P2P systems. VDR places no restrictions on the underlying overlay topology and utilizes a novel concept we call *virtual directions* to provide efficient query lookup.

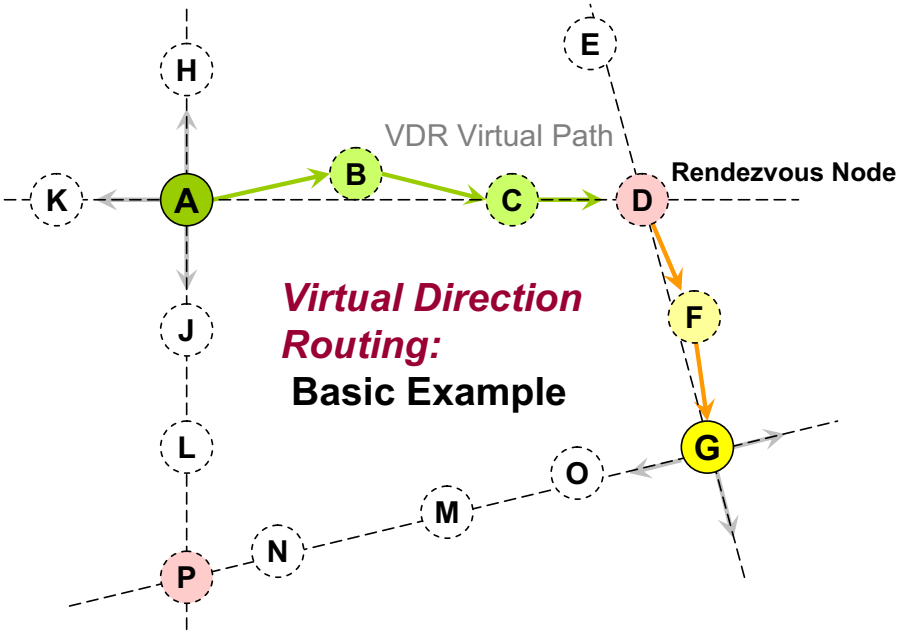


Figure 5.1: Virtual Direction Routing (VDR) Basic Example

In VDR, each node forms a set of *virtual interfaces* (int_i) and assigns immediate neighbors to an interface based on a hash of their unique IDs (e.g. moded with the number of interfaces). State information is replicated at nodes along *virtual orthogonal lines* originating from each node and periodically updated. When a lookup is initiated, instead of flooding the network, query packets are also forwarded along

virtual orthogonal lines until an intersection with the seeded data occurs. If more than one neighbor is assigned to a virtual interface, ties are broken by selecting the neighbor with the ID closest to the search ID. In this way, seed and query packets automatically "gravitate" toward each other increasing the likelihood of intersect.

Key contributions of VDR include:

- Introduction of the concept of Virtual Directions to eliminate the need for virtual coordinate space or DHT structures to locate items in structured-based approaches.
- A flat, highly scalable, and resilient to churn routing algorithm.

We will show that:

- VDR performs much better in replication and reach than random walk
- VDR scales much better than flood-based techniques such as normalized flooding techniques [118]
- VDR performs especially well in dense connectivity situations where the number of neighbors is high. This is valuable as the P2P overlay networks can easily achieve dense connectivity by installing links to several other peers/nodes
- In dynamic network environments where nodes frequently go on and off, VDR significantly outperforms its counterparts in terms of end-to-end reachability and throughput.

The rest of this chapter is organized as follows: Section 5.1 outlines the concept of VDR including a detailed explanation of neighbor to interface assignment as well as node id seeding and lookup. Section 5.2.1 gives a performance evaluation of VDR against several protocols under varying conditions while finally section 5.2.3 gives performance evaluations of VDR under environments with network churn.

5.1 Virtual Direction Routing

The concept of Virtual Direction Search (VDR) is simple: in flat networks, a pair of orthogonal lines centered at different points will intersect at two points at minimum. By replicating and seeding a node's ID along orthogonal lines and performing path query searches along those same directions, one can find intersections and successfully route packets from source to destination through the rendezvous point. This becomes increasingly difficult, however, with internet topologies because internet topologies are typically not flat, instead exhibiting a power-law distribution. It is important therefore, to be able to map nodes to *virtual interfaces* such that by forwarding along *virtual directions*, one can ensure successful destination lookup in an unstructured manner without flooding the network. In this section, we outline VDR and discuss various techniques for mapping neighbors to interfaces in a globally consistent and low maintenance manner under various topologies as well as information replication and lookup.

5.1.1 Virtual Interface Assignment

In this section, we define the concept of *virtual interfaces* as used in VDR. Traditionally, interfaces are physical devices that offer points of connection between other devices. These devices can be physical connectors or wireless antennas that negotiate links between neighbors. In VDR, each node partitions its set of one hop (or low latency) neighbors into a set number (n) of virtual interfaces. The total number of virtual interfaces per node can be fixed or varied but the partitioning strategy (i.e. hash functions) must be globally consistent. We will assume for now that the total number of virtual interfaces a node has (n) is fixed and globally consistent (i.e. all nodes decide on the same number of virtual interfaces and this number does not change).

Each virtual interface is assigned an ID from 0 to $n-1$ and each one hop neighbor (as determined by physical neighbors or by a latency constraint) is assigned to a specific interface. In assigning nodes to an interface, it is important to keep the assignment globally consistent even in the presence of high churns. In other words, nodes assigned to a specific interface should always be assigned to the same interface

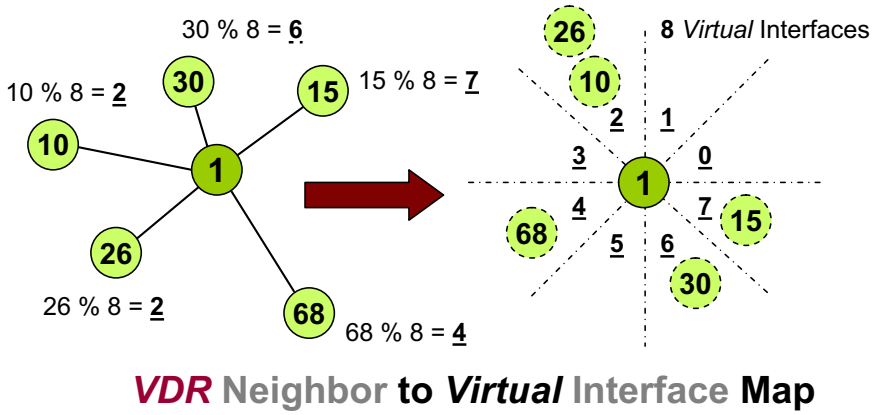


Figure 5.2: VDR Virtual Interface Assignment

even if they are unreachable for a certain amount of time. This will minimize the dynamism and make replicated data less susceptible to network dynamism.

Assuming each node has a unique identifier (e.g. IP address), we employ a simple heuristic to assign neighbors to an interface: 1) Hash each neighbor node ID to 160 bit IDs using SHA-1 [127] and 2) Mod the resulting value by the number of interfaces and assign the node to the interface ID with the resulting value. By assigning neighbors in the preceding manner, we are able to consistently map neighbors to the same interface despite network churn. It is important to note that with these conditions, some interfaces might have more neighbors assigned to them than others. We evaluate another technique whereby we first perform the hash but attempt to make sure that no interface is assigned additional neighbors until all other interfaces have the same number of neighbors in section 5.2.

After all the neighbors have been assigned to a virtual interface, a *virtual north* is randomly chosen for each node. This is done by randomly selecting an interface to be the *virtual north*. This selection is important because information is later forwarded out orthogonal directions with respect to this *virtual north*.

5.1.2 State Information Replication

In order to minimize network flooding, each node replicates its own ID to specific neighbors in the network to make itself easier to locate. To do this, each

node periodically *seeds* its own ID to nodes along orthogonal paths with respect to its own virtual north. Each node will select 4 interfaces that are orthogonal to each other and choose the neighbor along that virtual interface which has the closest hashed ID match to the source node's hashed ID.

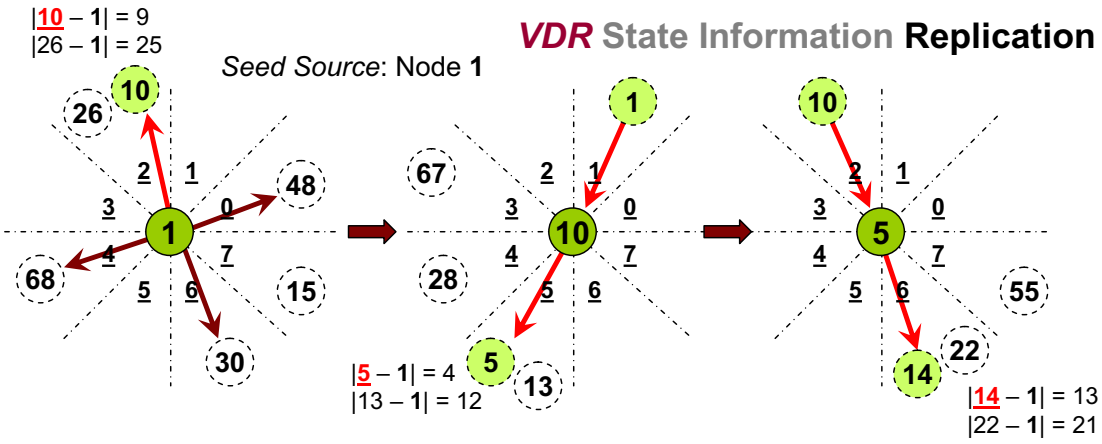


Figure 5.3: VDR State Information Seeding Example

When the neighbor node receives this *seed* packet, it will note the previous hop and source of the packet in its routing table (storing the source as the destination and the previous hop as the next hop) and forward the packet out the interface that is virtually opposite of the receiving interface. The packet is not flooded to all neighbors assigned that that virtual interface, however, but the neighbor that has a hashed ID closest to the source's hashed ID. This will ensure that the packet forward is biased toward nodes that are closer in ID to the source so searching for nodes will form a much higher level of convergence. The packet is forwarded until the TTL is reached. Algorithm 14 gives the algorithm for VDR State Dissemination.

A secondary heuristic (in addition to pure random walk) is used for comparison in our simulations: randomly choosing a neighbor in each virtual direction rather than biasing it toward the ID of the source.

5.1.3 Route Query

When a node wants to do a search for another node in the network, it generates a search request (SREQ) packet and forwards it along virtually orthogonal interfaces

Algorithm 14 VDR State Information Dissemination

```
SendStatePacket(p)
 1: // Check if we are the source - forward opposite if not
 2: if p → Src = ID then
 3:    // We are the source, forward orthogonally
 4:    // Hash Node ID to 32 Bit SHA-1
 5:    Ψ ← SHA1(ID)
 6:    // Get Interface ID of Virtual North
 7:    j ← GetVirtNorthIntID
 8:    α ← NumInterfaces
 9:    // Send out Orthogonal Directions
10:    for i = 1, i ≤ 4, i++ do
11:       Φ ←FindClosestHashedNeighbor(j)
12:       // Send to Neighbor
13:       send(Φ)
14:       j ← ((j + α/4)%α)
15:    end for
16: else
17:    // We are forwarding - only forward opposite
18:    // Hash Packet Source ID to 32 Bit SHA-1
19:    Ψ ← SHA1(p → Src)
20:    // Get received interface ID
21:    j ← (p → Recv_Int_Id)
22:    // Get opposite interface j ← ((j + α/2)%α)
23:    Φ ←FindClosestHashedNeighbor(j)
24:    // Send to Neighbor
25:    send(Φ)
26: end if
```

with respect to its virtual north. Upon receipt of the packet, each neighbor will update its routing table with a "destination - next-hop" entry based on the SREQ packet's source and previous hop and check to see if it has a routing entry to the node the source is searching for. If not, it will forward the node to the interface virtually opposite the receiving interface until it reaches a node that has information to the search destination or reaches its own TTL.

