

**RESILIENT AND EFFICIENT GRADIENT-BASED
COORDINATE ROUTING IN WIRELESS MESH
NETWORKS**

by

Jai-Jin Lim

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in The University of Michigan
2007

Doctoral Committee:

Professor Kang G. Shin, Chair
Associate Professor Mingyan Liu
Assistant Professor Jason N. Flinn
Assistant Research Scientist Daniel L. Kiskis

© Jai-Jin Lim 2007
All Rights Reserved

*To Soyoung, Seoyeon, and Junhyun,
with love and thanks*

ACKNOWLEDGMENTS

If we knew what it was we were doing, it would not be called research, would it?(Albert Einstein). We shall not cease from exploration. And the end of all our exploring will be to arrive where we started and know the place for the first time (T.S. Eliot). If people never did silly things, nothing intelligent would ever get done (Ludwig Wittgenstein). Insanity: doing the same thing over and over again and expecting different results (Albert Einstein). The important thing is not to stop questioning. Curiosity has its own reason for existing (Albert Einstein). I wonder how close I come to that.

Never regard study as a duty, but as the enviable opportunity to learn to know the liberating influence of beauty in the realm of the spirit for your own personal joy and to the profit of the community to which your later work belongs (Albert Einstein). I wonder what my attitude was like and whether I become a man of value now.

My sincere gratitude to Prof. Kang G. Shin for updating me on numerous things during my entire graduate study. I was often overwhelmed by his passion that hardly withers away. But he looks like to hide his whole passion, only to emit a fraction of them and wait for me to rise to my own feet. This has comforted me much but often cornered me. Nevertheless, I feel comfortable with him since I put my story and thoughts down in front of him, and faith in him.

I'm also deeply indebted to the others, including Professor Mingyan Liu, Professor Jason Flinn, and Dr. Daniel L. Kiskis, for their advice, service, and patience.

Especially, thanks to Professor Mingyan Liu for serving my thesis committee even during her pregnancy and postpartum care periods; to Professor Jason Flinn for constructive comments during my prelim and thesis preparations; to Dr. Daniel L. Kiskis for steering me in the initial periods of my study.

A word of gratitude to all of the past and current members of Real-Time Computing Laboratory that share time and memory with me, including Young-June Choi, Sangsoo Park, Katharine Chang, Zhigang Chen, Min-gyu Cho, Yoo-Jin Choi, Xin Hu, Hyoil Kim, Kyu-Han Kim, Ashwini Kumar, Alex Min, Wei Sun, Chang-Hao (Howard) Tsai, Jian Wu, and Jisoo Yang. Fellowship with each and every member is an invaluable memory. Without this competent, good-natured group, this work would not have been completed. Special thanks to Deborah Van Hoewyk and Stephen Reger, who proofread this thesis.

A warm hug to numerous colleagues and friends across the campus and the globe. Grateful thanks to many people with Korean Bible Church for their spiritual and physical support: especially to Pastor Seung-Yoon Choi, the elders, the deacons, the members of the Man's Prayer Group.

My utmost thanks to God for His incomparable riches of grace and forever praise to Him who is always at work for me. I am praying to Him that He continues to lead me through the rest of my life according to His plan.

TABLE OF CONTENTS

DEDICATION	ii
ACKNOWLEDGMENTS	iii
LIST OF FIGURES	viii
LIST OF TABLES	xiii
ABSTRACT	xiv
CHAPTER	
I. INTRODUCTION	1
1.1 Motivation	1
1.2 Contributions	2
1.3 Thesis Outline	7
II. MANY-TO-ONE GRADIENT-DESCENDING STATELESS ROUTING	12
2.1 Introduction	12
2.2 Related Work	16
2.3 Gradient-based Unicast Routing	19
2.3.1 Overview	19
2.3.2 Defining Height	20
2.3.3 Forwarding Packets	24
2.4 Lightweight Gradient-field Maintenance	25
2.4.1 Overview	25
2.4.2 Creating Gradient-Field	26
2.4.3 Maintaining Gradient-Field	27
2.4.4 Handling No-Connection	27
2.4.5 Multi-Path Approach	29
2.5 Evaluation	30
2.5.1 Over Unplanned Topology	34

2.5.2	Over Unplanned+Random Topology with a Mobile Sink	35
2.6	Conclusion	36
III. ONE-TO-MANY GRADIENT-ASCENDING STATELESS ROUTING		41
3.1	Introduction	41
3.2	Related Work	45
3.3	Membership-based Broadcast	47
3.3.1	Overview	47
3.3.2	Data Structure and Hash Functions	49
3.3.3	Footprint and Operators	50
3.3.4	Clocking Footprints $\mathbf{C}(\mathcal{B}_c)$	51
3.3.5	Footprint Loss Model	51
3.4	Stateless One-to-Many Gradient-Ascending Routing	51
3.4.1	Per-Sink Bloom Filters	52
3.4.2	Gradient-Field Construction	52
3.4.3	Footprints Accumulation and Loss Model	53
3.4.4	Forwarding Gradient-Ascending Packets	54
3.4.5	Reliable Membership-based Broadcast	55
3.5	Evaluation	57
3.5.1	Primitive Membership-based Broadcast	58
3.5.2	Impact of Unicast-like Retransmission	62
3.5.3	Impact of Gradient-Based Suppression	63
3.5.4	Impact of Maximum Counter	67
3.5.5	Impact of Loss Model	67
3.6	Conclusion	67
IV. ONE-TO-ONE GRADIENT-HEIGHT VECTOR ROUTING		70
4.1	Introduction	70
4.2	Related Work	73
4.3	Gradient-Height Vector Routing	75
4.3.1	Overview	75
4.3.2	Type of Coordinates	76
4.3.3	Distance Function	79
4.3.4	Greedy Forward	79
4.3.5	Recovery from Greedy Failure	80
4.4	Evaluation	81
4.4.1	Wireless Channel	81
4.4.2	Topologies and Flows	81
4.4.3	Performance Metrics and Questions	83
4.4.4	Main Results	84

4.4.5	Impact of Wireless Channel Variations	86
4.4.6	Impact of Greedy Routing Variations	87
4.4.7	Impact of Auto Rate Algorithms	88
4.4.8	Impact of Length-estimate	89
4.4.9	Impact of Placement and Number of Landmarks . .	90
4.4.10	Impact of Distance Function	91
4.5	Conclusion	91
V. MULTI-RATE OPPORTUNISTIC GREEDY ROUTING . .		98
5.1	Introduction	98
5.2	Related Work	101
5.3	Generic Opportunistic Greedy Forwarding	102
5.4	Derivation of Practical Multi-rate Metrics	104
5.4.1	Unpenalized Multi-rate Metric	104
5.4.2	Penalized Multi-rate Metric	105
5.4.3	On-line Estimation	107
5.5	Evaluation	108
5.5.1	Wireless Channel	108
5.5.2	Auto Rate Algorithms	112
5.5.3	Topologies and Flows	112
5.5.4	Performance Metrics and Questions	114
5.5.5	Main Results on Line Topology	115
5.5.6	Evaluation with Unplanned Non-uniform Topology .	118
5.5.7	Evaluation of Penalty Model: nmep vs. pmep	120
5.6	Conclusion	122
VI. CONCLUSIONS AND FUTURE DIRECTIONS		123
6.1	Conclusions	123
6.2	Future Directions	124
6.2.1	Performance Comparisons of Coordinate Routing vs. Topology-based Routing in Wireless Mesh Networks	125
6.2.2	Reward-Maximizing Opportunistic Handover in Wire- less Mesh Networks with Mobile Users	125
APPENDIX		127
BIBLIOGRAPHY		131

LIST OF FIGURES

<u>Figure</u>		
1.1	Converting a stateful tree to a stateless mesh	3
1.2	Thesis organization	8
2.1	GDSR converts a tree structure to a mesh structure and routes packets over the mesh structure in order to harness resilience and efficiency provided by multiple neighbors.	14
2.2	Gradient-based unicast routing uses height, rather than <code>nexthop</code> pointer. The “state” is a pair of <code><nexthop,up/down></code>	20
2.3	In GDSR, link failures are loosely-coupled with the gradient recovery. Dotted arrows represent such a loose-coupling.	21
2.4	After overhearing that node Y advertises its current height (H_Y) and height-determinant (D_Y), node Z updates its own H_Z and D_Z based on both the measured length-estimate (L_Z) and the advertised information (H_Y and D_Y).	23
2.5	Wireless channel variations during the simulation.	31
2.6	MistLab [51], a non-uniform real-world wireless sensor network tested with 61 nodes over a field of 16,000 square feet, is reproduced in our <i>ns-2</i> simulation as an <code>nsMistLab</code> 802.11b wireless network over a field of $1km \times 1km$. Additional randomly-deployed 100 nodes are omitted.	32
2.7	Performance evaluations of many-to-one over the unplanned <code>nsMistLab</code> topology. Results are presented with nodes ordered in their distance to the sink.	37
2.8	Impact of different height definitions: the hops, the extrapolated, and the cost (on the same minimum-cost path).	38

2.9	Performance improvements over the single-path approach. The values in a table is expressed in terms of the median (m) and the mean (u) performances in percentage (%).	39
2.10	Performance improvements over the single-path approach. The values in a table is expressed in terms of the median (m) and the mean (u) performances in percentage (%).	40
3.1	Classification of routing and interaction patterns with respect to the sink in a typical many-to-one routing scenario. The ‘height’ of a node in the gradient field can be any scalar value such as hop count, delay, and distance from that node to the sink.	42
3.2	FBF can be directly coupled with stateless routing or used as a low-level primitive to localize the route-discovery query. The hashed footprint for a packet en route is shown without a box, whereas the Bloom filter at each node is shown with a box. A membership test of “M” against networked Bloom filters is performed when sink “S” sends data to it. Two independent hash functions and a Bloom filter of length 5 bits are associated with each node. Each node selects its hash functions independently of others.	47
3.3	Routing performances under the grid topology with $n = 121, k = 1$. L represents the percentage of nodes that are pre-hashed to the filter. (!1,others) means that all but node 1 are sending data to the default sink located at (5,5).	59
3.4	Improved routing performances under membership-based broadcast with different retry limits $r = 0, 1, 2, 3$	62
3.5	Impact of suppression on routing performances when $n = 121, k = 1$	64
3.6	Impact of the suppression on routing performances for $n = 121, k = 1$. $H_s(i)$ is hop count from node i to sink s , which is indicated by “hop”.	65
3.7	Impact of the maximum counter (mc) on the routing performances under $n = 121, k = 1$	66
3.8	Impact of the random loss model on the routing performances under $n = 121, k = 1$	68
4.1	Illustration of one-to-one GHVR	75