If, however, an entry to the search destination exists, the node will prepare a search reply (SREP) packet which contains the number of hops to the search destination and send it in the reverse direction, relying on routing table entries of the reverse path to get back to the source. Under network churn, if a node in

Algorithm 15 VDR Find Closest Hashed Neighbor

`FindClosestHashedNeighbor(`j`)`

1: // Finds neighbor in virtual direction w/ closest ID match
2: $\gamma_{min} \leftarrow$ 0xFFFF
3: // Get Each Neighbor in Virtual Interface
4: **for all** $k \in$ `Neighbor List(`j`)` **do**
5: $\Theta \leftarrow SHA1(k)$
6: // Check Hash Distance
7: $\gamma \leftarrow abs(\Theta - \Psi)$
8: **if** $\gamma \leq \gamma_{min}$ **then**
9: $\gamma_{min} \leftarrow \gamma$
10: // Store Send Next Hop
11: $\Phi \leftarrow k$
12: **end if**
13: **end for**
14: Return(Φ)

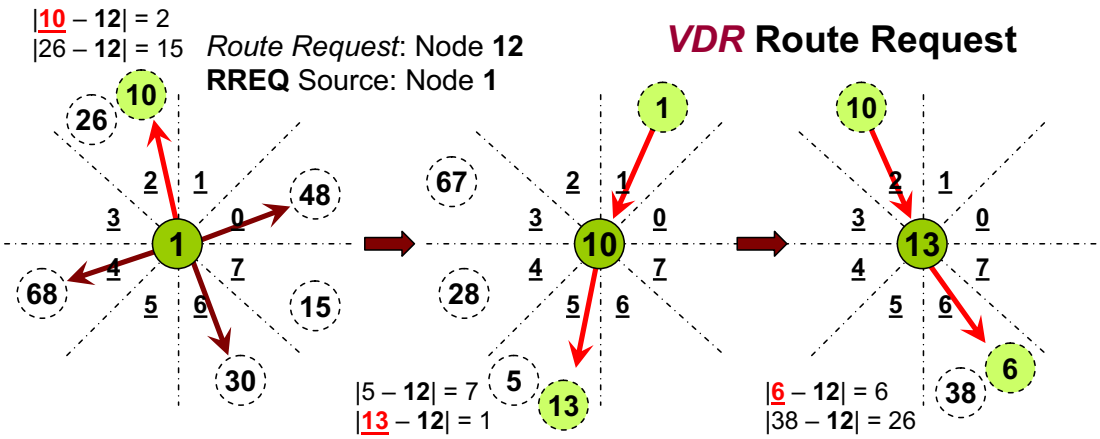


Figure 5.4: VDR Route Request Example 1

the reverse path is no longer active, VDR will re-select a node in the same virtual direction that has the closest hashed ID match to the original source of the SREQ packet to forward to. This ensures a globally consistent biasing of the packets toward the intended destination despite path breakages due to network churn. The algorithm for route queries is similar to Algorithm 14 except that instead of hashing the packet source to Ψ, the packet query destination is hashed to Ψ.

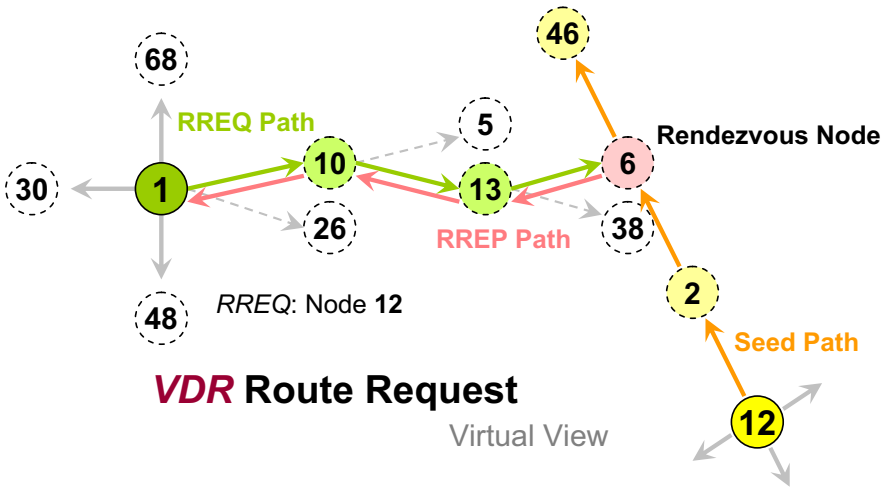


Figure 5.5: VDR Route Request Example 2

5.1.4 Path Deviation

There are instances when nodes wishing to forward in a specific interface find that no neighbors are assigned that virtual interface. VDR employs a strategy to correct for path deviations in an attempt to maintain virtual straight lines. The strategy is fairly straight-forward and employs an angle correction method based on encoding a *multiplier* in the header based on the number of interfaces deviated from the intended send direction. More information can be found in chapter 3.

5.2 Performance Evaluation

In this section, we provide performance evaluations of VDR under various parameters and against some basic random-walk techniques and flooding techniques. The simulations were performed using PeerSim [130] under a cycle-driven model. We wire our topology such that each node has a K out-degree. Because links are bidirectional, it is expected that each node has an average of $2K$ one hop neighbors. Although internet topology is power-law (many nodes have few connections while some nodes have a large number of neighbors), we can assume this topology because 1) peer-to-peer systems are overlay networks and connections are often virtual, 2) 1 hop neighbors can be physical one hop neighbors or links with the lowest latency

Table 5.1: VDR Default Simulation Parameters

Parameter	Values
Number of Nodes	50,000
Number of Virtual Interfaces	8
Simulation Cycles	150
Churn percentage	0% - 50% every 5 cycles
Seed/Search TTL	10 - 100 hops
Seed Entry Expiry	10 Cycles (in churn environments)
Interface Assignment	VDR Hash, VDR Hash w/ NB Shift
Seed/Search Strategy	VDR, VDR-Random, Random Walk
Number of Queries	1000 Randomly Generated

to the source, and 3) peer-to-peer systems represent a subset of the whole network and small-world examples show relatively flat topologies [119, 120].

The performance metrics evaluated include *reachability*, *path stretch vs. shortest path*, and network-wide *state distribution*. We examine these metrics under conditions of varying seed and search *TTL* and *strategies*, *average number of immediate neighbors*, *number of virtual interfaces*, and *percentage of network churns*. All simulations were averaged over 10 runs under random topologies and 95% confidence intervals were mapped. Unless otherwise stated, 1000 randomly generated source and destination queries were generated to start somewhere between cycle 30 to 100. Table 5.1 outlines our default simulation parameters.

Interface assignment refers to the strategy used to assign neighbors to virtual interfaces. The techniques used consist of the standard VDR hash strategy as described in Section 5.1 and a modified heuristic that attempts to evenly distribute the neighbors to each interface (VDR w/ NB Dist). The purpose behind this is to make sure one interface doesn't have a lot of neighbor assignments while the others have none.

The search and seed strategies used include VDR, VDR-Random (VDR-R), and Random Walk Routing (RWR). VDR is the exact strategy described in Section 5.1 while VDR-Random (VDR-R) utilizes the same node to interface assignment technique, but randomizes the node forwarding in a specific direction. In short, if a

virtual interface has multiple nodes assigned to it, VDR-Random will choose a random neighbor associated with that interface rather than choose the neighbor with the hash closest in distance from the source node (for seed packets) or query-search node (for search packets). The random walk strategy is not a pure random walk but its built around the same concept. For the random walk strategy, 4 "walkers" are used with each source node seeding and search for information by sending out 4 random neighbors. Each of the walkers are essentially random walk packets and are dropped after a certain TTL.

5.2.1 Evaluation of VDR in Churn-less Environments

In this section, we examine the effect of search and seed packet TTL, number of virtual interfaces, and average number of neighbors per node on reachability, path stretch, and state distribution under the three seed strategies as listed above (VDR, VDR-Random, and Random Walk Routing) in a fixed, no churn environment. Each node utilizes 8 virtual interfaces with out-degree k assigned to 10. Because links are bi-directional, this means that each node has an average of about 20 neighbors with the deviation from the average to be quite small.

For all cases, seed information is sent only once and the expiry time for each entry is set to the number of simulation cycles as we assume that the network is not dynamic and continual send is redundant. This is also important because under the random walk search (RWR) technique, continual sending of the seed packets lead to different neighbors chosen each time leading to huge confusion in path choices (essentially, all nodes in the network would know a source after a set time if the expiry was set high).

5.2.1.1 Effect of Seed and Search TTL

In this subsection, we examine the effect of search and seed packet TTL on reachability, path stretch, and state distribution under three seed and search strategies. We expect that VDR should provide higher connectivity and lower path stretch than the other strategies (VDR-Random and Random Walk) under lessened seed/search TTL simply because it biases the packets toward a specific ID. Figures 5.6-5.8 give our results.

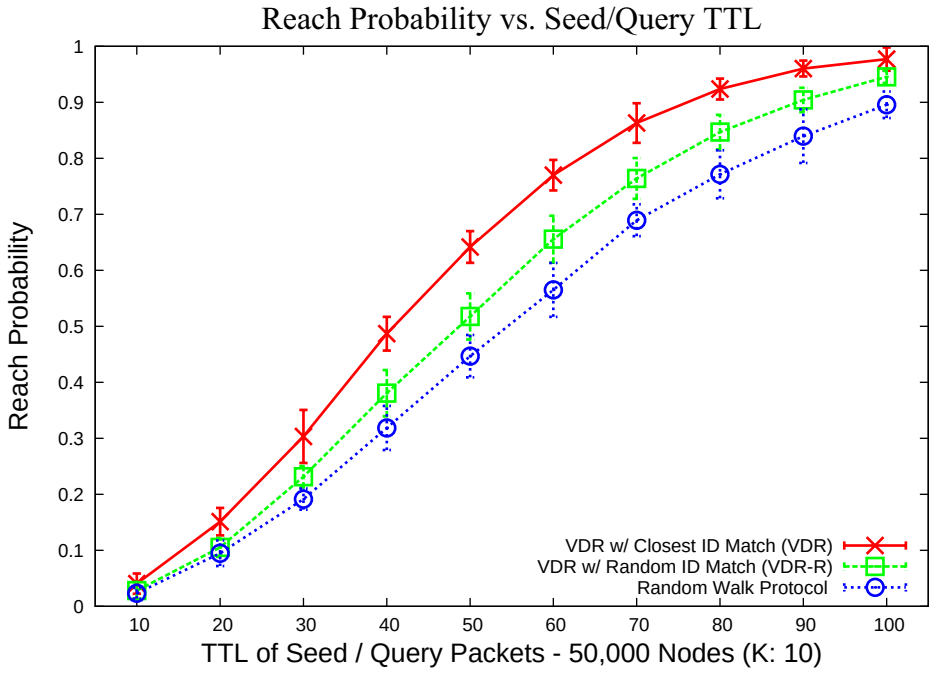


Figure 5.6: Effect of seed and search TTL on VDR reachability. VDR converges to 100% search success much faster than random walk.

It can be seen in figure 5.8 that VDR is able to find information with a higher success rate with less search and seed TTL. This is beneficial because lower TTL lowers the amount of packets traveling network-wide and frees up the links for actual traffic. It is interesting to note that even with a TTL of 100, VDR achieves almost 100% reachability in a network of 50,000 nodes. The random walk search (RWR) technique, as expected, converged the slowest, requiring a much higher TTL to even come close to VDR. The reason that RWR even comes close to VDR is because of the fixed network environment. Under network churns, however, state maintenance would grow dramatically simply because seed dissemination would no longer be sent to the same nodes.

We see from figure 5.7 that the path from the source to a seed node is also much shorter in VDR. Again, this is due to sent packets being biased toward the ID with the closest match. Path stretch (figure 5.8) shows similar results. It is interesting to note the substantially high number of hops traversed through VDR, VDR-Random,

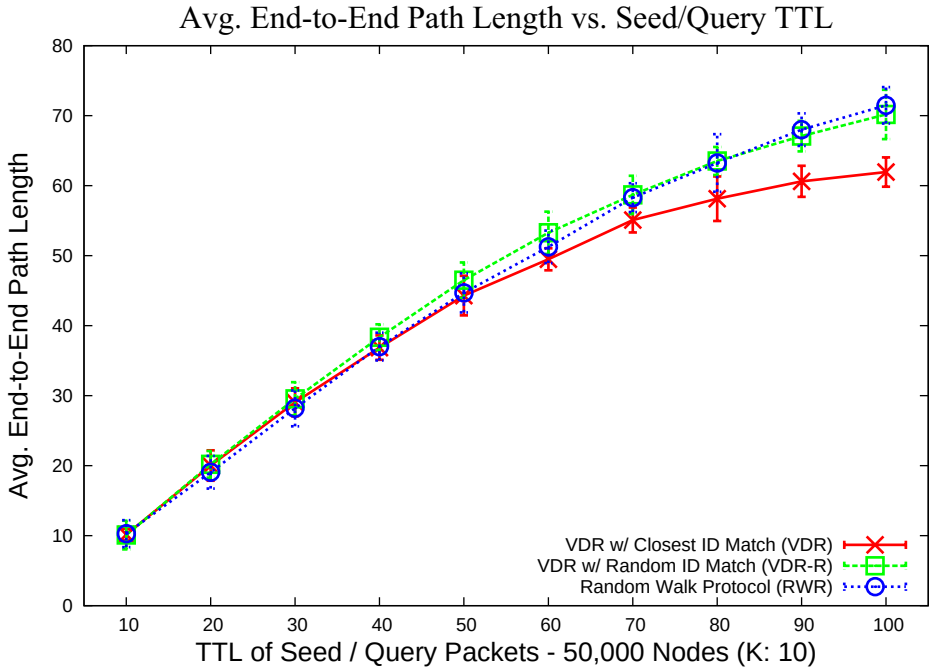


Figure 5.7: Effect of seed and search TTL on VDR end-to-end shortest path. VDR find information seeds with 25% less hops than random walk.

and RWR as compared to shortest path. The shortest path in a wired network grows on order of $Log(N)$ where N is the number of packets in the network. Therefore, it is expected that with 50,000 nodes in the network, the shortest path should be roughly 4.7 hops. It makes sense that these path lengths increase with increased TTL because source and destination pairs that are now farther away can be reached and so the average path length increases with increased reach.

5.2.1.2 Effect of Number of Virtual Interfaces

In this section, we examine the effect of modifying the number of virtual interfaces on reach probability, end-to-end path stretch, and number of states maintained network-wide. With finer granularity (more virtual interfaces), it is expected that the difference between VDR and VDR-R will become smaller because the randomness in neighbor selection for each interface will be reduced as there would only be 1 neighbor per interface. In our simulations, we ran a 50,000 node network with

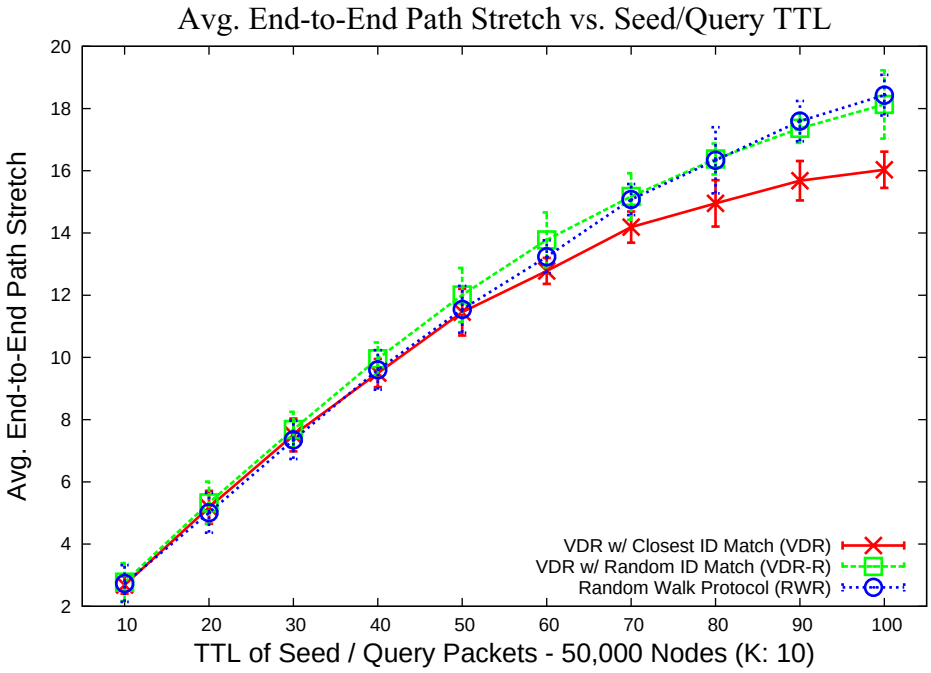
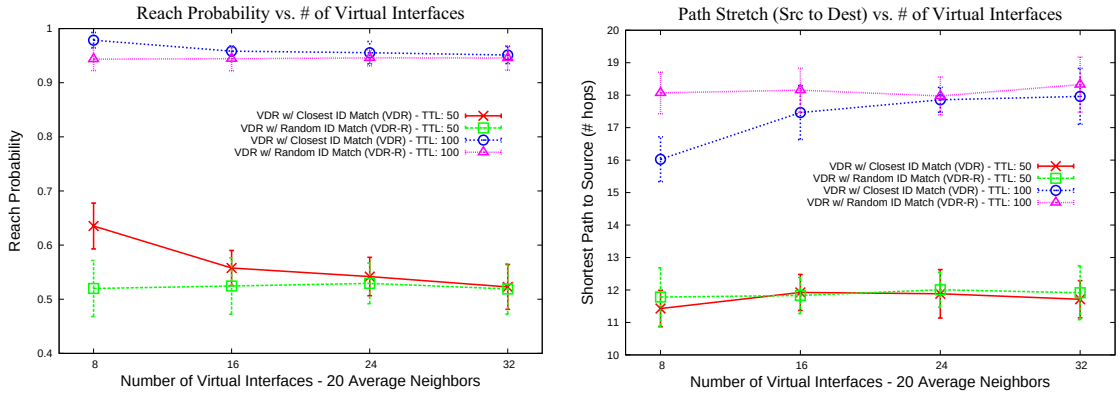


Figure 5.8: Effect of seed and search TTL on VDR end-to-end path stretch. In VDR, path stretch from source to actual data (destination) is roughly 15% less than with random walk.

each node having an average of 20 neighbors. Figures 5.9-5.10 show our results for simulating VDR and VDR-R with a search/seed TTL of 50 and 100.

As shown in figure 5.9(a), VDR has much higher reach probability with lower number of virtual interfaces. This is due to the biasing of IDs such that there is a better convergence. The results are more pronounced at lower seed/search TTL simply because there isn't a saturation of states. The closer VDR gets to 100% reach, the less TTL will affect the packet reach probability resulting in less difference in reach. One of the reasons for greater reach is the lowered path length required for VDR as compared to VDR-R. This again, is due to the biasing of packet IDs. It is interesting that the lower the TTL, the lower the path stretch observed. This is because there is a smaller fraction of delivery success and only the paths that succeed (the shorter ones) are measured.

Figure 5.10 shows the *spread* of states maintained network-wide. VDR and VDR-R average around the same number of states per node. The state deviation



(a) (b)

Figure 5.9: Effect of number of virtual interfaces on reach probability and path stretch. VDR has higher reachability than VDR-R and RWR with increased neighbors because it and search/seed TTL of 50 hops because of biasing packets toward the query destination.

(state spread) pictures how well distributed the states are network-wide. A smaller spread equates to a more evenly distributed network. We see that although VDR provides higher reach and better path stretches, the states are spread rather unevenly network-wide. This is a result of announcement packets constantly biasing their information to nodes with IDs closer to themselves. As a result, neighbors that aren't sent seed packets are often left with fewer states to maintain (only the ones that come in through request packets).

5.2.1.3 Effect of Number of Neighbors

In overlay networks, neighbor nodes are often assigned randomly based on the latency from a specific node rather than physical links. Because of this flexibility in neighbor assignment, it becomes interesting to examine how increasing the number of neighbors per node affects reach, path stretch, and state distribution in networks utilizing VDR, VDR-Random, and RWR.

In these simulations, we fix the virtual interfaces to 8 and increase the k constant (the number of out-degrees) from 5 (average of 10 neighbors/node) to 20 (average of 40 neighbors/node). Because as k is increased, a greater number of neighbors will be assigned to each interface, it is expected that the biasing effect in

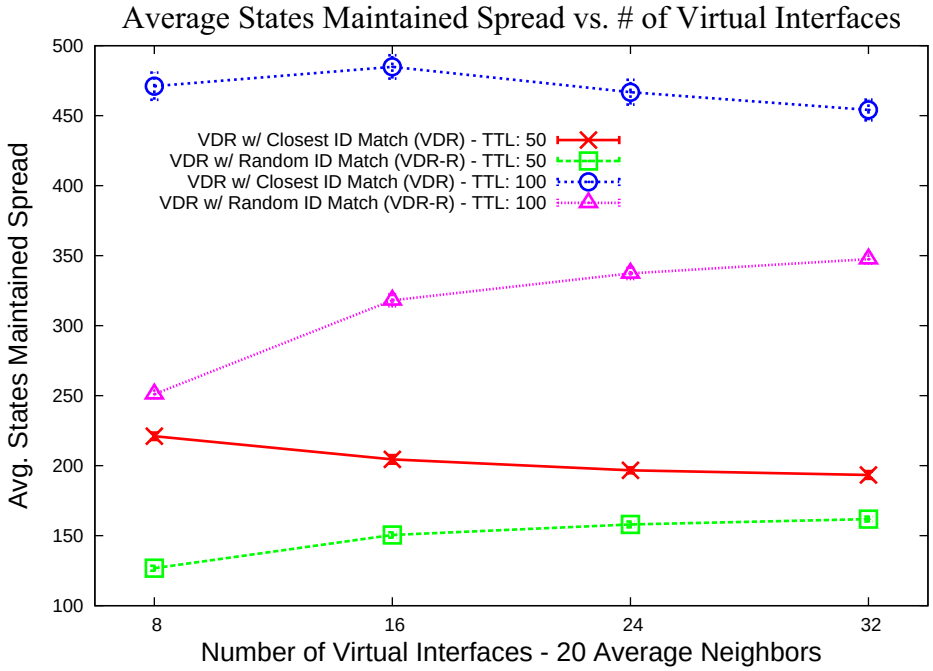
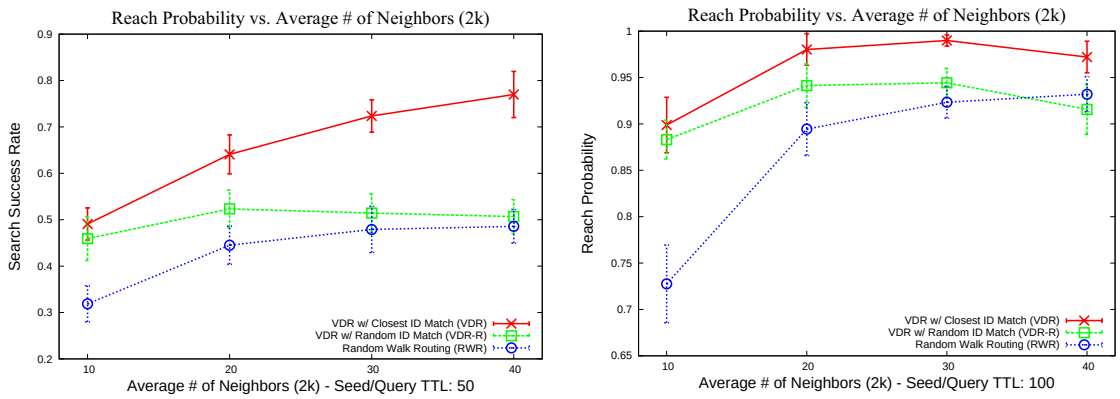


Figure 5.10: VDR has high state distribution deviation suggesting an uneven distribution of state networkwide.

VDR will yield much more beneficial results over VDR-Random for larger k values. As the k is increased, we also expect to observe increased path stretch under lower search/seed TTL simply because the number of nodes in the network are fixed and if each node has more neighbors, paths to each node is inherently shorter (lower shortest path yielding higher path stretch). One would also expect higher reach with increased k because end-to-end paths to all nodes are essentially shorter.

Figure 5.11 shows our results for reachability/search success while increasing k for each of the search and seed strategies at 50 and 100 TTL. It can be seen that with VDR, as the number of neighbors increase, higher reach occurs. Under the same conditions, we see that VDR-Random and RWR yield significantly *less* reach than VDR. Comparing VDR to VDR-Random, we see that as the number of neighbors increase, VDR-Random reach remains relatively constant. This is due in part to the forwarding mechanism found in VDR-Random. In VDR-Random, although the number of neighbors (and thus the number of neighbors assigned to each interface) increases, its decision-making strategy is still to choose a random



(a) (b)

Figure 5.11: Effect of number of neighbors on VDR Reachability for various routing strategies. VDR has higher reachability than VDR-R and RWR with increased neighbors because it and queries and seeds through packet biasing.