4.2	A reference model for the various landmark-based virtual coordinates.	77
4.3	After overhearing that node Y advertises its current height (H_Y) and height-determinant (D_Y), node Z updates its own H_Z and D_Z based on both the measured length-estimate (L_Z) and the advertised information (H_Y and D_Y).	78
4.4	Illustration of recovery from a greedy failure.	81
4.5	MistLab [51], a non-uniform real-world wireless sensor network testbed with 61 nodes over a field of 16,000 square feet, is reproduced in our <i>ns-2</i> simulation as an nsMistLab 802.11b wireless network over a field of $1km \times 1km$.	82
4.6	The improved(+) or reduced(-) delivery ratios (%) of the greedy routing relative to the extrapolated coordinate over 10 different topologies. The m and u represent the median and the mean of comparison results. Node-by-node comparison results for the set 0 in (b) are plotted in (a).	85
4.7	The improved(+) or reduced(-) delivery ratios (%) of the greedy routing relative to the extrapolated coordinate over 10 different topologies. Node-by-node comparison results for the set 0 in (b) are plotted in (a). The wireless channels are changed according to (c).	93
4.8	Performances of the <i>connectivity-first greedy routing</i> with a line topology under the straightline, hops, and extrapolated coordinates. Note the scale of Y axis in (b).	94
4.9	The improved(+) or reduced(-) delivery ratios (%) of the <i>connectivity-first</i> greedy routing relative to the extrapolated coordinate over 10 different topologies.	95
4.10	The improved(+) or reduced(-) delivery ratios (%) of the greedy routing relative to the extrapolated coordinate when a variant of the best auto rate algorithm [28] is used.	95
4.11	The improved(+) or reduced(-) delivery ratios (%) of the <i>connectivity-first</i> greedy routing relative to the extrapolated coordinate when a variant of the best auto rate algorithm [28] is used.	96
4.12	The improved(+) or reduced(-) delivery ratios of the greedy routing relative to the extrapolated coordinate when its length-estimate is changed \sqrt{c} (default), c , c^2 where $c = \text{dlvrate}^{-1}$.	96

4.13	The greedy routing performance (in terms of the median value) relative to the extrapolated coordinate with the number of landmarks varied.	97
4.14	The greedy routing performance (in terms of the mean value) relative to the extrapolated coordinate with the number of landmarks varied.	97
4.15	The delivery ratios of the greedy routing under different distance functions with the ideal path-distance coordinate.	97
5.1	Parameters in an abstracted multi-rate multi-hop wireless network.	103
5.2	Tradeoffs of throughput vs. distance for several bit-rate schemes in the simulated IEEE 802.11b. Labels in the x-axis represent the distance in meters. The plot for throughput vs. distance (1 st row) is decomposed into two sub-plots; bit-rate vs. distance (2 nd row) and delivery ratio vs. distance (3 rd row).	110
5.3	“Panoramic” evaluations by staging out different simulation configurations during the entire evaluation.	111
5.4	MistLab [51], a non-uniform real-world wireless sensor network testbed with 61 nodes over a field of 16,000 square feet, is reproduced in our <i>ns-2</i> simulation as an nsMistLab 802.11b wireless network over a field of $1km \times 1km$	113
5.5	Performance evaluations in isolation. See Algorithm 7 for fwdrate , bwdrate , and dlvrate	115
5.6	Routing performance on the line topology with fixed 11 Mbps, which is the default data rate for 802.11b, and several auto rate algorithms. Labels in the x-axis represent the stage number in Figure 5.3. . . .	116
5.7	Throughput improvement on the line topology by each component of MuGR	118
5.8	Throughput improvement on nsMistLab by each component of MuGR . SPEED/ELD uses a link metric that is updated solely on data being unicast.	119
5.9	E2e delay improvement on nsMistLab by each component of MuGR . SPEED/ELD uses a link metric that is updated solely on data being unicast.	119

5.10	Relative throughput improvement by pmep (over nmep under the line topology).	121
5.11	Relative e2e delay improvement by pmep (over nmep under the line topology).	121
5.12	Relative throughput improvement by pmep (over nmep under nsMist-Lab).	121
5.13	Relative e2e delay improvement by pmep (over nmep under nsMistLab).121	

LIST OF TABLES

Table

2.1	Configurations for 802.11b with the shadowing model.	31
3.1	Configurations for 802.11b with the shadowing model.	58
4.1	Configurations for 802.11b with the shadowing model.	83
5.1	Configurations for 802.11b with the shadowing model.	109

ABSTRACT

RESILIENT AND EFFICIENT GRADIENT-BASED COORDINATE ROUTING IN WIRELESS MESH NETWORKS

by

Jai-Jin Lim

Chair: Kang G. Shin

We first study how to make routing protocols *resilient* against opportunistic or accidental switching in wireless mesh networks: opportunistic switching uses a new link to improve performance, whereas accidental switching uses a new link to recover from unexpected failures. By using resilient routing protocols, packets can be routed reliably without necessarily triggering route update or recovery upon each link switching.

We have designed three resilient routing protocols, covering many-to-one, one-to-many, and one-to-one communications in wireless mesh networks. They all eliminate the explicit *route state*, $\langle \text{nexthop}, \text{up/down} \rangle$, for a given destination. Therefore, any event changing *nexthop* or *up/down* does not trigger route update or recovery in the proposed routing protocols.

To accomplish resilient many-to-one routing to a particular gateway without the explicit route state, a gradient field around that gateway is formed in the network. In this gradient field, each node determines its height with respect to the gateway. The determination of the height relative to the gateway is called a *gradient-based virtual coordinate assignment*, since the height is also used as a coordinate with respect to the gateway.

In the proposed one-to-many routing protocol, routing from the gateway to a specific node or set of nodes is carried out using membership-based broadcast. Membership-based broadcast does not require any explicit route state from the gateway to other nodes. Instead, it is bandwidth-efficient broadcast that exploits the histories of packet forwarding to the gateway. Each node uses a Bloom filter to maintain the packet-forwarding histories.

In the proposed one-to-one routing protocol, each node has the multiple heights relative to each of the pre-selected gateways. These multiple heights of a node are treated as the multi-dimensional geographic locations of that node. Packets between two nodes are then routed without flooding the route request to find a route or maintaining periodically a route between the two nodes.

After designing three resilient gradient-based coordinate routing protocols, we further study how to develop an *efficient* packet-forwarding strategy in the physical or virtual coordinate space. Specifically, we seek a forwarding strategy that exploits the diversity of neighbors capable of using different data rates, channels, or radios. Since coordinate routing does not maintain any explicit route state, it can exploit such neighbors opportunistically and aggressively to maximize throughput without any overhead at the network layer. However, making a switch from one neighbor to another entails cost in delay or overhead at the physical or medium access control

layers. Therefore, we have developed an algorithm that finds a switching order among multiple neighbors that minimizes the switching cost.

CHAPTER I

INTRODUCTION

1.1 Motivation

Wireless networks have become increasingly popular, both for various purposes and on different scales. They range from the existing nation-wide 2G or 3G cellular wireless networks for voice calls and low-rate data traffic to emerging wireless networks for high-rate data traffic such as municipal wireless mesh networks, indoor wireless local area networks, or even application-oriented wireless sensor networks.

Ignoring the non-technical issues involved in the co-existence of heterogeneous wireless networks (e.g., network ownership), a mix of networks enables users to opportunistically switch to the best link available to improve the throughput or meet the application-level quality of service. Even focusing solely on IEEE 802.11-based wireless mesh networks, neighbors can use different data rates, channels, or radios because of the technological advances that enable multi-channels, multi-radios on a single wireless node.

The potential advantages of using different wireless links as needed call for a new design principle of cross-layer optimization between the vertical layers of a networked system, to the extent that users at the medium access control or network layers can opportunistically switch to a new link by changing such link configurations as data

rate, channel, or radio to maximize the network throughput or capacity. From the standpoint of a routing protocol, such opportunistic switching is equally viewed as accidental switching against unexpected failures, which usually triggers an expensive route recovery procedure at the network layer.

Unfortunately, most of the current wireless routing protocols are tightly-coupled with the route state $\langle \text{nexthop}, \text{up/down} \rangle$. If a route going through a particular link is diagnosed as being down, a new route discovery procedure is invoked to get around that broken link. However, the route recovery procedure usually requires to flood the route recovery packets in the network, thereby incurring high overhead and wasting bandwidth. Thus, efficient routing with little or no maintenance overhead needs to be designed to improve network capacity.

The goal of this thesis research is to devise a methodology that hides a routing protocol from the effects of opportunistic or accidental switching as much as possible. Any expected benefits from opportunistic switching below the network layer will otherwise be offset by the overhead associated with the reconfiguration or the rediscovery of routes between nodes. At the same time, we would like to allow the routing protocol to exploit opportunistic switching at the network layer to detour a broken link or to improve routing performance.

1.2 Contributions

To be resilient against opportunistic or accidental switching, a routing protocol must avoid maintaining any explicit route state between nodes to the greatest extent possible. The protocol is thus relieved of the reconfiguration or recovery overhead associated with opportunistic or accidental switching. The need for maintaining the explicit route state is eliminated by constructing a gradient field on the network with

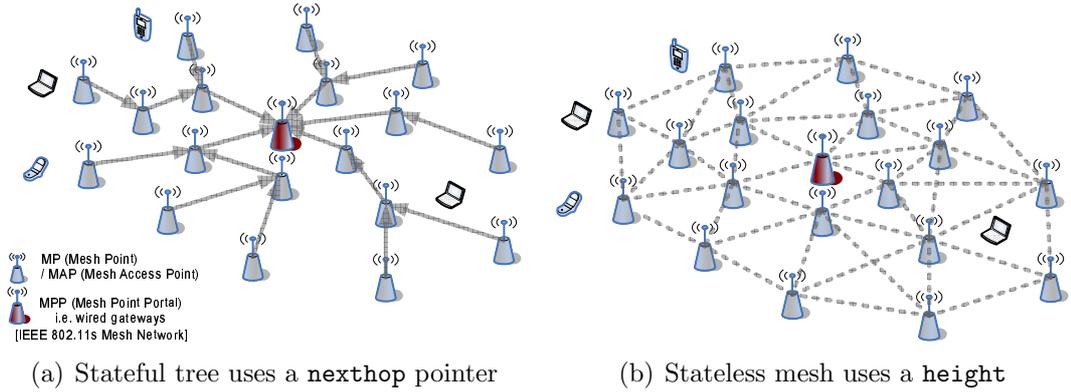


Figure 1.1: Converting a stateful tree to a stateless mesh

respect to one or more nodes called *landmarks*. Each node determines its height in relation to each of the elected landmarks. The determination of height relative to a particular landmark is called a *gradient-based virtual coordinate assignment*, since the height is also used as a coordinate with respect to that landmark. We then run the standard geographic routing on the constructed gradient-based coordinates as if the coordinates represent the actual geographic locations.