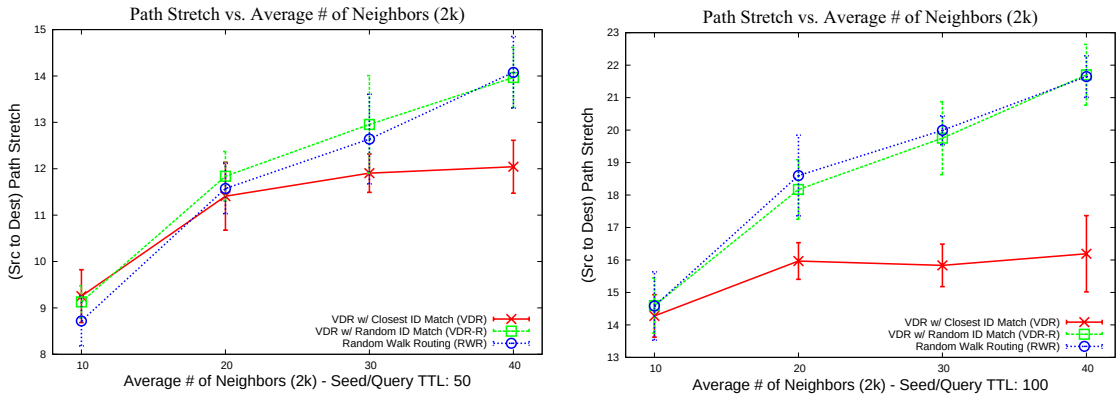
neighbor in a specific virtual interface direction.

The assignment of nodes to a virtual interface negatively impacts the options available to send and therefore the gains by simply having more neighbors (and thurs shorter end-to-end paths) are offset by the losses due to assigning neighbors to rigid virtual interfaces. Because VDR-Random still randomly chooses nodes in a specific interface direction, this results in a relatively constant reach even under increased k.

Figure 5.12 shows the results for end to end path stretch while increasing k for a query and seed TTL of 50 and 100. It's interesting that overall, the path stretch increases with increased number of neighbors. This makes sense because paths chosen are less efficient due to the greater number of neighbors assigned to each interface. Comparatively, however, VDR still yields only slightly shorter path stretch than VDR-Random and RWR with increased number of neighbors. This is due to the biasing effect of forwarding.

5.2.1.4 Evaluation of State Distribution

Its interesting to examine how evenly the state is spread network-wide because in flat topologies, even distribution suggests no single point of failure. Because VDR



(a) (b)

Figure 5.12: Effect of number of neighbors on end-to-end path stretch for various routing strategies. Path stretch increases with more neighbors because in a network of fixed number of nodes, with more connections to and from each node, the average end to end shortest path decreases.

is essentially a biased random-walk technique, it is expected that state is fairly evenly distributed throughout the network. To simulate state distribution, we generated a fixed overlay network with an average of 20 neighbors each. Keeping this overlay network fixed, we ran the simulation 10 times with varying initial *virtual orientations* and took snapshots of the state throughout the simulation, averaging the state per node for each run over all 10 runs. A histogram of the frequency of a average states maintained is shown in figure 5.13.

As figure 5.13 shows, the average states maintained is less evenly distributed in VDR compared to VDR-R and RWR. This suggests that some nodes have more information than other nodes. We suspect this is due to certain nodes with hashed IDs closer to the average being chosen as an appropriate "next hop" more than the other nodes.

5.2.1.5 Evaluation of Network Load Distribution

It has been shown that network congestion can be controlled and limited by routing packets using two-phase routing algorithms [50] [49]. Current overlay measure route cost through hop count and at times, load. In high-traffic networks, by choosing the shortest path, nodes with many connections will become saturated with

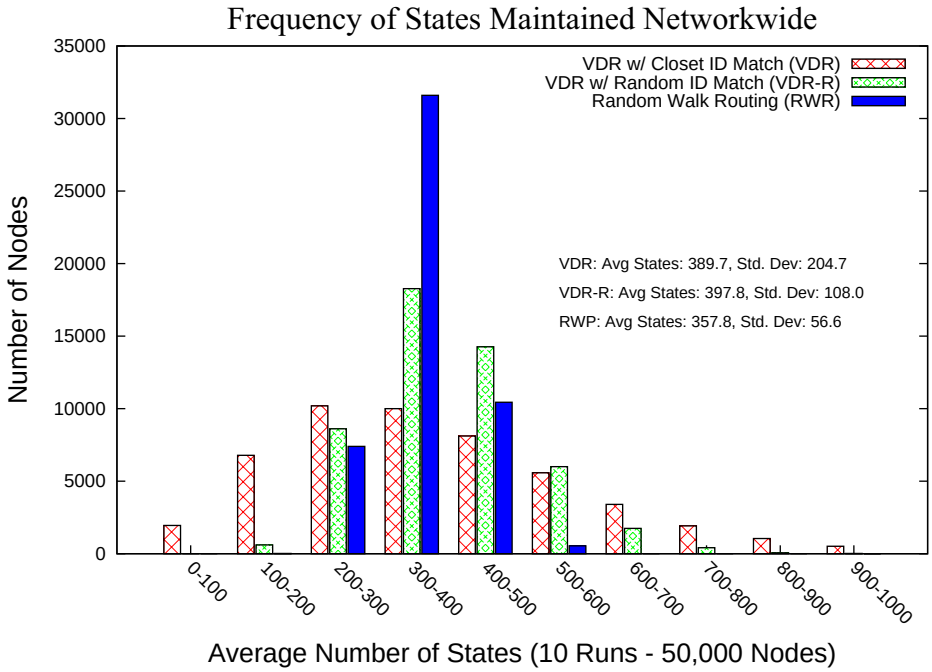


Figure 5.13: State Distribution Network-wide. VDR has high state distribution deviation suggesting an uneven distribution of state networkwide.

packets. Busch et al. [50] has shown that by drawing a perpendicular bisector between source and destination and forwarding packets from source to a random point on the perpendicular bisector which in-turn forwards to destination when that point is reached, load can be balanced across the network. In much the same way, VDR inherently implements a seemingly two-phase routing algorithm because it provides rendezvous abstractions whereby the source sends search packets until it rendezvous with seed information. As a result, it is interesting to see the distribution of load network-wide.

In this subsection, we measure network load by taking snapshots of queue lengths of 50,000 nodes at specific intervals in time. Essentially, we fix the wiring of the overlay network with the only variable for each run being the *virtual orientation*. We then run the simulation 10 times and average out the instantaneous queue lengths per snapshot per simulation run for *each* node. By understanding the variation from the mean number of packets in the queues per node, we can see how

evenly distributed the load is across the network.

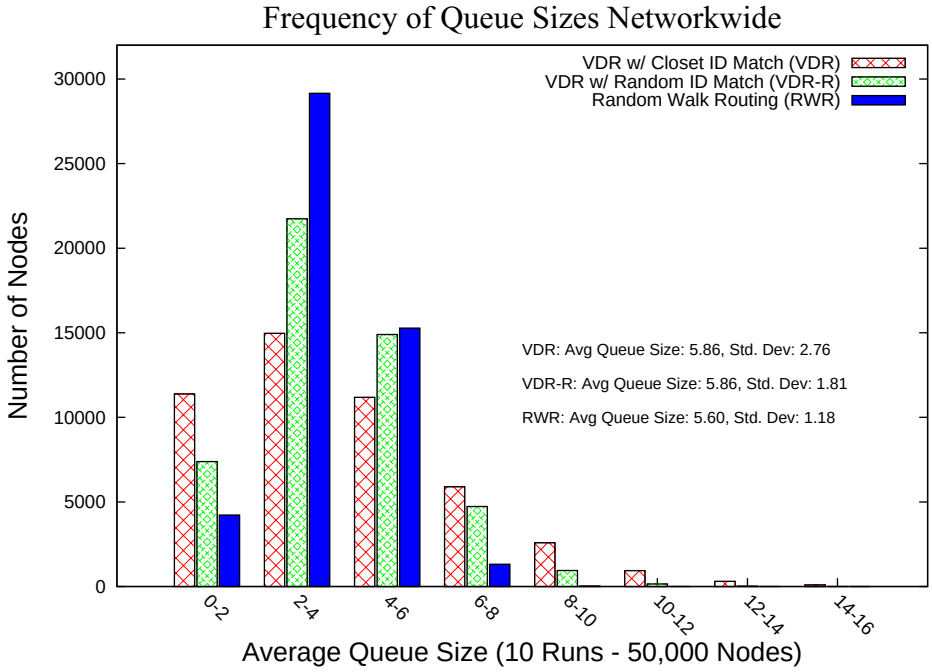


Figure 5.14: Average Queue Size Distribution Network-wide. VDR has high queue size distribution deviation suggesting an uneven distribution of load networkwide.

Figure 5.14 shows the histogram for the number of nodes with queue sizes in the intervals given. It can be seen that there is greater spread of load using VDR compared to VDR-R and RWR suggesting that some nodes incur heavier load than other nodes. This is to be expected because VDR chooses shorter paths and constrains neighbor sending to virtual interfaces. As can be seen, RWR performs the best because random walk models are known to distribute load fairly evenly.

5.2.2 Summary of VDR Performance Evaluations in Static Networks

Below we summarize our findings in evaluating VDR in static networks:

- VDR reaches 3.5% more nodes than VDR-R and 9% more nodes than our modified random walk routing strategy (RWR).
- VDR-R produces the same reach and path stretch results with increasing number of virtual interfaces. This is due to the randomization of sends. VDR,

however, increases reach with fewer number of virtual interfaces because of its biasing technique. The gains disappear if less than 8 interfaces.

- Increasing the number of neighbors generally increased reach and end-to-end path stretch. This was probably due to more node choices per neighbor to bias information.

- VDR states and queues/load are not well distributed.

5.2.3 Evaluation of VDR in Dynamic Environments

In this section, we examine the effect of network churn on reachability, end-to-end path stretch, overall network load and state distribution under the three seed/search strategies as listed above (VDR, VDR-Random, and Random Walk). We simulate churn in the following manner: First, all nodes are connected by assigning an average of k out nodes from each node. Because the links are bi-directional, each node generally has roughly $2k$ neighbors. We then "turn off" half the nodes in the network probabilistically essentially dropping the average number of neighbors to k. The inactive nodes now serve as "raw material" for new connections and nodes currently in the original set can be either turned off or on per simulation cycle.

For our simulations, we fix the number of nodes active to be a constant at half the total available nodes and every 5 cycles, randomly activate a percentage of nodes with respect to the active nodes and deactivate the same number of nodes randomly. When nodes are deactivated, all the packets in their incoming queue are dropped and routing tables emptied. When they are activated, the connections that were originally formed with neighbor nodes remain the same. Thus, nodes can be active and inactive at any point in the simulation and have essentially maintain the same state.

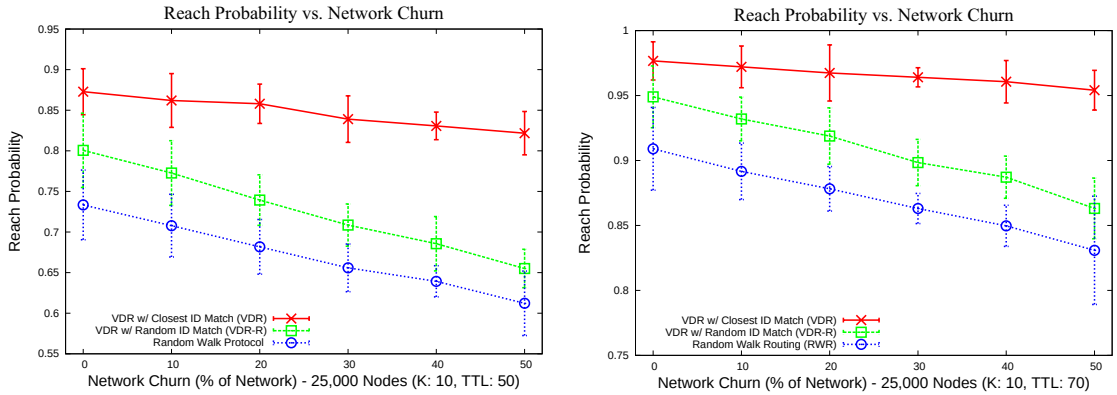
The simulator keeps track of all the nodes that have *ever* been active and queries are generated based on any node that has *ever* been active. This makes sense as in an overlay network, resources that have never been allocated will never be able to be found. Expiry time for each routing entry is set 10 cycles which is the same as the seed/announcement packet send interval. As per the VDR algorithm, search queries are sent out virtual orthogonal directions until they intersect a node

with a path to the destination in their routing table. When this occurs, a search reply packet is generated and sent in the reverse path. In the event of reverse path nodes no longer being up, a node in the same virtual direction is chosen with an ID closest in match to the source of the search query (the destination of the search reply). Under RWR, another node is randomly chosen. In our scenarios, we simulated 25,000 active nodes with a total pool of 50,000 nodes under various churn percentages. The TTL of the seed/announcement packets was set to 150 and each node contained an average of 20 one-hop neighbors.