Our first contribution is the design of many-to-one Gradient-Descending Stateless Routing (GDSR). Taking Figure 1.1 as an example of wireless mesh networks, typical scenarios for GDSR occur when mesh points (MPs) / mesh access points (MAPs) want to deliver packets to a wired Internet gateway known as mesh point portal (MPP).

In Figure 1.1(a), every node maintains an explicit `nexthop` pointer toward the wired Internet gateway. This is equivalent to maintaining a spanning tree rooted at that gateway. Whenever a node finds that its `nexthop` pointer has become invalid due to a link failure, it must invoke a route recovery procedure to find a valid `nexthop` pointer. This way, the spanning tree is maintained.

GDSR does not use the stateful `nexthop` pointer and replaces it with a `height`

with respect to the gateway. The height does not contain any explicit route state; it does not imply the next hop nor whether the next hop is reachable.¹ Instead, it just represents how far a node is away from the gateway, or how high a node is with respect to the gateway in the gradient field. Within the gradient field, neighbors at a lower height than the current node are all eligible to forward packets. This creates the stateless forward mesh that allows multiple forwarders at each node, as shown in Figure 1.1(b). GDSR selects the next hop among multiple neighbors that makes the most efficient progress toward a destination.

A linear-extrapolation technique is proposed to determine the height of a node in the gradient-field. The height, or the gradient-based virtual coordinate, determined via linear extrapolation is here called the *extrapolated coordinate*. The height of a node is defined as the length of the minimum cost path from that node to the gateway. The length of a path is the sum of lengths of links along the path, and the length of a link is geographic distance between the two end points of the link. The proposed linear extrapolation estimates the length of a link based on the quality of the link, which is periodically measured between neighboring nodes. The extrapolated coordinate can be changed dynamically according to local channel conditions.

Our second contribution is the development of one-to-many Gradient-Ascending Stateless Routing (GASR). It addresses how to support the routing of packets from the gateway to a specific node or a set of nodes in the network. Again with Figure 1.1, typical scenarios for GASR occur when an MPP sends data back to one or more MPs/MAPs. The major constraints to GASR are that MPs/MAPs know how to route packets to MPP only in the gradient-descending direction, and that there is

¹Note that in a strict sense, the height is also a state kept on a node. As explained above, it, however, does not contain any route state or information. Thus, “stateless routing” in this research needs to be narrowly interpreted in lieu of whether the defined route state or information is explicitly required by a routing protocol.

no route discovery or maintenance in the gradient-ascending direction.

To deal with this issue, GASR is built on novel *membership-based broadcast* with a Bloom filter on each node. Each node maintains its Bloom filter independently of other nodes and hashes into the filter the source address of gradient-descending packets that go through the node. Membership-based broadcast allows the gateway to specify the intended recipients and is relayed by any intermediate node that finds any of those recipients specified in the broadcast to be in its filter. GASR tries to strike a balance between the number of broadcasts and a delivery ratio by developing both *gradient-based suppression* and *unicast-like retransmission* techniques.

Both GDSR and GASR complete bidirectional stateless routing with respect to a particular gateway, but they are not sufficient to offer general one-to-one routing between two nodes. Our third contribution is therefore the development of one-to-one Gradient-Height Vector Routing (GHVR). It supports the routing of packets between any two nodes in the network by using multiple heights of a node with respect to a set of reference landmarks. Landmarks do not have to be wired gateways. They can be any pre-configured node that generates and sends the periodic beacons into the network; multiple gradient-fields are created at the same time via the linear extrapolation used in GDSR. Thus, every node can be provided with global gradient-based coordinates with respect to those elected landmarks.

GHVR runs the standard geographic routing by making it believe that the given global gradient-based coordinates are the actual geographic locations in a multi-dimensional space. Unlike other gradient-based coordinate routing that uses the discrete hop count to the selected landmarks as virtual coordinates, GHVR uses a *continuous* coordinate in that the linear extrapolation it relies on yields a coordinate represented in a real number, not an integer (e.g., hop count). Compared to the

discrete coordinate that uses hop count, the fine-grained continuous coordinate in GHVR turns out to not only reduce the number of landmarks required, but also improve significantly the packet- delivery ratio.

So far, we have focused on building and exploiting an artificial gradient field with respect to a single gateway or multiple landmarks. GDSR, GASR, and GHVR are all driven by a single important goal: elimination of the explicit route state and creation of a mesh structure resilient against link or node failures. They yield the same stateless forwarding mesh if the geographic locations of nodes were available.

To the network layer, it does not matter whether coordinates are true geographic locations or gradient-based virtual coordinates. The biggest advantage of coordinate routing can be preserved once nodes are provided with their physical or virtual coordinates. That is, nodes can opportunistically choose the next hop on a per-packet basis without incurring any route rediscovery overhead. Moreover, underlying opportunistic channel or radio switching algorithms below the network layer can be relieved of any burden of route reconfiguration.²

Our last contribution is the development of Multi-rate Opportunistic Greedy Routing (MuGR). It addresses how to develop an *efficient* packet forwarding strategy that can exploit the diversity of neighbors capable of using different data rates, channels, or radios. Since coordinate routing does not maintain any route, it can exploit such neighbors opportunistically and aggressively to maximize the throughput. However, making a switch from one neighbor to another incurs delay or overhead at the physical or the medium access control layer. Therefore, we propose an algorithm for finding a switching order among multiple neighbors that keeps the cost of that

²When different channels are allowed on nodes, we assume that there exists a common channel to disseminate the channel or radio selection information among neighbors. In such a case, that common channel is used to construct a gradient field.

order as low as possible.

In summary, this research develops a suite of resilient and efficient routing protocols with the gradient-based coordinate information for 802.11-based wireless mesh networks. Typically, these networks are composed of both the wired Internet gateways and wireless nodes equipped with multi-channel, multi-rate, multi-radio capabilities. The proposed coordinate-based routing protocols make the network layer resilient and efficient in the presence of opportunistic or accidental switching below the network layer by both eliminating the explicit route state between nodes and allowing the network layer to opportunistically choose the next hop. The approaches developed through this research can be easily applied to other types of wireless networks with similar characteristics.

1.3 Thesis Outline

The proposed routing protocols can be viewed as a single framework, shown in Figure 1.2. We first present three stateless routing protocols that use the gradient-based virtual coordinates. Their common feature is the use of a gradient-field to make them resilient against link or node failures. We then address how to increase the efficiency of packet forwarding in either the physical (geographic) or virtual (gradient-based) coordinate space.

Many-to-One Gradient-Descending Stateless Routing. We study the routing of packets from many nodes to the gateway in wireless mesh networks, and present a novel protocol called GDSR. It addresses two goals. The first goal is to make a routing protocol resilient against link or node failures. The second goal is to make the protocol achieve not only the long-term optimality of the minimum-cost routing, but also its adaptability to local channel conditions on

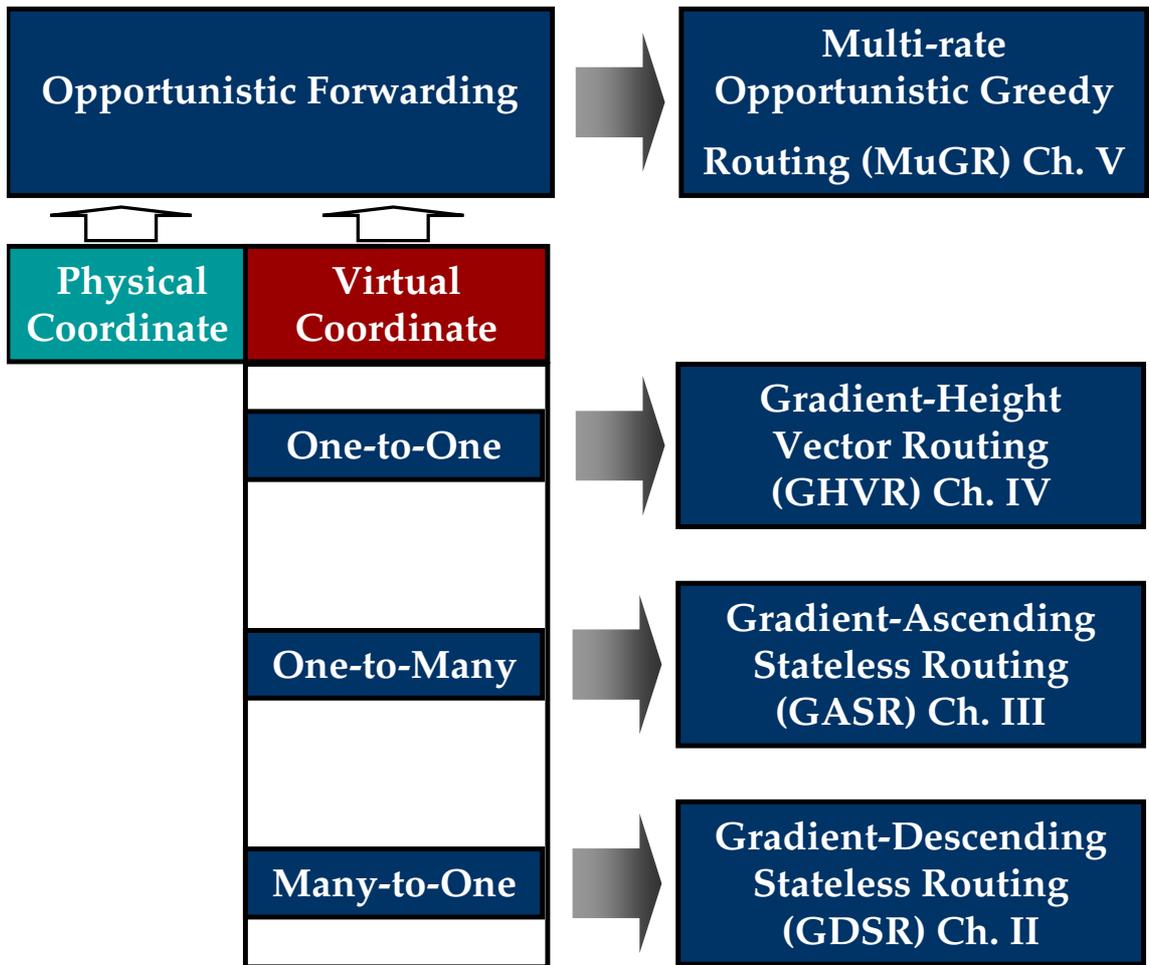


Figure 1.2: Thesis organization

a short-term scale, e.g., at the time of a few packet transmissions. GDSR meets these goals by constructing a gradient-field around the gateway and eliminating any route state between nodes. It is also built on a lightweight gradient-field maintenance protocol that is triggered only when no neighbor at a lower cost is found nearby. Our simulation using *ns-2* demonstrates that GDSR with our gradient-field definition heuristics is a viable approach to accomplishing the aforementioned goals.