5.2.3.1 Effect of Churn on Search Success

In this subsection, we examined how the percentage of network churns affect search success. We consider a successful search to have occurred when a search query is initiated and it receives a search reply. It's expected that VDR outperforms VDR-R and RWR simply because it orders neighbors into a more structured fashion with virtual interface assignments. With the RWR, four "walkers" are sent out to random neighbors in search for seed information planted by four seed "walkers". These seed packets are sent out periodically to different neighbors so while at some point there might be more state network-wide, the expiry of the routing information removes stale routes quickly. Our results are in figure 5.15.

As figure 5.15 shows, VDR has the highest percentage of search success/reach under the same network churn rate compared to VDR-R and RWR. It outperforms VDR-R because of the biasing effect of the neighbor send. Because each node has about 15 neighbors and 8 virtual interfaces, there is a possibility that if a neighbor is down (or swapped), VDR will choose another neighbor that is atleast biased toward the search query source (search reply destination) whereas VDR-R will simply randomly choose a node. VDR outperforms RWR simply because in sending search replies, if a previous hop is no longer available, then it must randomly choose a neighbor to forward. If it was forced to perform a random walk until it reached the search query source, it would most definitely result in a packet loss the majority of the times. However, because the random walk need only intersect a node with a path in its routing table to the search query source, there is still relatively high



(a) (b)

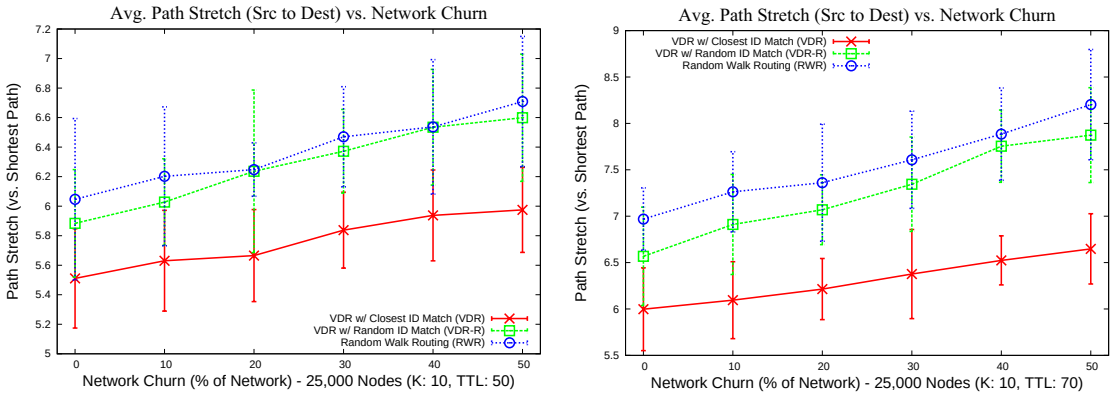
Figure 5.15: Effect of network churn percentage on VDR Reachability for various routing strategies. VDR maintains much higher reachability than VDR-R and RWR with increased percentage of network churn. It also much more robust to network churn, dropping only 5% reach for 50 seed/search TTL and 2% for 70 seed/search TTL compared to VDR-R and RWR which dropped 12-15% going from 0% to 50% network churn for a seed/search TTL of 50 and 7-8% drop for a seed/search TTL of 70.

reach (∼81% even for 50% network churn with a search/seed TTL of 70).

It is also important to understand the rate at which search success/reach drops with respect to the percentage of churns. As can be seen from figure 5.15, VDR drops only 5% in reach from 0% to 50% network churn for a search/seed TTL of 50 and only 2% for a search/seed TTL of 70. This is important because even with 50% nodes turning off and new ones being added, there is still a high degree of reach and robustness to search. VDR-R and RWR, on the other hand, drops about 12-15% in reach for 50 TTL and 7-8% in reach for 70 TTL simply because of the random nature of their send as described before: if a search query reply packet finds the next hop inactive, it must retrace its path without any kind of "hints".

5.2.3.2 Effect of Churn on Path Stretch

Figure 5.16 shows our result on path stretch under churn for VDR, VDR-R, and RWR. It can be seen that as network churn increases, the path stretch increases. This is consistent with expectations as when reverse paths are not reachable, new



(a) (b)

Figure 5.16: Effect of network churn percentage on VDR end-to-end path stretch for various routing strategies. VDR performs with the shortest amount of path stretch as compared to VDR-R and RWR because of a consistent virtual direction and biasing effect of packets.

neighbors must be chosen resulting in often longer paths.

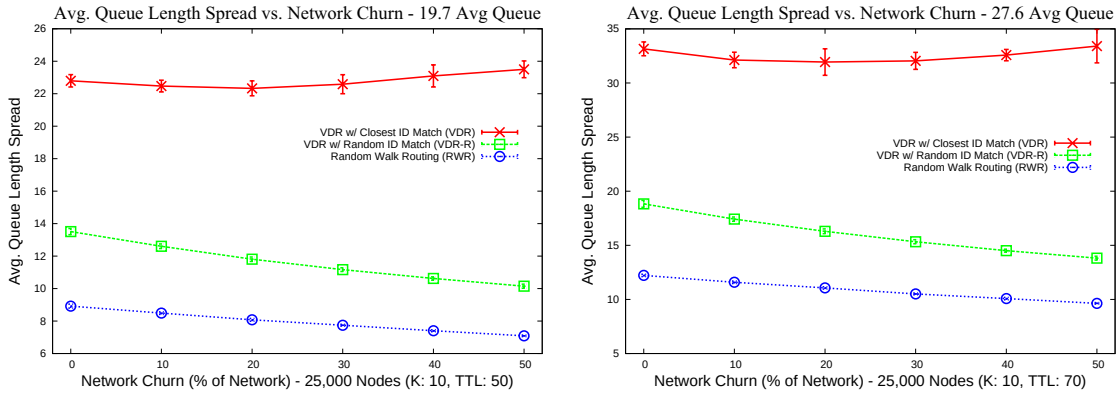
What is interesting, however, is that VDR actually generates much lower path stretch compared to VDR-R and RWR despite the fact that VDR-R and RWR have lower search success and reach. In general, paths with shorter hops are less affected by network churn and therefore it is expected that with lower reach/search success, the paths are generally shorter in general. We see therefore, that VDR not only provides higher path reach, but that it also finds shorter paths.

5.2.3.3 Effect of Churn on Network Load

It has been shown that network congestion can be controlled and limited by routing packets using two-phase routing algorithms [50] [49]. Current overlay measure route cost through hop count and at times, load. In high-traffic networks, by choosing the shortest path, nodes with many connections will become saturated with packets. Busch et al. [50] has shown that by drawing a perpendicular bisector between source and destination and forwarding packets from source to a random point on the perpendicular bisector which in-turn forwards to destination when that point is reached, load can be balanced across the network. In much the same way, VDR inherently implements a seemingly two-phase routing algorithm because it provides

rendezvous abstractions whereby the source sends search packets until it rendezvous with seed information. As a result, it is interesting to see the distribution of load network-wide.

In this subsection, we measure network load by taking snapshots of queue lengths of nodes that are active at specific intervals in time. What we measure is the deviation from the average queue length (in our case, 19.7 average packets per node). A higher deviation means that certain nodes have more packets on average than other nodes. One possible issue with measuring instantaneous queue sizes and averaging it over several snapshots network wide is that it does not take into consideration nodes that have just become active compared to nodes that have been active for an extended amount of time. We balance this issue out by taking the RWR as the base case simply because under random walk strategies, it is known that state is relatively well distributed. Figure 5.17 shows our results.



(a) (b)

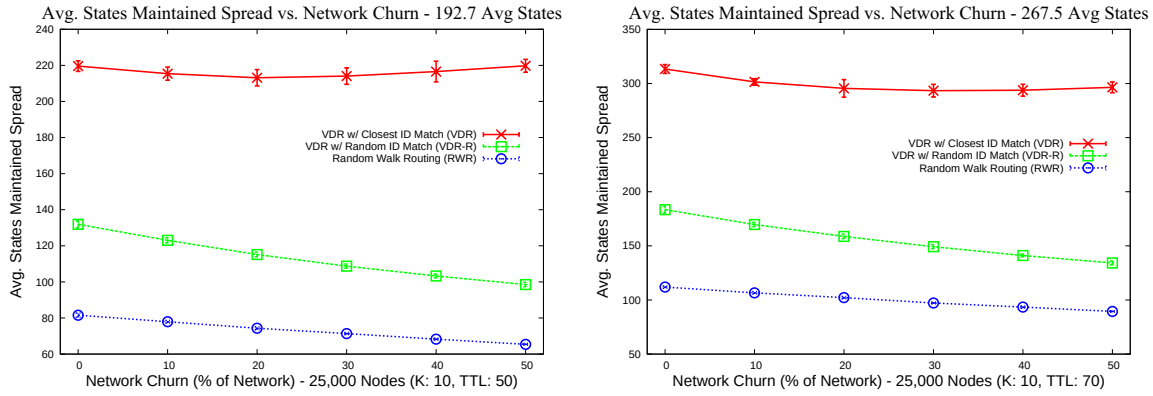
Figure 5.17: Effect of network churn percentage on VDR queue size distribution. VDR has a large spread in queue length distribution suggesting that load is not evenly balanced network-wide (about .65X more than VDR-R and 1X more than RWR).

It can be seen that as expected, VDR has the highest deviation from the average of 14.33 average packets in the queue per node. This is expected as again, certain nodes have preferred paths due to closer ID matching. VDR-R has a higher deviation than RWR because it also makes the routing more rigid by mapping neighbor nodes to virtual interfaces. We see that the queue length spread per node

for VDR is about .65X more than VDR-R and 1X more than RWR. As network churn increases, more queues are emptied per churn interval resulting in fewer packets per node in the queues overall.

5.2.3.4 Effect of Churn on State Distribution

Similar to subsection 5.2.3.3, it is interesting to understand the state distribution network-wide. Comparing VDR to the other techniques, we see that in general, the states network-wide are fairly consistent at about 171 average states per node. It is interesting, therefore, to understand how those states are distributed to see whether or there are more single points of failure with using VDR as compared to the other strategies. The same issues with network churn apply as in subsection 5.2.3.3 and we likewise address this issue by comparing to the baseline random walk technique (RWR) since it is well known that random walk techniques distribute state fairly evenly. Figure 5.18 shows our results.



(a) (b)

Figure 5.18: Effect of network churn percentage on VDR state distribution. VDR showed the highest spread in states maintained network-wide suggesting that certain nodes have more states maintained than other nodes (about .56X higher than VDR-R and 1.5X higher than RWR).

As figure 5.18 depicts, VDR has the highest deviation from the average states maintained per node. This suggest that state is not very evenly distributed network-wide. However, when compared to RWR (the baseline), we see that VDR state spread is only about 1.5X higher than RWR. This again, is due to VDR biasing

random walks such that sending favors certain nodes (ie: announcement packets generally travel through the same neighbors).

Below we summarize our findings in evaluating VDR in dynamic networks:

- VDR shows a 3-4X reach retention rate going from 0% to 50% network churn compared to VDR-R and RWR, showing itself to be much more robust to network churn.
- VDR, even under churn, does not spread state or load evenly.

CHAPTER 6
Future Work

Although we have presented a good amount of work on the feasibility of utilizing directionality to tackle issues associated with scalable, unstructured, mobile, and overlay environments, there are several arenas of work left open to explore. In this chapter, we will outline and briefly detail each area that could be extensions projects. We break this chapter into two major sections: extension future work and novel new directions that are possible. Extensions deal directly with the current protocols and how to make the current protocols more robust. New directions deal with taking the concept of local directionality and applying it in different contexts in new and novel ways.

6.1 Extension Future Work

6.1.1 Virtual Direction Abstraction Analysis

Many of the concepts of rendezvous abstractions through intercepting lines have been presented in detail in this thesis. VDR comes as an abstraction of these concepts into overlay networks. Although an interesting idea, the overlay networks we considered are small world and relatively flat. It would be interesting to understand how this concept of virtual directions can be applied to power-law and other types of topologies. It would also be interesting to perform analysis on VDR to find upper bounds in reachability, path stretch, load, and states maintained network-wide. It would also be interesting to explore how state can be more evenly distributed while at the same time providing high reach.

6.1.2 Hybrid Orthogonal Rendezvous Routing Protocol

In chapter 3 we presented Orthogonal Rendezvous Routing Protocol (ORRP) which uses directionality to provide routing in a scalable manner in wireless mesh networks. One drawback of ORRP is that it requires nodes each equipped with the same number of directional interfaces. We seek to understand whether we can pro-

vide routing based on rendezvous abstractions like ORRP but in a scenario where nodes have varying number of interfaces and angular spreads (including omnidirectional antennas). It is our hope to also provide a hardware prototype of said protocol in a controlled, limited context.

6.1.3 Analysis of Effect of Knobs in MORRP

In chapter 4 we presented Mobile Orthogonal Rendezvous Routing Protocol (MORRP), which leverages bloom filters to provide probabilistic routing. There are several factors that affect delivery success and aggregate goodput and we have identified several "knobs" to explore. Much of our simulations involved fixing one "knob" and systematically iterating through the others. It is interesting to understand from a more analytical perspective how each knob affects routing and we hope to explore mathematical bounds and expected values for each setting. Additionally, spread decay 4.2.1.2 was not evaluated holistically and we intend to provide additional heuristics to see the gains of spread decay with respect to node distance.

6.2 New Directions with Local Directionality

The concept of local directionality is interesting because it adds an extra element of diversity to play with. In this section, we outline some interesting directions for future work based on the concept of local directionality.

- **Multi-path/Multi-interface Diversity** - Knowing local directionality lends itself to some interesting possibilities in routing along different interfaces in the same general direction to provide for multi-path diversity.

- **Directional Network Coding** - Network coding focuses on leveraging the broadcast medium to code combine packets such that each broadcast optimizes the number of packets sent. Network coding is typically not seen as valuable for directional antennas because by nature, directional antennas limit the coverage. What if there was a way to do network coding with directional antennas using the concept of local directionality? By knowing which *direction* a node receives packets from, it is possible to code combine more information in inter-

faces opposite of the received interface such that "more information" is sent along the forward direction.

- **Destination-based Routing** - Knowing a node's physical mobility destination provides interesting opportunities for routing packets. If a node has a sense of local directionality, if it receives a packet from a neighbor with information about the neighbor's movement pattern, it can easily predict where it will be after a certain time based on the direction it receives the packet. This opens up a lot of opportunity in opportunistic routing strategies.

CHAPTER 7
Summary and Conclusions

At the onset of this thesis, we asked a fundamental question: given the push toward more directional forms of communications, is it possible to leverage directionality in layer 3 routing? Beginning two basic primitives: a) local directionality is sufficient to maintain forwarding of a packet on a straight line, and b) two sets of orthogonal lines in a plane intersect with high probability even in sparse, bounded networks, we examined how directionality can be used in two particular contexts: fixed wireless mesh networks and mobile adhoc networks.

For fixed wireless mesh networks, we proposed the Orthogonal Rendezvous Routing Protocol (ORRP), an unstructured forwarding paradigm based on directional communication methods and rendezvous abstractions. By taking the intersection of orthogonal lines originating from source and destination, packets from the source are forwarded to rendezvous nodes which in turn hand them over to the destination, providing simplified routing. We have shown that ORRP provides connectivity under lessened global information (close to 98% reachability in most general cases), utilizes the medium more efficiently (due to directionality of communications), and state-scales on order $N^{3/2}$ at the cost of roughly 1.12 times the shortest path length. In addition, simulations performed on random topologies show that state information is distributed rather evenly throughout the system, and, as a result, no single point of failure is evident.

Additionally, we sought to understand whether adding more lines yielded significant performance gains over the two orthogonal line case. Our analysis showed that the jump between one line and two lines yields significant increases in reach probability and path stretch while the addition of more lines gives only *marginal gains* in reach probability but should choose much better paths resulting in smaller path stretch.

Packetized simulations indicated that in non-void, non-mobile scenarios, there is a significant increase in delivery success and throughput from one to two lines but

as suggested by our analysis, the gains after adding additional lines are slim. Average path length was also shown to decrease until shortest path was almost reached in increasing number of lines. Additionally, as the number of lines increased, total states maintained in the network increased fairly linearly (but still order $N^{3/2}$. As voids were added, however, average path length remained fairly constant due to similar paths taken despite seemingly more paths to choose from. With mobility, it was shown that the addition of lines had very little affect on delivery success but dropped average path length marginally as expected. Overall, the addition of lines yields only marginal gains over the two orthogonal lines scenario.

Because the inherent nature of maintaining straight line paths are difficult in *mobile* environments, we next sought to understand how *directionality* could be used to solve issues associated with high mobility. We presented Mobile Orthogonal Rendezvous Routing Protocol (MORRP), an *unstructured*, *probabilistic*, and *highly mobility tolerant* forwarding paradigm based on directional communication methods and rendezvous abstractions. By utilizing directional routing tables (DRTs), a novel replacement for traditional routing tables, information about nodes in a specific region and nodes along a straight line path is maintained probabilistically. DRTs map interface *directions* to a probabilistic *set-of-IDs* which are decayed and spread locally within a node based on time and *local* node velocity and decayed by number of hops from the source. DRTs provide *regions* where a node can be found in the near-field case and *directions* to send in the far-field case.

When a destination is outside of the near-field region, MORRP relies on taking intersections of orthogonal lines originating from source and destination and forwarding packets from the source to rendezvous nodes which in turn hand them over to the destination providing simplified routing. We have outlined several "knobs" associated with MORRP and evaluated *distance decay factor*, *time decay factor*, and *near-field and far-field threshold* under conditions of varying mobility. It can be seen that spread decay affects networks that are sufficiently dense and has very little affect on sparse networks. Additionally, we compared MORRP against AODV, OLSR, GPSR/GLS, ORRP and modified versions of AODV and OLSR to support multiple interfaces under varying conditions of mobility and node densities and found that

MORRP provides higher reach probability, average path selection, and has much lower control packet overhead. In short, MORRP provides high connectivity even in highly mobile, dense, and unstructured environments.

We then took the same concept of two lines in a plane intersecting with high probability, and brought it to overlay networks by introducing Virtual Direction Routing (VDR). VDR builds off of the concepts introduced in ORRP and intelligently maps next hop neighbors to *virtual interfaces* and routes packets based on the intersection of it virtual lines of node ID seed information and request packets. We show that in a small-world, unstructured, flat topology, VDR provides high reach even with low seed/search TTL (∼98% reach for a TTL of 100 for a 50,000 node network) and that VDR is robust to churn (dropping only 2% in reach going from 0% to 50% network churn).

In summary, we have shown that it is not only possible to take advantage of directionality in layer 3 routing, but the inherent characteristics of directionality yield significant benefits from providing an unstructured, lightweight, and scalable routing solution. We believe that as technologies such as directional antennas, FSO transceivers, and potentially THz FSO transceivers come into mainstream, much of the work presented in this thesis can provide a starting point on how to fully take advantage of the directional nature of communications.

LITERATURE CITED

[1] B. Cheng, M. Yuksel, S. Kalyanaraman, "Orthogonal Rendezvous Routing Protocol for Wireless Mesh Networks," Proceedings of IEEE International Conference on Network Protocols (ICNP), pages 106-115, Santa Barbara, CA, November 2006.

[2] B. Cheng, M. Yuksel, and S. Kalyanaraman, "Directional Routing for Wireless Mesh Networks: A Performance Evaluation," Proceedings of IEEE Workshop on Local and Metropolitan Area Networks (LANMAN), Princeton, NJ, June 2007.

[3] B. Cheng, M. Yuksel, and S. Kalyanaraman, "Rendezvous-based Directional Routing: A Performance Analysis," In Proceedings of IEEE International Conference on Broadband Communications, Networks, and Systems (BROADNETS), Raleigh, NC, September 2007.

[4] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu. Data-centric storage in sensornets with GHT, a geographic hash table. *MONET*, 8(4):427442, 2003.

[5] B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," Proceedings of ACM/IEEE MOBICOM, Boston, MA, August 2000.

[6] J. Li, J. Jannotti, D.S.J. De Couto, D.R. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," Proceedings of ACM/IEEE MOBICOM, pp. 120-130, Aug. 2000.

[7] D. Niculescu, B. Nath, "Trajectory Based Forwarding and Its Applications," Proceedings of ACM/IEEE MOBICOM, 2003.

[8] Hrishikesh Gossain, Tarun Joshi, Carlos De Morais Cordeiro, and Dharma P. Agrawal, "DRP: An Efficient Directional Routing Protocol for Mobile Ad Hoc Networks," IEEE Trans. Parallel and Distributed Systems, vol. 17, no. 12, 2006, pp. 1439-1451.

[9] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, Architecture and Evaluation of an Unplanned 802.11b Mesh Network, *ACM/IEEE MOBICOM* 2005.

[10] C. Perkins and P. Bhagwat. Highly Dynamic Destination- Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. Proceedings of ACM SIGCOMM, pages 234-244, 1994.

[11] C. Perkins and E. M. Royer. "Ad hoc On-Demand Distance Vector Routing." Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, February 1999, pp. 90-100.

[12] D. Johnson, D. Maltz, J. Broch. DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. *Ad Hoc Networking*, edited by Charles E. Perkins, Chapter 5, pp. 139-172, Addison-Wesley, 2001.

[13] W. T. Tsai , C. V. Ramamoorthy , W. K. Tsai, O. Nishiguchi, An Adaptive Hierarchical Routing Protocol, *IEEE Transactions on Computers*, v.38 n.8, p.1059-1075, August 1989.

[14] BBN Technologies. Hazy Sighted Link State (HSLS) Routing: A Scalable Link State Algorithm, http://www.cuwireless.net/downloads/HSLS.pdf.

[15] R. R. Choudhury and N. Vaidya. "Impact of Directional Antennas on Ad Hoc Routing", Proceedings of the Eighth International Conference on Personal Wireless Communication (PWC), Venice, September 2003.

[16] R. R. Choudhury, X. Yang, R. Ramanathan, and Nitin Vaidya. "Using directional antennas for medium access control in ad hoc networks." In Proceedings of ACM MobiCom'02, September 2002.

[17] R. Ramanathan. "On the performance of ad hoc networks using beamforming antennas", Proceedings of ACM MOBIHOC, October 2001.

[18] Su Yi, Yong Pei, and Shivkumar Kalyanaraman, "On the Capacity Improvement of Ad Hoc Wireless Networks Using Directional Antennas," Proceedings of ACM MOBIHOC, Pages 108-116, Annapolis, MD, June 2003.

[19] Asis Nasipuri, Shengchun Ye, and Robert E. Hiromoto. A MAC protocol for mobile ad hoc networks using directional antennas. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2000), September 2000.

[20] Y. Ko, V. Shankarkumar, and N. H. Vaidya. Medium access control protocols using directional antennas in ad hoc networks. In IEEE INFOCOM'2000, March 2000.

[21] L. Bao and J.J.Garcia-Luna-Aceves. Transmission scheduling in ad hoc networks with directional antennas. In ACM MobiCom'02, September 2002.

[22] K. Chebrolu, B. Raman, and S. Sen, "Long-Distance 802.11b Links: Performance Measurements and Experience", 12th Annual International Conference on Mobile Computing and Networking (MOBICOM), Sep 2006, Los Angeles, USA.

[23] M. Yuksel, J. Akella, S. Kalyanaraman, and P. Dutta, "Free-Space-Optical Mobile Ad-Hoc Networks: Auto-Configurable Building Blocks", To appear in ACM/Springer Wireless Networks, 2007.

[24] D. Braginsky and D. Estrin, "Rumor Routing Algorithm for Sensor Networks," Proceedings of WSNA, Atlanta, GA, October 2002.

[25] T. He, C. Huang, B. M. Blum,J. A. Stankovic,and T. F. Abdelzaher, "Range-Free Localization Schemes in Large Scale Sensor Networks," Proceedings of ACM/IEEE MOBICOM, 2003.

[26] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) using AoA," Proceedings of IEEE INFOCOM, San Francisco, CA, 2003.

[27] M. Gerla, X. Hong, and G. Pei, "Landmark routing for large ad hoc wireless networks," Proceedings of IEEE GLOBECOM, San Francisco, CA, Nov. 2000.

[28] The Network Simulator. ns-2. http://www.isi.edu/nsnam/ns.

[29] Municiple Wireless Broadband Information. http://www.muniwireless.com

[30] Champaign-Urbana Wireless. http://www.cuwireless.net

[31] Seattle Wireless. http://seattlewireless.net

[32] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," Proceedings of ACM/IEEE MOBICOM, San Diego, California, September 2003.

[33] S. M. Das, H. Pucha, and Y. C. Hu, "Performance Comparison of Scalable Location Services for Geographic Ad Hoc Routing," Proceedings of IEEE INFOCOM, 2005.

[34] T. Clausen and P. Jacquet. OLSR RFC3626, October 2003. http://ietf.org/rfc/rfc3626.txt.

[35] B. Chen and R. Morris. L+: scalable landmark routing and address lookup for multi-hop wireless networks. In Technical Report 837, MIT LCS, March 2002.