One-to-Many Gradient-Ascending Stateless Routing. We study the routing of packets from the gateway to a specific node or set of nodes in wireless mesh networks (*gradient-ascending routing*), especially when nodes only know how to route packets to the gateway (*gradient-descending routing*). In other words, gradient-descending routing offers *no* route for gradient-ascending packets and is thus unidirectional in that sense. Given unidirectional gradient-descending routing, we propose novel one-to-many GASR, which does not depend on any on-demand route discovery or on flooding for routing packets in the gradient-ascending direction. Central to GASR is the use of membership-based broadcast with a Bloom filter on a node that keeps track of the source address of gradient-descending packets. Membership-based broadcast allows the gateway to specify the intended recipients and is thus relayed by any intermediate node that finds any of those recipients to be in its filter. GASR tries to strike a balance between the number of broadcasts and a packet-delivery ratio by developing both gradient-based suppression and unicast-like retransmission techniques. Our simulation using *ns-2* demonstrates that GASR with a reasonably-sized filter reduces significantly, by a factor of 5, the number of broadcasts when compared to flooding.

One-to-One Gradient-Height Vector Routing. We study the routing of packets between any nodes by using a set of reference landmarks already installed in the network. Nodes in the network will be assigned multiple gradient-based coordinates, which are heights with respect to the given set of landmarks. The proposed routing algorithm is called **GHVR**. Unlike similar approaches that use the discrete hop count to the landmarks, it uses fine-grained *continuous* heights, represented as real numbers. Our simulations using *ns-2* demonstrate that **GHVR** requires fewer landmarks and improves the routing performance significantly (by almost a factor of 2) over that of the discrete hop-count coordinate.

Multi-rate Opportunistic Greedy Routing. We study efficient forwarding strategies in coordinate routing for multi-hop wireless mesh networks. These networks comprise nodes capable of using different data rates or channels, depending on wireless channel conditions. In general, coordinate routing can exploit such neighbors opportunistically and aggressively without any route maintenance overhead, thanks to its stateless property. Nevertheless, making a switch from one neighbor to another incurs delay or overhead below the network layer. When faced with neighbors with different data rates or channels across different locations in the coordinate space, one fundamental question is whether there is any packet forwarding strategy that can exploit the diversity of neighbors so as to maximize the throughput while keeping the neighbor-switch cost as low as possible. We devise such a strategy by defining the generic reward and penalty model associated with packet transmissions. Moreover, two practical link metrics are derived from it. Our simulation using *ns-2* demonstrates that **MuGR** with these derived link metrics effectively improves

throughput over the state-of-the-art link metric by 16–39% for the simulation settings we investigated.

CHAPTER II

MANY-TO-ONE GRADIENT-DESCENDING STATELESS ROUTING

2.1 Introduction

Many-to-one routing is a common routing pattern for many applications in wireless sensor or mesh networks. For example, a typical application in wireless sensor networks (WSNs) is concerned with the collection of data and events from a large number of sensors via a single base station. Also, wireless mesh networks (WMNs) are usually deployed to offer a community wireless connections to a wired Internet gateway. Since most communications in these networks happen from multiple sensor/mesh nodes to a sink (a base station or gateway), support for resilient and efficient many-to-one routing in the presence of link or node failures is crucial to their intended functionality.

In this chapter we propose *Gradient-Descending Stateless Routing* (GDSR), a novel many-to-one routing protocol that achieves resilient and efficient delivery of packets to the sink. GDSR achieves the resilience by offering *exploratory* multiple paths toward the sink. The exploratory multiple paths are discovered as packets move toward the sink; no prior setup is required to find multiple paths. Using the exploratory multiple paths, a route recovery procedure is triggered only when no

path to the sink is available, not every time a path is disconnected. On the other hand, GDSR achieves the efficiency by *dynamically* selecting the next hop that makes the most efficient progress toward the sink. The next hop is not set *a priori*, but determined dynamically, on a per-packet basis, among the multiple neighbors of each node.

Existing single-path [66], multi-path [14, 25, 64], or gradient-broadcast [57, 69, 70] many-to-one routing also tries to achieve resilient and efficient delivery of packets to the sink. The main problem with the single- or multi-path approach is the high overheads of maintaining paths against failures and/or sending multiple copies of a packet along multiple paths. It is also slow in reacting to changes in local links. The major problem with the gradient-broadcast approach is the difficulty of controlling the number of broadcasts to achieve a high delivery ratio. Unfortunately, existing gradient-broadcast protocols have not addressed this problem: the work in [57] doesn't present any way of controlling the number of broadcasts, whereas the work in [69, 70] reduces the number of broadcasts with "credit," determination of which still remains an open problem.

What distinguishes GDSR from existing single- or multi-path many-to-one routing is the construction of a multi-path mesh structure, and the development of an efficient packet-forwarding strategy over the mesh structure. Section 2.5 shows that GDSR outperforms these path-based protocols. Unlike the gradient-broadcast approach, GDSR is unicast-based and thus, free of redundant broadcasts. Moreover, the GDSR's gradient-field is built using the estimated length of the minimum-cost path, whereas the gradient-fields in [57, 69, 70] are based on the cost itself. Our evaluation of the impact of different definitions of gradient shows that the GDSR construction method is superior to the others.

forwarded to any node that makes the most efficient progress toward the sink on a per-packet basis. Moreover, failing to transmit packets to a particular next hop after the allowed maximum number of transmission attempts does not necessarily mean that the link associated with the next hop is broken, nor does it mean that all the routes to the sink are broken. In single- or multi-path-based many-to-one routing, a link failure is presumed after the allowed maximum number of transmission attempts, and then route update and recovery is triggered. However, GDSR translates every link failure to the degradation of a link’s quality, which lowers the efficiency of progress toward the sink via that link. As long as there exist next hops that make progress toward the sink, GDSR selects the link that makes the most efficient progress toward the sink.

GDSR is said to be *stateless*. Our definition of stateless or stateful routing is solely based on whether a routing protocol explicitly maintains the route state for each given sink. The route state is a pair of $\langle \text{nexthop}, \text{up/down} \rangle$, where `nexthop` is the next hop toward the sink and `up/down` indicates if the `nexthop` link is up or down. The mesh structure created by GDSR removes the need for maintaining this route state, although it is built with height.¹

The main contributions of this chapter are:

- A novel many-to-one routing protocol, GDSR, that offers resilient and efficient delivery of packets to the sink. It exploits redundant nodes in a node’s neighborhood to the maximum degree possible. Our simulation results confirm that GDSR achieves a higher delivery ratio with much reduced delivery overhead than single- or multi-path protocols.

¹Our definition of “state” in GDSR is limited to the route state alone, even though height can be thought of as a simplified state in a broader sense.

- Resilience to link/node failures by building a virtual gradient-field around the sink that yields multiple paths to the sink. To construct the gradient-field without using nodes' geographic locations, we propose a novel method that assigns each node a height relative to the sink. This method uses a linear extrapolation based on the quality of a link to estimate the length of a path, regardless of the definition of cost.
- Efficiency in delivering packets by selecting the next hop among multiple neighbors that makes the most efficient progress over the gradient field on a per-packet basis. Thus, it overcomes the problem of the single- or multi-path approach, i.e., slow reaction to changes in local links.
- Lightweight gradient-field recovery by entering the recovery mode only when no lower-cost neighbor can be found, thus reducing significantly the rate of triggering the expensive recovery mode.

The rest of this chapter is organized as follows. Section 2.2 discusses the related work. Section 2.3 describes how GDSR builds the gradient-field and uses it to forward packets. Section 2.4 describes how to construct an initial tree structure and use it to maintain the gradient-field. Section 2.5 comparatively evaluates GDSR via simulation. Finally, Section 2.6 concludes the chapter.

2.2 Related Work

Several protocols have already been proposed for achieving robust packet delivery to the data sink in the presence of node or link failures. Braided multipath [25] extends directed diffusion (DD) [32] by constructing alternative paths that are not necessarily node-disjoint with the primary path. Caching and multipath routing pro-

tocol (CHAMP) [67] achieves robust delivery by salvaging lost packets from the cache at intermediate nodes. Energy-aware routing (EAR) [64] and ReInForM [15] have similar features, where multiple paths to the sink are created and maintained by the sink. EAR selects probabilistically a next hop toward the sink, whereas ReInForM sends multiple copies of a single packet along multiple paths while controlling the number of redundant packets. In contrast with these protocols, GDSR chooses the next hop that makes the most efficient progress to the sink (or gateway).

Meshed multipath routing (M-MPR) [14] selects the next-hop node dynamically based on the neighbors's conditions. It aims to distribute traffic evenly among neighbors, thereby balancing the load among them. M-MPR is an on-demand routing protocol through which a source can form a meshed multi-path topology to a given destination. GDSR forms a similar mesh structure by using a lightweight gradient-field maintenance protocol. However, the gradient-field maintenance protocol used by GDSR solves several problems with the application of distance-vector routing [54, 55] to many-to-one communications.

The gradient-field can be built naturally or artificially. The best-known type of natural gradient-field is the one formed by using the geographic distance to a destination that defines the heights of nodes in the field. However, it requires knowledge of nodes' geographic locations. In case there is no natural gradient field, an artificial gradient-field can be built by defining a scalar field on the network. One of the most popular approaches to defining the scalar field on the network is to use hop count, delay, or cost along the minimum-cost path to the sink [57, 69, 70].

Once the gradient-field is set up, any node making positive progress in the gradient-descending direction is eligible to forward packets. GRAd [57] is a gradient-based broadcast protocol where packets are broadcast and any eligible neighbor de-

cides whether or not to relay them. However, this scheme is not efficient as it is likely to produce too many broadcasts regardless of the local channel condition.