[36] M. Lim, J. Chesterfield, J. Crowcroft, J. Chesterfield, "Landmark Guided Forwarding," 13TH IEEE International Conference on Network Protocols (ICNP'05), pp. 169-178, 2005.

[37] Z. Haas, J. Pearlman, P. Samar, The Zone Routing Protocol (ZRP) for Ad Hoc Networks, IETF Internet Draft, draft-ietf-manet-zone-zrp-04.txt, July 2002.

[38] M. Caesar, M. Castro, E. Nightingale, G. O'Shea and A. Rowstron, "Virtual Ring Routing: Network routing inspired by DHTs", Sigcomm 2006, Pisa, Italy, September 2006.

[39] H. Pucha, S. M. Das, and Y. C. Hu. Imposed route reuse in ad hoc network routing protocols using structured peer-to-peer overlay routing. IEEE Transactions on Parallel and Distributed Systems, 2006.

[40] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, I. Stoica, "Geographic Routing without Location Information," in Proc. ACM/IEEE Mobicom 2003.

[41] R. R. Choudhury, N. Vaidya "Performance of Ad Hoc Routing using Directional Antennas" Journal of Ad Hoc Networks - Elsevier Publishers, November, 2004.

[42] B. Leong, S. Mitra, and B. Liskov, "Path Vector Face Routing: Geographic Routing with Local Face Information". Proceedings of the 13th IEEE International Conference on Network Protocols (ICNP 2005). Boston, MA, November 2005.

[43] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker, "GHT: A geographic hash table for data-centric storage in sensornets," In Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA) 2002.

[44] S. Kalyanaraman S. Yi, Y. Pei and B. Azimi-Sadjadi, "How is the capacity of ad hoc networks improved with directional antennas," Accepted for Publication in Wireless Networks, 2005.

[45] R. R. Choudhury, X. Yang, R. Ramanathan and N. H. Vaidya, "Using Directional Antennas for Medium Access Control in Ad Hoc Networks," In Proceedings of MobiCom, pp. 59-70, Sep. 2002.

[46] M. Sekido, M. Takata, M. Bandai and T. Watanabe, "Directional NAV Indicators and Orthogonal Routing for Smart Antenna Based Ad Hoc Networks," Proc. IEEE International Conference on Distributed Computing Systems Workshops, pp. 871-877, June 2005.

[47] H. Willebrand and B. S. Ghuman, Free Space Optics (Sams Pubs, 1st edition, 2001).

[48] D. J. T. Heatley, D. R. Wisely, I. Neild, and P. Cochrane, Optical Wireless: The story so far, IEEE Communications (December 1998), Volume 36, pp. 72 - 74.

[49] M. Kodialam, T. V. Lakshman, and Sudipta Sengupta, "Efficient and Robust Routing of Highly Variable Traffic", Third Workshop on Hot Topics in Networks (HotNets-III), San Diego (USA), November 2004.

[50] C. Busch, M. Ismail, and J. Xi, "Oblivious Routing on Geometric Networks", Proceedings of the 17th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), pp 316-324, Las Vegas, Nevada, July 2005.

[51] Sean C. Rhea and John Kubiatowicz, "Probabilistic Location and Routing," in *Proceedings of IEEE Infocom*, 2002.

[52] Abhishek Kumar, Jun Xu, and Ellen W. Zegura, "Efficient and scalable query routing for unstructured peer-to-eer networks," in *Proceedings of IEEE Infocom*, 2005.

[53] D. Britz, R. Miller, "Mesh Free Space Optical System: A method to Improve Broadband Neighborhood Area Network backhaul", Proceedings of IEEE LANMAN, Princeton, NJ, June 2007.

[54] M. Mitzenmacher, "Compressed bloom filters," IEEE/ACM Transactions on Networking, vol. 10, no 5, pp. 604-612, 2002.

[55] H. Gossain and C. Cordeiro and D. P. Agrawal, "MDA: An Efficient Directional MAC scheme for Wireless Ad Hoc Networks," Proceedings of IEEE GLOBECOM, 2005.

[56] H. Gossain and T. Joshi and C. Cordeiro and D. P. Agrawal, "A Cross-Layer Approach for Designing Directional Routing Protocol in MANETs," Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), March 2005.

[57] K. Chintalapudi and R. Govindan and G. Sukhatme and A. Dhariwal, "Ad hoc localization using ranging and sectoring," Proceedings of IEEE INFOCOM, March 2004.

[58] U. N. Okorafor and K. Marshall and D. Kundur, "Security and Energy Considerations for Routing in Hierarchical Optical Sensor Networks," Proceedings of the International Workshop on Wireless and Sensor Networks Security, 2006.

[59] U. N. Okorafor and D. Kundur, "OPSENET: A Security Enabled Routing Scheme for a System of Optical Sensor Networks," Proceedings of the International Conference on BroadbandCommunications, Networks and Systems (BROADNETS), 2006.

[60] U. N. Okorafor and D. Kundur, "Efficient routing protocols for a free space optical sensor network," Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), 2005.

[61] J. Llorca and A. Desai and U. Vishkin and C. Davis and S. Milner, "Reconfigurable Optical Wireless Sensor Networks," Proceedings of the SPIE Conference, Remote Sensing, 2003.

[62] Stuart D. Milner and Christopher C. Davis, "Hybrid Free Space Optical/RF Networks for Tactical Operations," Proceedings of IEEE MILCOM, 2004.

[63] F. Liu and U. Vishkin and S. D. Milner, "Bootstrapping Free-Space Optical Networks," Proceedings of the IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2005.

[64] J. Llorca and Archana Anibha and S. D. Milner, "Design and implementation of a complete bootstrapping model for free-space optical backbone networks," Proceedings of SPIE Optics and Photonics, Free-Space Laser Communications VI, 2006.

[65] Stephen G. Lambert and William L. Casey, "Laser Communications in Space," Artech House, 1995.

[66] R. M. Gagliardi and S. Karp, "Optical Communications," John Wiley and Sons, 1976.

[67] U.S. Department of Energy, Energy Efficiency and Renewable Energy, Solid-State Lighting, http://www.netl.doe.gov/ssl.

[68] A. Bergh and G. Craford and A. Duggal and R. Haitz, "The Promise and Challenge of Solid-State Lighting," Physics Today Online, 2001.

[69] "CEI/IEC825-1: Safety of Laser Products," International Electrotechnical Commission, 1993.

[70] J. D. Rancourt, "Optical Thin Films," Macmillan, 1987.

[71] I. I. Kim et al., "Wireless optical transmission of fast ethernet, FDDI, ATM, and ESCON protocol data using the TerraLink laser communication system," SPIE Optical Engineering, 1998.

[72] B. Pattan, "Robust Modulation Methods and Smart Antennas in Wireless Communications," Prentice Hall, 1999.

[73] A. Paulraj and D. Gesbert, "Smart antennas for mobile communications, Chapter in Wiley Encyclopedia on Electrical and Electronics Engineering," Wiley, 1998.

[74] D. Kedar and S. Arnon, "Backscattering-Induced Crosstalk in WDM Optical Wireless Communication," IEEE/OSA Journal of Lightwave Technology, 2005.

[75] D. J. Goodwill and D. Kabal and P. Palacharla, "Free space optical interconnect at 1.25 Gb/s/channel using adaptive alignment," Optical Fiber Communication Conference and the International Conference on Integrated Optics and Optical Fiber Communication (OFC/IOOC), 1999.

[76] G. E. F. Faulkner and D. C. O'Brien and D. J. Edwards, "A cellular optical wireless system demonstrator," IEEE Colloquium on Optical Wireless Communications, 1999.

[77] K. Ho and J. M. Kahn, "Methods for crosstalk measurement and reduction in dense WDM systems," IEEE/OSA Journal of Lightwave Technology, 1996.

[78] G. Pang et al., "Optical Wireless based on High Brightness Visible LEDs," IEEE Industry Applications Conference, 1999.

[79] G. Pang et al., "LED Wireless," IEEE Industry Applications Conference, 2002.

[80] H. Uno and K. Kumatani and H. Okuhata and I. Shrikawa and T. Chiba, "ASK digital demodulation scheme for noise immune infrared data communication," ACM/Springer Wireless Networks, 1997.

[81] A. J. C. Moreira and R. T. Valadas and A. M. O. Duarte, "Optical interference produced by artificial light," ACM/Springer Wireless Networks, 1997.

[82] Y. E. Yenice and B. G. Evans, "Adaptive beam-size control scheme for ground-to-satellite optical communications," SPIE Optical Engineering, 1999.

[83] M. Naruse and S. Yamamoto and M. Ishikawa, "Real-time active alignment demonstration for free-space optical interconnections," IEEE Photonics Technology Letters, 2001.

[84] J. W. Armstrong and C. Yeh and K. E. Wilson, "Earth-to-deep-space optical communications system with adaptive tilt and scintillation correction by use of near-Earth relay mirrors," OSA Optics Letters, 1998.

[85] S. Arnon and N. S. Kopeika, "Performance limitations of free-space optical communication satellite networks due to vibrations-analog case," SPIE Optical Engineering, 1997.

[86] S. Arnon and S. R. Rotman and N. S. Kopeika, "Performance limitations of free-space optical communication satellite networks due to vibrations: direct detection digital mode," SPIE Optical Engineering, 1997.

[87] G. C. Boisset and B. Robertson and H. S. Hinton, "Design and construction of an active alignment demonstrator for a free-space optical interconnect," IEEE Photonics Technology Letters, 1999.

[88] E. Bisaillon and D. F. Brosseau and T. Yamamoto and M. Mony and E. Bernier and D. Goodwill and D. V. Plant and A. G. Kirk, "Free-space optical link with spatial redundancy for misalignment tolerance," IEEE Photonics Technology Letters, 2002.

[89] X. Zhu and J.M. Kahn, "Free-Space Optical Communication through Atmospheric Turbulence Channels," IEEE Transactions on Communications, 2002.

[90] F. Shubert, "Light-Emitting-Diodes-dot-org, " http://www.lightemittingdiodes.org.

[91] D. C. O'Brien, et al., "High-speed integrated transceivers for optical wireless," IEEE Communications Magazine, 2003.

[92] P. Djahani and J. M. Kahn, "Analysis of infrared wireless links employing multibeam transmitters and imaging diversity receivers," IEEE Transactions on Communications, 2000.

[93] S. Arnon, "Effects of atmospheric turbulence and building sway on optical wireless-communication systems," OSA Optics Letters, 2003.

[94] J. M. Kahn and J. R. Barry, "Wireless Infrared Communications," Proceedings of the IEEE, 1997.

[95] S. Acampora and S. V. Krishnamurthy, "A broadband wireless access network based on mesh-connected free-space optical links," IEEE Personal Communications, 1999.

[96] D. J. T. Heatley and D. R. Wisely and I. Neild and P. Cochrane, "Optical Wireless: The story so far," IEEE Communications, 1998.

[97] V. Vistas and A. C. Boucouvalas, "Performance Analysis of the Advanced Infrared (Air) CSMA/CA MAC Protocol for Wireless LANs," ACM/Springer Wireless Networks, 2003.

[98] "Terescope 10 Series," http://www.mrv.com/product/MRV-TS-004.

[99] "Cotco, Ultra Bright Red 5mm LED, Model number LC503AHR2-30P-A," http://www.cotco.com.

[100] Lightpointe Inc., http://www.lightpointe.com, 2004.

[101] Terabeam Inc., http://www.terabeam.com, 2004.

[102] H. Willebrand and B. S. Ghuman, "Free Space Optics," Sams Pubs, 2001.

[103] C. Chuah and D. Tse and J. M. Kahn, "Capacity of Multi-Antenna Array Systems in Indoor Wireless Environment," Proc. of IEEE Global Commun. Conf., 1998.

[104] M.P. Christensen and P. Milojkovic and M.J. McFadden and M.W. Haney, "Multiscale optical design for global chip-to-chip optical inter connections and misalignment tolerant packaging," IEEE Journal on Selected Topics in Quantum Electronics, 2003.

[105] T. S. Rappaport, "Wireless Communications: Principles and Practice," Prentice-Hall, 2006.

[106] J. Akella and M. Yuksel and S. Kalyanaraman, "Error Analysis of Multi-hop Free-Space-Optical Communication," Proceedings of IEEE International Conference on Communications (ICC), May 2005.

[107] B.M. Williams and L. A. Hoel, "Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical results," Journal of Transportation Engineering, 2003.

[108] J. M. Kahn and R. H. Katz and K. S. J. Pister, "Next century challenges: mobile networking for smart dust," Proceedings of MOBICOM, 1999.