Another gradient broadcast protocol, called GRAB [69, 70], overcomes this problem. It controls the number of broadcasts in transit with a *source-assigned credit* carried in the packet header, but how the source decides the credit remains an open problem. Even if the credit were determined somehow, GRAB is not adaptive to the local channel condition that packets en route to the sink experience. GRAB tends to be inefficient or unreliable depending on the credit. Actually, one of the evaluation results (Sec. 4.2.1 [70]) shows that GRAB is less robust when the packet-loss rate is high.

The idea of building a gradient-field by GDSR is similar to [57, 69, 70]. However, GDSR is unicast-based and adaptive to the local channel condition because the gradient-field is continually updated by the underlying link-quality measurement scheme. Another gradient-field-based routing protocol for the wired Internet was proposed in [4]. The authors of [4] use a link-state routing protocol to build the gradient-field.

Single-path with repair routing (SWR) [66] forwards packets along a pre-established single path to the sink. It has been shown to outperform GRAB, but its main drawback is that it must perform local route repair whenever a link breakage is detected. GDSR exploits multiple neighbors in the gradient-field to the greatest extent possible, and only runs a gradient maintenance protocol when no lower-cost neighbor can be found. The gradient maintenance protocol for GDSR adopts many-to-one on-demand distance vector routing (MODV) [49], which is efficient and lightweight single-path routing and has also been shown to outperform SWR.

Link-quality in wireless networks has been the focus of research over the last few

years to understand and characterize link-quality in wireless sensor networks [72] or mesh networks [1, 6, 12, 17, 18]. New link or path metrics have been proposed, including the expected transmission count (ETX) [12], the expected transmission time (ETT), the weighted cumulative ETT (WCETT) [18], and the effective number of transmissions (ENT) [41].

Even though these metrics are shown to be effective in improving the throughput, their broadcast-based measurements create mismatches between broadcast-based neighbor sets and unicast-based neighbor sets. For example, in 802.11-based wireless networks, broadcast is sent at the default rate of 2Mbps, whereas unicast uses the default rate of 11Mbps. This difference in data rate makes the range of broadcast much greater than that of unicast. The hybrid link-quality measurement scheme in GDSR recognizes such disparity and incorporates it in the calculation of link-quality. More recent development on accurate link measurement, e.g., efficient and accurate link-quality monitor (EAR) [38], also addressed a similar issue.

2.3 Gradient-based Unicast Routing

2.3.1 Overview

GDSR can be viewed as a cross-layer design between the network and the MAC layers, as illustrated in Figure 2.2. In general, disconnection of a link is detected via either consecutive losses of the periodic HELLO message or an explicit MAC-layer notification such as the IEEE 802.11 MAC-layer link failure.² When a link is found broken, usual unicast routing should find a valid `nexthop` pointer via a route recovery procedure that floods route-request packets or link-state updates. On the other hand, gradient-based unicast translates a link failure into the degradation of

²In GDSR, 1-hop HELLOs are periodically exchanged. This implies that a link failure is first reported before any node failure. Therefore, for the rest of this chapter only link failures are considered.

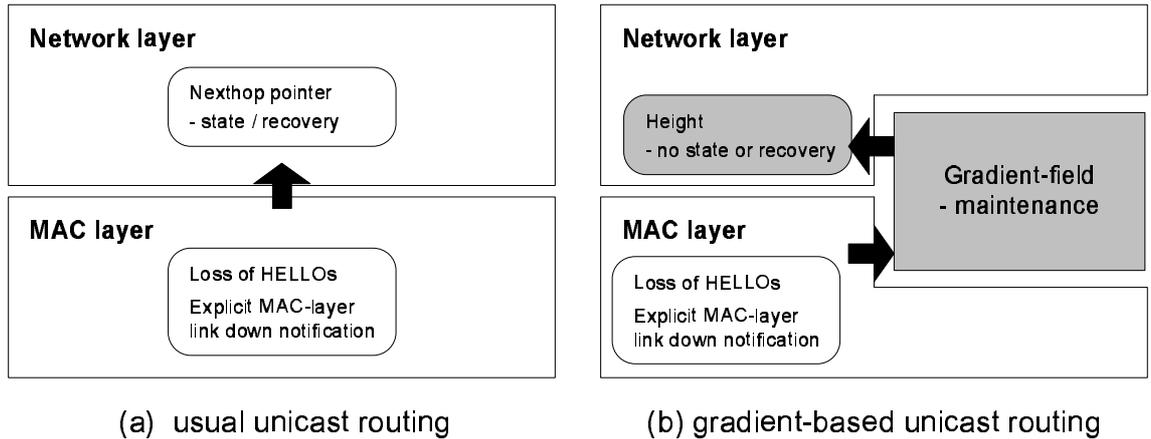


Figure 2.2: Gradient-based unicast routing uses height, rather than *next hop* pointer. The “state” is a pair of $\langle \text{next hop}, \text{up/down} \rangle$.

a link’s quality. The degraded link-quality triggers the gradient recovery only when no lower-cost neighbor can be found. It is referred here to as occurrence of the *no-connection* event to represent the case when no minimum-cost path is found.

Figure 2.3 details the interactions in Figure 2.2(b), illustrating the loose coupling of link failures with gradient recovery, i.e., not every link failure triggers gradient recovery. Two mechanisms, threshold-based neighbor exclusion and no-connection condition, prevent an immediate trigger. The threshold-based neighbor exclusion purges from the neighborhood a node whose bidirectional link-quality is less than a preset threshold. The no-connection condition triggers the gradient recovery when the minimum-cost path cannot be found after the neighbor exclusion.

2.3.2 Defining Height

The height of a node is defined to be the sum of link lengths along the minimum cost path from that node to the sink. We need to estimate a link’s length (*per-link length*) without using any geographic location. The per-link length can be estimated using the received signal strength. Using the received signal strength is

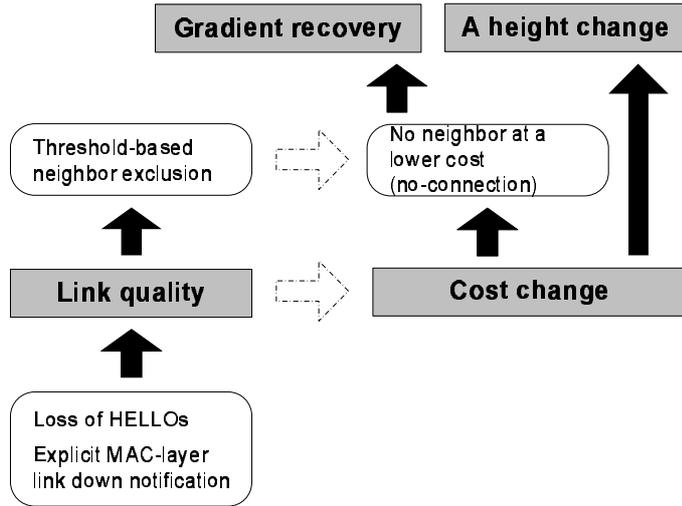


Figure 2.3: In GDSR, link failures are loosely-coupled with the gradient recovery. Dotted arrows represent such a loose-coupling.

intuitive since the signal strength decays over distance. As noted in [19], using the received signal strength method, however, needs to sample the environment to train the distance measurement system due to the lack of an accurate mathematical model of the environment. Also, it is noted that an estimation error cannot be improved significantly without using a complex model of the environment.

Other than alleviating these limitations of inaccurate estimation, improving the accuracy of a distance estimation method may not be worth the required complexity. Considering our intended use of the per-link length estimation for routing, any length estimation method that reflects proportional proximity among nodes will be sufficient. Moreover, we want to devise as simple a solution as possible for ease of implementation without modifying wireless network device drivers.

We develop a simple per-link length estimation method based on the link-quality that represents the packet reception ratio over a link. Given the link-quality (`dlvrate`), we assume that $\text{dlvrate} \propto 1/d^2$ with respect to distance (d). Our choice for this exponent is based on the fact that the received signal strength attenuates $1/d^2$ in

Input: an ETX probe piggybacked with the periodic HELLO and a locally-observed data transmission result.

Data: a link table \mathcal{L}

```

1  $l \leftarrow \mathcal{L}.lookup(nb)$ ;
2 if upon receiving an ETX probe  $e$  then
3    $l.update\_bwdrate()$ ; /*  $d_r$  */
4    $l.fwdrate \leftarrow e.bwdrate$ ; /*  $d_f$  */
5    $dlvrate \leftarrow l.fwdrate \times l.bwdrate$ ;

6 if upon receiving an ACK event  $a$  then
7    $dlvrate \leftarrow 1/a.txattempts$ ;

8 if upon receiving a FAILURE event  $f$  then
9    $dlvrate \leftarrow 1/(f.txattempts + 1)$ ;
10   $l.dlvrate \leftarrow \alpha \times dlvrate + (1 - \alpha) \times l.dlvrate$ ;
11  return;

12  $l.dlvrateFilter.put(dlvrate)$ ;
13  $l.dlvrate \leftarrow \alpha \times l.dlvrateFilter.get() + (1 - \alpha) \times l.dlvrate$ ;

```

Algorithm 1: Hybrid link-quality measurement used to estimate the per-link length.

the open space and that the received signal strength directly affects the receiver’s successful packet reception. A different curve fitting with other path exponents is possible. Our per-link length estimation may produce a lower resolution than an ideal method, but our evaluation in Section 2.5 shows that GDSR with the proposed method performs well, as compared to that with the actual length of a link.

Every node runs the link-quality measurement scheme described in Algorithm 1. This link-quality measurement scheme is based on the well-known ETX [12] which uses periodic broadcasts and is calculated per each link using the forward delivery ratio d_f and the backward delivery ratio d_r such that $ETX = (d_f d_r)^{-1}$. Since broadcast-based ETX does not reflect properly the link-quality for unicast, $dlvrate$ in Algorithm 1 adjusts the broadcast-based link-quality with each success or failure result of unicast.

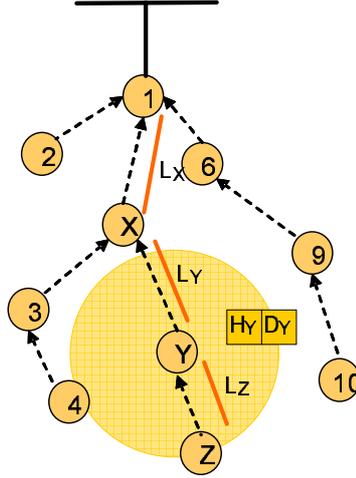


Figure 2.4: After overhearing that node Y advertises its current height (H_Y) and height-determinant (D_Y), node Z updates its own H_Z and D_Z based on both the measured length-estimate (L_Z) and the advertised information (H_Y and D_Y).

Given the assumption of $\text{dlvrate} \propto 1/d^2$, our per-link *length-estimate* from dlvrate between two nodes is given by

$$\sqrt{\text{dlvrate}^{-1}} \quad (2.1)$$

Our heuristics for determining the height or coordinate of a node with respect to a particular sink then takes an induction step over the minimum-cost path.³ For immediate 1-hop neighbors of the sink, their initial height between them and the sink is assigned their initial length-estimate to the sink. Referring to Figure 2.4, we assume that node Z has node Y as the next hop on the minimum-cost path to the sink (node 1), and that node Y broadcasts periodically its current height (H_Y) and height-determinant (D_Y), whose meaning will become clear below. Every neighbor around node Y then updates the link-quality between node Y and itself.

Besides updating the link-quality, Z updates H_Z and D_Z of its own since Y is the

³We assume that the link cost is dlvrate^{-1} , even though any definition of the link cost can be used.

next hop on the minimum-cost path from Z to the sink:

$$H_Z \leftarrow H_Y + L_Z H_Y / D_Y \text{ and } D_Z \leftarrow D_Y + L_Z \quad (2.2)$$

where L_Z is the length-estimate measured by Z for the link between Y and itself. Throughout this chapter, the height of a node determined this way is also called the *linear-extrapolated* height.

Every measurement needs to be smoothed out. Algorithm 1 is adaptive to the latest link-quality thanks to the large smoothing constant α and eliminates instantaneous noises in the measurement by using the *median filter* [53] (lines 11 ~ 12). Currently, α is set to 0.5 and a 3-span median filter that eliminates one spike in the input is used. Likewise, other measurements such as the length-estimate, the height, the height-determinant, and costs are also to be smoothed.

2.3.3 Forwarding Packets

Packets are forwarded over the gradient-field as follows. Given the height function H_s with respect to sink s , node c forwards packets to its neighbor $i \in Nbr(c)$ such that $i = \arg \max_{x \in Nbr(c)} (H_s(c) - H_s(x)) / C_{cx}$ and $H_s(c) > H_s(x)$, i.e., making forward progress. C_{cx} is the cost between nodes c and x . By definition, $H_s(s) = 0$. Each node will relay the received packets to its own next hop that satisfies the above condition or will drop them if such a next hop cannot be found. This forwarding strategy is called the *next-hop selection* rule.

If the height is linearly proportional to the minimum cost, both the above next-hop selection rule and the minimum-cost routing will choose the same next hop. This can be proved as follows. Suppose node c has the minimum cost C_c^s to sink s and $H_s(c) = \gamma C_c^s, \gamma > 0$. We also assume that c has selected i as the next hop to sink s according to the next-hop selection rule. Since i is selected by c as the

next hop, we know $(H_s(c) - H_s(i))/C_{ci} > (H_s(c) - H_s(x))/C_{cx} \forall x \in Nbr(c) - \{i\}$, implying $(C_c^s - C_i^s)/C_{ci} > (C_c^s - C_x^s)/C_{cx}$ from $H_s(c) = \gamma C_c^s$. If the next hop along the minimum-cost path from node c to sink s is not node i but node j , then $(C_c^s - C_i^s)/C_{ci} < 1$ and $(C_c^s - C_j^s)/C_{cj} = 1$. This contradicts the fact that $(C_c^s - C_i^s)/C_{ci} > (C_c^s - C_x^s)/C_{cx}, \forall x \in Nbr(c) - \{i\}$. Thus, node i should be the next hop along the minimum-cost path from node c to sink s .

The height H is not necessarily proportional to the minimum cost. If a link cost (C) is linearly proportional to the actual link length ($|\cdot|$), e.g., $C_{cx} = \alpha|cx|$, $\alpha > 0$, and the length-estimate (L) is also linearly proportional to the actual link length, e.g., $L_{cx} = \beta|cx|$, $\beta > 0$, H will be linearly proportional to the minimum cost. However, it should be noted that the linear relationship between the height and the cost is not necessary to improve routing performance; the linear relationship just shows how the minimum-cost path routing can be simulated with the next-hop selection rule. Our evaluation of the impact of different height functions on routing performance will show that our height definition outperforms the height definition that is linearly proportional to the minimum cost. This evaluation suggests that GDSR with the current height definition performs better than the minimum-cost routing with the same cost definition.

2.4 Lightweight Gradient-field Maintenance

2.4.1 Overview

The maintenance of a gradient-field uses an encounter record with the sink, which is conceptually the same as the destination-sequenced distance in DSDV [54] or AODV [55]. However, the maintenance protocol itself is fundamentally different from them. The *encounter* record is defined as a pair of $\{\mathbf{sinkseq}, \mathbf{sinkcost}\}$, where

`sinkseq` is a monotonically-increasing sequence number assigned by the sink, and `sinkcost` is the distance (in cost) to that sink. As it follows from the name, `sinkseq` defines a temporal order of nodes with respect to the sink: the larger `sinkseq`, the more recent encounter a node has with the sink. Likewise, `sinkcost` defines a spatial order of nodes with respect to the sink: the smaller `sinkcost`, the closer a node is to the sink. Therefore, the encounter defines a spatio-temporal order of nodes with respect to the sink.

2.4.2 Creating Gradient-Field

A sink starts by broadcasting a `SETUP` advertisement message. It has a tuple `{sinkaddr, sinkseq, sinkcost, lifetime, interest, ttl}`, where `sinkaddr` and `sinkseq` is an encounter record with the sink. The `sinkcost` and `interest` parameters are application-specific, which represent the cost to the sink and the sink’s interest, respectively. When `SETUP` is broadcast by the sink, `sinkcost` is set to 0. The collection network may be permanent or semi-permanent, depending on the value of `lifetime`: if this field contains a non-zero number, the network will disappear when the `lifetime` expires. Propagation of `SETUP` is bounded by the specified `ttl` (in hop count).

At every new `SETUP`, the sink makes a *self-encounter* record `{sinkseq=<uniq>, sinkcost=0}`, where `<uniq>` is a monotonically-increasing sequence generator. When a node receives a `SETUP` message `m` from node `prev`, it makes the sequential decisions as described in Algorithm 2. Lines 5 and 6 in the algorithm describes a rule called “freshness rule” in that a fresher encounter with the sink is preferred. The second rule appears at lines 7 and 8, representing a “short-cut rule” in that the encounter with a smaller cost to the sink is preferred. Both rules are called henceforth simply

Input: Received SETUP m from node `prev`

```

1  $m.sinkcost \leftarrow m.sinkcost + linkcost(this, prev);$ 
2 if a node has no prior encounter with the sink then
3    $\lfloor$  insert it and set sinksucc to prev
4 if an encounter record  $e$  with the sink is found then
5   if  $(e.sinkseq < m.sinkseq)$  then
6      $\lfloor e \leftarrow m.e$  and sinksucc  $\leftarrow prev$ 
7   if  $((e.sinkseq = m.sinkseq) \wedge (e.sinkcost > m.sinkcost))$  then
8      $\lfloor e \leftarrow m.e$  and sinksucc  $\leftarrow prev$ 
9 if  $m$  is not dropped then
10  $\lfloor$  rebroadcast it with  $m.sinkcost$ .
```

Algorithm 2: Creating a gradient-field

the SUCCESSOR-UPDATE RULE since they change `sinksucc` for a given sink.

2.4.3 Maintaining Gradient-Field

Nodes advertise their encounter with the sink via a 1-hop HELLO message. It has a tuple $\{\text{sinkaddr}, \text{sinkseq}, \text{sinkcost}, \text{lifetime}\}$, where `lifetime` is the remaining lifetime of the link associated with a successor toward the sink. Upon reception of HELLO, nodes apply the same SUCCESSOR-UPDATE RULE as in the reception of SETUP in Algorithm 2. The difference in handling two messages is that the received HELLO is not rebroadcast.

2.4.4 Handling No-Connection

Both SETUP and HELLO allow a node to find the minimum cost and the next-hop to the sink. Again, this does not mean that the state of the minimum-cost path is maintained. As illustrated in Figure 2.3, a gradient-recovery procedure is invoked only under the no-connection condition. Each node knows when it happens from the learned `sinkcost` in HELLOs or SETUPS.

If the no-connection happens, a node issues a local route request REQUEST. The

Input: Received REQUEST m from node $prev$

```

1 if an encounter record  $e$  with the sink is found then
2   if ( $self = m.sinkaddr$ ) then
3     send REPLY with a new self-encounter record
4   if ( $e.sinkseq > m.sinkseq$ ) then
5     send REPLY with  $e$ 
6   if ( $(e.sinkseq = m.sinkseq) \wedge ((e.sinkcost + m.srccost) \leq m.sinkcost)$ )
   then
7     send REPLY with  $e$ 
8   if my successor for  $m.sinkaddr \neq m.src$  then
9     unicast REQUEST to the successor for  $m.sinkaddr$  with
      $m.srccost \leftarrow m.srccost + linkcost(this, prev)$ 

```

Algorithm 3: Handling the no-connection condition

REQUEST message is in the form of $\{sinkaddr, sinkseq, sinkcost, src, srccost\}$, where src is the node that issues REQUEST, and $srccost$ is the cost to src . When REQUEST is broadcast by src , $srccost$ is set to 0.

Upon reception of a REQUEST message, immediate neighbors or multi-hops-away nodes make the sequential decisions as described in Algorithm 3. In the algorithm, there is a rule called ROUTE-REPLY RULE that determines whether to send REPLY or not. Note that REQUEST is being unicast toward the sink if there is no feasible successor in the immediate neighborhood and that REPLY is being unicast to the originator even though REQUEST is received via 1-hop broadcast. Such unicast solicitation on multi-hop neighbors for a sink is one of the key factors that reduce the overhead of recovering from the no-connection condition by exploiting path redundancy.

The REPLY message is in the form of $\{sinkaddr, sinkseq, sinkcost, lifetime\}$. While REPLY is being relayed to REQUEST. src , a reverse path cached during the propagation of REQUEST is used for the relay of REPLY. Each time REPLY is relayed, REPLY. $sinkcost$ is changed accordingly. In case multiple REPLYs are received, the

SUCCESSOR-UPDATE RULE is applied on every received message. Since REPLY is likely to carry the information of a new path to the sink, an intermediate node also applies the same SUCCESSOR-UPDATE RULE.