[109] J. Akella and M. Yuksel and S. Kalyanaraman, "A Relative Ad-hoc Localization Scheme using Optical Wireless," Proceedings of IEEE/Create-Net/ICST International Conference on Communication System Software and Middleware (COMSWARE), 2007.

[110] Cognitive and Self-Healing Routing for Sensor Networks E. Gelenbe, P. Liu, B.K. Szymanski, M. Lisee, and K. Wasilewski First International Alliance Conference, Washington, D.C., September, 2007.

[111] Self-healing routing: a study in efficiency and resiliency of data delivery in wireless sensor networks K. Wasilewski, J. Branch, M. Lisee and B.K. Szymanski Proc. Conference on Unattended Ground, Sea, and Air Sensor Technologies and Applications, SPIE Symposium on Defense & Security, Orlando, FL, April, 2007.

[112] A Self-selection Technique for Flooding and Routing in Wireless Ad-hoc Networks G.G. Chen, J.W. Branch, and B.K. Szymanski Journal of Network and Systems Management, vol.14, no.3, September 2006, pp. 359-380.

[113] Self-selective Routing for Wireless Ad hoc Networks G.G. Chen, J.W. Branch, and B.K. Szymanski Proc. IEEE Int. Conf. Wireless and Mobile Computing, Networking and Communications WiMob 2005, Vol. 3, Montreal, Canada, Aug. 2005, pp. 57-64.

[114] Local Leader Election, Signal Strength Aware Flooding, and Routeless Routing G. Chen, J. Branch, B. Szymanski Proc. 5th IEEE International Workshop on Algorithms for Wireless, Mobile, Ad Hoc Networks and Sensor Networks, WMAN05, April, 2005, Denver, CO.

[115] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Frans Kaashoek, F. Dabek, and H. Balakrishnan. "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications." IEEE Transactions on Networking, February 2003.

[116] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. "A Scalable Content-Addressable Network." Proceedings of ACM SIGCOMM 2001.

[117] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," IEEE Transactions on Information Theory, vol. IT-46, no. 2, pp. 388-404, March 2000.

[118] C. Ghantsidis, M. Mihail, and A. Saberi. "Hybrid search schemes for unstructured peer-to-peer networks." Proceedings of IEEE INFOCOM, 2005.

[119] A. Iamnitchi, M. Ripeanu, and I. Foster. "Small-world filesharing communities." Proceedings of IEEE INFOCOM, 2004.

[120] J. Kleinberg. "Navigation in a small world." Nature, page 845, 2000.

[121] A. Kumar, J. Xu, and E. W. Zegura. "Efficient and scalable query routing for unstructured peer-to-peer networks." Proceedings of IEEE INFOCOM, 2005.

[122] H. Guclu and M. Yuksel. "Scale-Free Overlay Topologies with Hard Cutoffs for Unstructured Peer-to-Peer Networks." Proceedings of IEEE ICDCS, 2007.

[123] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Frans Kaashoek, F. Dabek, and H. Balakrishnan. "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications." IEEE Transactions on Networking, February 2003.

[124] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. "A Scalable Content-Addressable Network." Proceedings of ACM SIGCOMM 2001.

[125] Gnutella home page. http://gnutella.wego.com.

[126] Kazaa home page. http://www.kazaa.com.

[127] SHA-1 RFC. http://tools.ietf.org/html/rfc3174.

[128] B. Cheng, M. Yuksel, S. Kalyanaraman, "Orthgonal Rendezvous Routing Protocol for Wireless Mesh Networks," Proceedings of ICNP 2006.

[129] M. Mitzenmacher, "Compressed bloom filters," IEEE/ACM Transactions on Networking, vol. 10, no 5, pp. 604-612, 2002.

[130] PeerSim: A Peer-to-Peer Simulator. http://peersim.sourceforge.net/.

[131] A. Nasipuri, J. Mandava, H. Manchala, R. Hiromoto, "On Demand Routing Using Directional Antennas in Mobile Ad Hoc Networks", Proceedings of IEEE WCNC 2000.

[132] U. Acer, S. Kalyanaraman, A. Abouzeid, "Weak State Routing for Large Scale Dynamic Networks", Proceedings of MOBICOM 2007, Sept 2007.

[133] D. Niculescu and B. Nath, Ad Hoc Positioning System (APS) using AoA, INFOCOM 03, San Francisco, CA,2003.

[134] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In ACM MOBICOM, Boston, MA, August 2000.

[135] T. He, C. Huang, B. M. Blum,J. A. Stankovic,and T. F. Abdelzaher. Range-Free Localization Schemes in Large Scale Sensor Networks, MobiCom 2003.

[136] J. Akella, M. Yuksel, S. Kalyanaraman, A Relative Ad-hoc Localization Scheme using Optical Wireless IEEE/CreateNet COMSWARE 2007, Bangalore, India, January 2007.

[137] HLS patch for ns-2.29. http://www.cn.uni-duesseldorf.de/staff/kiess/software/hls-ns2-patch.

[138] OLSR for ns-2.29. http://masimum.dif.um.es/?Software:UM-OLSR

[139] W. Kiess, H. Fler, J. Widmer, M. Mauve. Hierarchical Location Service for Mobile Ad-Hoc Networks. ACM SIGMOBILE Mobile Computing and Communications Review (MC2R) 8 (4), pp. 47–58, October 2004.

[140] T. Kleine-Ostmann, K. Pierz, G. Hein, P. Dawson and M. Koch, Audio signal transmission over THz communication channel using semiconductor modulator. Electronics Letters, Vol. 40, No. 2. January 2004.

[141] T-A Liu, G-R Lin, Y-C Chang, C-L Pan, Wireless audio and burst communication link with directly modulated THz photoconductive antenna. Optics Express, Vol. 13, No. 25. December 2004.

[142] R. Piesiewicz, C. Jansen, D. Mittleman, T. Kleine-Ostmann, M. Koch, and T. Kurner. Scattering Analysis for Modeling of THz Communication Systems. IEEE Transactions on Antennas and Propagation, Vol. 55, No. 11. November 2007.

[143] B. Stadler, G. Duchak. TeraHertz Operational Reachback (THOR) A Mobile Free Space Optical Network Technology Program. IEEE Aerospace Conference Proceedings. 2004.

[144] W. Zhou and Y. Zhang. Propagation Characteristics of the Terahertz Pulse in Free Space. IEEE THz Technology, Ultrafast Measurements, and Imaging. 2006.

[145] M. Fitch and R. Osiander. Terahertz Waves for Communications and Sensing. Johns Hopkins APL Technical Digest. Vol. 25, No. 4. 2004.

APPENDIX A
Calculating ORRP Reach Probability

In this appendix, we outline our approach for calculating ORRP's reachability probability for a rectangular topology area. Similar approaches were taken to obtain the results for circular and elliptical topologies shown in Figure 3.11.

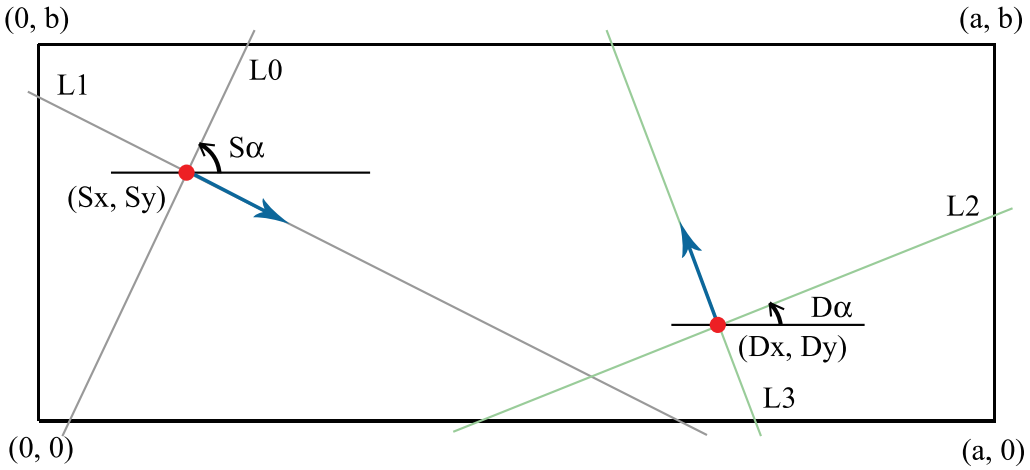


Figure A.1: ORRP Reach Probability Analysis Calculation

Given a Euclidean 2-D rectangular topology area defined by coordinate ranges $0 < y < b$ and $0 < x < a$, we assume that the nodes are randomly oriented with local "north" between 0° and 90°. Our goal is to find the probability that a randomly selected source-destination pair in this rectangular area will not be able to reach each other.

We first find the conditional probability that a particular source point will not be reachable by any other point in the area. Given a source located at (S_x, S_y) and oriented in S_α such that $S_\alpha \leq 90°$, $S_x \leq a$ and $S_y \leq b$ (node is within the bounds of the topology), we assume that L_0 and L_1 are orthogonal lines that intersect source S with one line oriented in the direction S_α. Now, suppose that the source S wishes to send to a destination node D located at $D = (D_x, D_y)$ with D_α such

that $0 \leq D_\alpha \leq 90°$, $D_x \leq a$, $D_y \leq b$ and L_2 and L_3 are orthogonal that intersect at D with one oriented in the direction D_α. We need to analytically construct the condition that the source S will be unreachable by any destination D. To do so:

Step 1: We formulate the slopes (m) and the equations for the four lines L_0, L_1, L_2, and L_3. As an example, for line L_0, we formulate as follows:

$$\begin{aligned} L_0 : m_0 &= tan(S_\alpha) \\ y_0(x) &= x \tan(S_\alpha) + S_y - \tan(S_\alpha) \times S_x \end{aligned} \tag{A.1}$$

Step 2: We determine four possible intersection points (excluding the source point S and the destination point D) among the lines L_0, L_1, L_2, and L_3:

$$\begin{aligned} L_2 \; and \; L_0 : \; & (x_{20}, y_{20}) \; s.t. \; y_0(x_{20}) = y_2(x_{20}) \\ L_2 \; and \; L_1 : \; & (x_{21}, y_{21}) \; s.t. \; y_1(x_{21}) = y_2(x_{21}) \\ L_3 \; and \; L_0 : \; & (x_{30}, y_{30}) \; s.t. \; y_0(x_{30}) = y_3(x_{30}) \\ L_3 \; and \; L_1 : \; & (x_{31}, y_{31}) \; s.t. \; y_1(x_{31}) = y_3(x_{31}) \end{aligned}$$

Step 3: We finally formulate the analytical unreachability conditions as that all four of the intersection points must NOT be in the topology rectangular area. Thus, constraints for intersection points for unreachability can be written as:

$$NOT(0 \leq x_{20} \leq a \; AND \; 0 \leq y_{20} \leq b) \tag{A.2}$$

$$NOT(0 \leq x_{21} \leq a \; AND \; 0 \leq y_{21} \leq b) \tag{A.3}$$

$$NOT(0 \leq x_{30} \leq a \; AND \; 0 \leq y_{30} \leq b) \tag{A.4}$$

$$NOT(0 \leq x_{31} \leq a \; AND \; 0 \leq y_{31} \leq b) \tag{A.5}$$

To numerically calculate unreachability probability, we first obtain the intersection point coordinates in terms of S_x, S_y, S_α, D_x, D_y, and D_α by using the line equations in the intersection point equalities (e.g. in (A.1)). For example, x_{20} and

y_{20} can be derived as follows:

$$
\begin{aligned}
& y_0(x_{20}) = y_2(x_{20}) \\
& x_{20}\tan(S_\alpha) + S_y - \tan(S_\alpha) \times S_x \\
& \quad = x_{20}\tan(D_\alpha) + D_y - \tan(D_\alpha) \times D_x
\end{aligned}
$$

$$
x_{20} = \frac{D_y - D_x\tan(D_\alpha) - S_y + S_x\tan(S_\alpha)}{\tan(S_\alpha - \tan(D_\alpha))} \tag{A.6}
$$

$$
\begin{aligned}
y_{20} = & \frac{D_y - D_x\tan(D_\alpha) - S_y + S_x\tan(S_\alpha)}{\tan(S_\alpha) - \tan(D_\alpha)} \\
& \times \ \tan(S_\alpha) + S_y - S_x\tan(S_\alpha)
\end{aligned} \tag{A.7}
$$

Then, we calculate the intersection point coordinates for all possible values of S_x and D_x between 0 and a, S_y and D_y between 0 and b, and S_α and D_α between 0° and 90°, while checking the unreachability constraints (A.2)-(A.5). By running through all possibilities, we calculate the ratio of the number of S-D pairs satisfying the unreachability constraints and the total possible number of S-D pairs, which is the unreachability probability.