However, the originator of REQUEST may still receive no REPLY before its route discovery timer expires. In such a case, the originator invalidates an encounter record for the sink. Then, it enters the *probe* mode in which it sends HELLO with `sinkcost` set to ∞ at every probe interval, which should be long enough. The probe mode is stopped when either a node receives a new encounter advertisement from neighbors or its link expires normally after `lifetime`. Any upstream node using that node as a successor to the sink will, in turn, detect a link or node failure associated with that node. Then, a new local recovery procedure will be started automatically.

2.4.5 Multi-Path Approach

By using the periodic HELLO, we can build an alternative forwarding structure called a *multi-path* structure. We present this for the evaluation of GDSR only. This approach always maintains primary-backup paths. Algorithm 4 describes how each node maintains the primary-backup paths under the multi-path approach when it receives the encounter information. The binary operator \succ over a set of the encounters with respect to sink d is defined such that $\langle s_i^d, h_i^d \rangle \succ \langle s_j^d, h_j^d \rangle \Leftrightarrow ((s_i^d > s_j^d) \vee ((s_i^d = s_j^d) \wedge (h_i^d < h_j^d)))$, where s_i^d and h_i^d represent `sinkseq` and `sinkcost` node i has with respect to sink d , respectively. Similarly, the binary operator $=$ is defined as the equality operator over the encounters such that $\langle s_i^d, h_i^d \rangle = \langle s_j^d, h_j^d \rangle \Leftrightarrow (s_i^d = s_j^d) \wedge (h_i^d = h_j^d)$. Whenever the primary path is broken, the secondary path is used. If both paths are broken, a recovery procedure will be invoked with REQUEST.

Input: $e_n^d = \langle s_n^d, h_n^d \rangle$, an encounter record with sink d received from a neighbor n ; h_n^d is already accordingly updated to reflect the hop count or the cost from this node.

Data: a routing table \mathcal{R}

```

1 if  $\mathcal{R}.lookup(d)$  is empty then
2    $\mathcal{R}.add(d)$ ;
3  $r \leftarrow \mathcal{R}.lookup(d)$ ;
4 if  $\langle s_n^d, h_n^d \rangle = \langle r.sinkseq, r.sinkcost \rangle$  then
5    $r.succ \leftarrow r.succ \cup \{n\}$ ;
6    $r.succ.n.flag \leftarrow 0$ ;                                     /* primary */
7 if  $\langle s_n^d, h_n^d \rangle = \langle r.priv.sinkseq, r.priv.sinkcost \rangle$  then
8    $r.succ \leftarrow r.succ \cup \{n\}$ ;
9    $r.succ.n.flag \leftarrow 1$ ;                                     /* backup */
10 if  $\langle s_n^d, h_n^d \rangle \succ \langle r.priv.sinkseq, r.priv.sinkcost \rangle$  then
11    $r.succ.decrflag()$ ; /* primary becomes invalid, backup becomes
    primary */
12    $\langle r.sinkseq, r.sinkcost \rangle \leftarrow \langle r.priv.sinkseq, r.priv.sinkcost \rangle$ ;
13    $\langle r.priv.sinkseq, r.priv.sinkcost \rangle \leftarrow \langle s_n^d, h_n^d \rangle$ ;
14    $r.succ \leftarrow r.succ \cup \{n\}$ ;
15    $r.succ.n.flag \leftarrow 1$ ;                                     /* new backup */

```

Algorithm 4: Maintaining the primary–backup paths in the multi-path approach

2.5 Evaluation

GDSR will be evaluated against the single-path and multi-path approaches. To better evaluate them, several realistic simulation settings are used. First, we used the shadowing model recommended by [42] as a more realistic propagation model to reflect wireless link characteristics such as link asymmetry and time-varying link-quality. The *ns-2* shadowing model and other configurations are set up as in Table 2.1. These settings are derived from our implementation of [58] for a semi-open environment.

Second, the wireless channel settings are continually changed according to Figure 2.5, each lasting 500 seconds. The node constraint “–” means that any change

Table 2.1: Configurations for 802.11b with the shadowing model.

Path loss exponent		4	
Shadowing deviation		4,5,7,9	
Reference distance		10	
Frequency	2.4 GHz	Transmit power	15 dBm
CPTthresh	10	CSTthresh	-105 dBm
RXThresh (11 Mbps) [†]		-79.84 dBm, (95% at 50m)	
RXThresh (5.5 Mbps) [†]		-85.68 dBm, (95% at 70m)	
RXThresh (2 Mbps) [†]		-90.05 dBm, (95% at 90m)	
RXThresh (1 Mbps) [†]		-94.31 dBm, (95% at 115m)	
Data Rate		11 Mbps (w/o auto rate)	
Basic Rate		2 Mbps	
MAC headers		(24,14,14,28) bytes @ Basic Rate	
PLCP length		192 bits @ 1 Mbps	
SIFS	10us	DIFS	50us

[†] These thresholds are configured for the path loss exponent 4 and the shadowing deviation 4, and remain unchanged regardless of changes in both parameters.

	S1	S2	S3	S4	S5	S6
path loss	4	4	4	4	4	4
deviation	4	9	9	7	7	7
node	-	-	-	G0+G1	-	-
	S7	S8	S9	S10	S11	S12
path loss	4	4	4	4	4	4
deviation	7	5	5	5	5	4
node	-	G0+G1	-	-	-	G0

Figure 2.5: Wireless channel variations during the simulation.

will be global across all the nodes, whereas two node constraints G0 and G1 mean that a change will be applied to those nodes satisfying G0 or G1 constraint. Selection of nodes subject to both constraints is arbitrary. For example, G0 constraint is applied to those nodes whose IDs are 1 (modulo 3), whereas the G1 constraint is applied to those nodes whose IDs are 54, 20, 35, 24, 11, 51, 42, 41, 45, and 26. Lastly, nodes subject to the constraint G1 also drop the received packet randomly at the physical layer with probability 0.3.

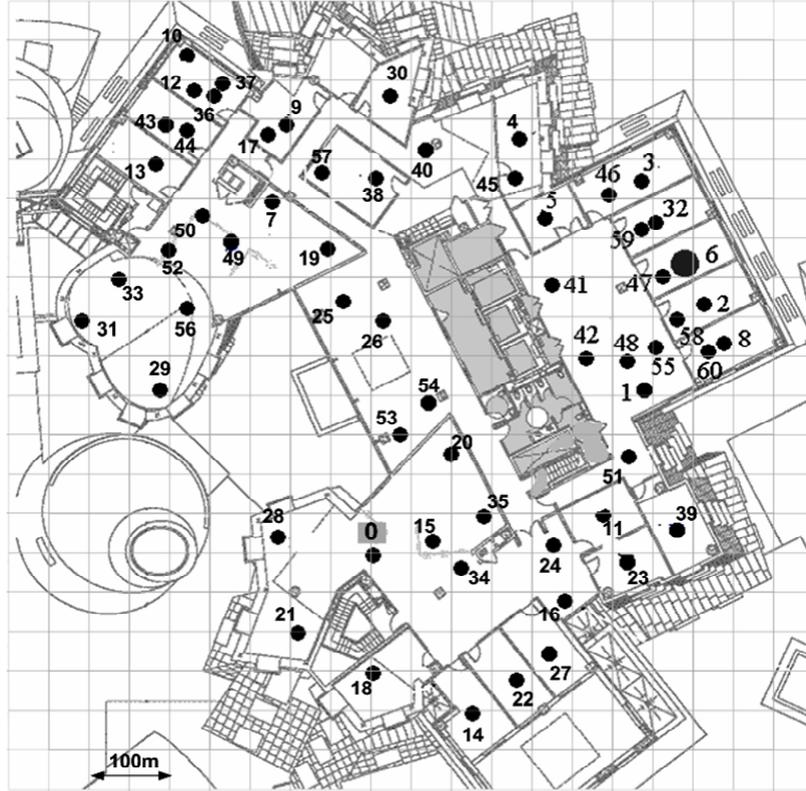


Figure 2.6: MistLab [51], a non-uniform real-world wireless sensor network test-bed with 61 nodes over a field of 16,000 square feet, is reproduced in our *ns-2* simulation as an nsMistLab 802.11b wireless network over a field of $1km \times 1km$. Additional randomly-deployed 100 nodes are omitted.

Given the time-varying link-quality, any encounter advertisement by neighbors via HELLO is ignored if it comes from a neighbor for which $dlvrate < 0.2$. By doing so, the chance of including instantaneous good links becomes small. The $dlvrate^{-1}$ between nodes is used as the link cost, representing the expected number of transmissions to succeed in transmitting a packet at the MAC layer.

Third, a topology simulating the real test-bed is also used. An MIT indoor wireless sensor network test-bed called MistLab [51] is reproduced as an unplanned topology in our *ns-2* simulation. The simulated MistLab, simply called here nsMistLab, has sixty-one 802.11b nodes spread non-uniformly over a square field as shown in Figure 2.6. This nsMistLab is used here for the default unplanned topology with

the default sink node 6.

The sink is sending `SETUP` in the beginning and every node except for the sink is sending continually an average MAC-level payload of 1500 bytes, or specifically $1500+U(-750, 750)$ bytes, to the sink. Each source is alternated at every 4 seconds in a round-robin fashion. Every node exchanges 1-hop `HELLO` periodically, which also piggybacks the link-quality probe as well as the encounter information with the sink. The same threshold-based neighbor exclusion is used for all protocols to make both path-based approaches more resilient to link failures. Additionally, packets with more than 40 transmission attempts at each hop will also be dropped — this number is arbitrarily chosen.

Given the simulation settings described so far, we measure several routing performance metrics. The delivery ratio represents the ratio of the total number of packets received by the sink to that sent by the source. But, the delivery ratio is measured without invoking any recovery procedure to assess the resilience of each approach to link failures only. Thus, a higher delivery ratio means that the considered approach is more resilient to link failures. Note that the single-path and multi-path approaches enter the recovery mode when all of their routes are broken, whereas `GDSR` does so when it detects the no-connection condition. Until the respective link failure is detected, packets will be routed normally. The delivery overhead, in terms of the ratio of the number of packets generated to the number of packets sent, is measured individually for each source. Note that the delivery overhead does not include the overhead of periodic `HELLO`s since all of the approaches use them.

2.5.1 Over Unplanned Topology

The delivery ratio in Figure 2.7(a) shows that **GDSR** outperforms the others across all nodes. Note that since **nsMistLab** is a sparse irregular topology, the results plotted in Figure 2.7 show many variations among nodes depending on their location in the network.

In terms of the e2e hop count, **GDSR** tends to traverse more hops. However, we should note that **GDSR** does not necessarily incur more overheads. Figures 2.7(c) and 2.7(d) show that despite longer paths it takes, **GDSR** still outperforms the others, in terms of the e2e delay and the delivery overhead. This is because **GDSR** is likely to choose a better next hop that makes an efficient progress among a larger number of next-hops.

We evaluate the impact of different height definitions. Other than the default linear-extrapolated height, both the number of hops and the minimum cost along the minimum-cost path are also investigated. Figure 2.8 shows the linear-extrapolated height outperforming other definitions. The hop-based height definition turns out to perform worst. This is due to the coarse-granularity of height definition, which does not tell which neighbors are making more progress toward the sink. On the other hand, the cost-based height definition shows the performance in-between. The major problem with it is that the cost is not suitable to reflect the linearity among nodes. For example, our link cost was dlvrate^{-1} , which generally has a non-linear relationship with the link length.

Since a link cost can be defined arbitrarily, using the cost directly as the node's height in gradient-based unicast routing is not a good idea. Instead, our heuristics try to extract the linear relationship among nodes on the minimum-cost path by using the link-quality, regardless of the cost definitions used.

2.5.2 Over Unplanned+Random Topology with a Mobile Sink

The goal of this evaluation is to investigate how the sink mobility affects the routing performance under each approach. The unplanned nsMistLab is augmented to have one mobile sink and additional 100 randomly-deployed nodes. The mobile sink moves around the field at the maximum speed of 20 m/s following the random waypoint model with the pause time 0.

Both Figures 2.9 and 2.10 indicate that GDSR still outperforms the others. The results represent the median (m) and the mean (u) performance improvements (%) relative to the single-path approach and are averaged over 10 different sets of randomly-deployed nodes.

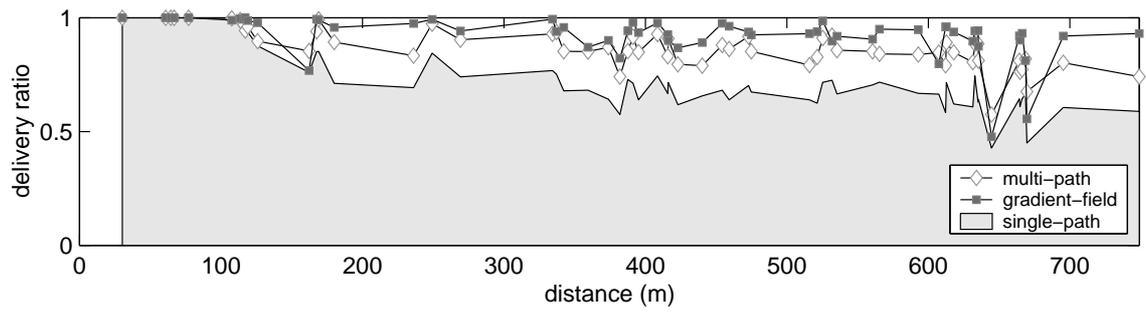
Two types of gradient-fields are also added for better comparisons. The first one is the *path-distance* gradient-field that represents the ideal extrapolated gradient-field since its length-estimate is replaced by the actual length. The actual length of a link is obtained with the assumption that the nodes' geographic locations are known. Given the same location assumption, a natural gradient-field can be formed with respect to the sink. The *locations* gradient-field refer to this.

The conclusion drawn from the previous evaluation over nsMistLab is also confirmed in this evaluation: GDSR still performs better than the path-based approaches. It is interesting to observe that the single-path approach performs better than the others in rare cases: those colored cells in Figures 2.9 and 2.10.

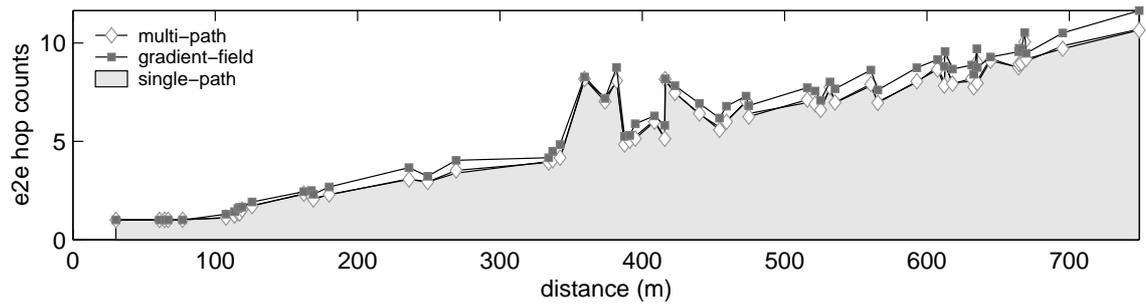
The evaluations of different types of gradient-fields with GDSR reveal that the ideal path-distance improves the delivery ratio of the extrapolated approach only marginally. This indicates that our heuristics for the length-estimate works well. Another interesting result is that routing with artificial gradient-fields is slightly better than the one with natural gradient-fields.

2.6 Conclusion

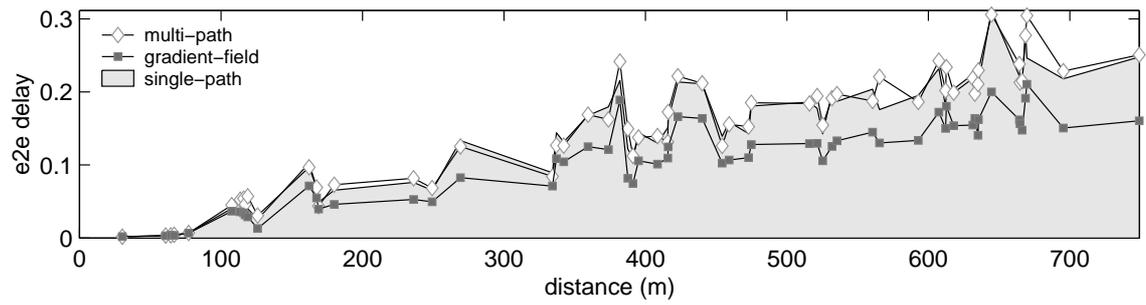
In this chapter we presented a new protocol, called GDSR, that offers efficient and resilient many-to-one routing in wireless networks. GDSR achieves both resilience and efficiency by eliminating the route state and building the gradient-field around the sink. Central to GDSR is a multi-path mesh produced by the gradient field, where the height of a node is defined as the estimated length of the minimum-cost path from that node to the sink. The estimation of the link length is based on link-quality measurements among neighbors. Our extensive simulation demonstrated that GDSR outperforms the other single- or multi-path many-to-one routing, while incurring much less delivery overhead. Finally, the proposed heuristics for estimating the link length are shown to be comparable to the ideal method when used with GDSR.



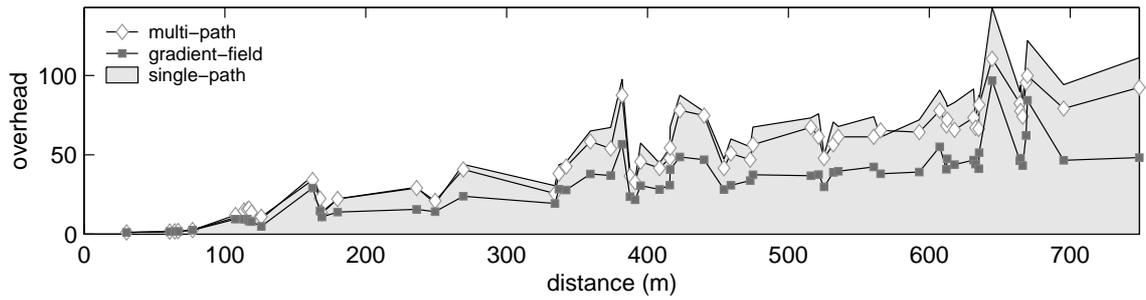
(a)



(b)



(c)



(d)

Figure 2.7: Performance evaluations of many-to-one over the unplanned nsMistLab topology. Results are presented with nodes ordered in their distance to the sink.

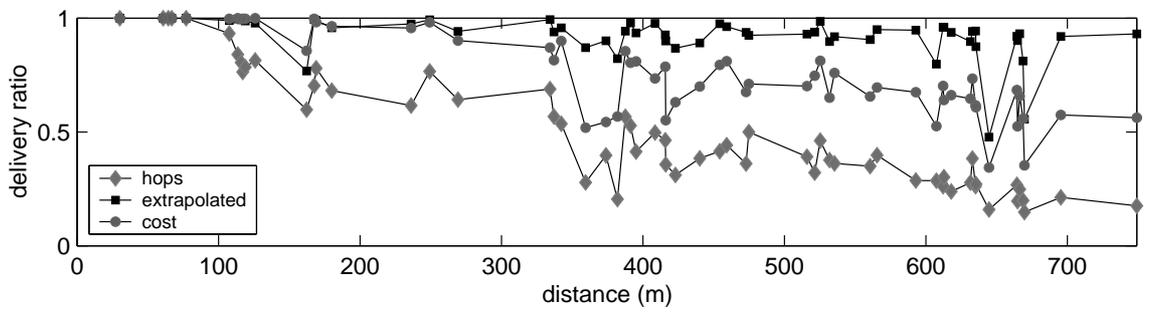


Figure 2.8: Impact of different height definitions: the hops, the extrapolated, and the cost (on the same minimum-cost path).

set	multi-path		gradient-field					
	m	u	extrapolated		path distance		locations	
			m	u	m	u	m	u
0	13.64	20.40	10.78	15.09	12.49	19.78	18.52	27.87
1	15.56	23.29	6.06	16.81	14.60	22.29	6.07	10.06
2	12.50	22.86	0.00	4.62	-1.82	-1.06	9.09	8.42
3	22.22	34.19	9.38	21.52	18.52	27.21	12.49	25.45
4	23.75	42.55	20.00	37.02	18.18	31.95	15.39	17.24
5	14.28	24.87	8.78	13.15	14.30	18.05	28.56	37.02
6	16.66	27.16	4.76	17.51	16.68	23.05	21.21	31.98
7	12.50	23.26	1.54	6.73	4.99	10.69	16.68	26.64
8	5.78	11.59	3.12	10.44	8.33	12.18	11.11	14.16
9	13.64	22.11	12.50	23.25	12.50	21.06	12.50	9.87

(a) delivery ratio

set	multi-path		gradient-field					
	m	u	extrapolated		path distance		locations	
			m	u	m	u	m	u
0	-0.56	0.70	1.73	5.04	5.56	8.94	6.04	11.02
1	-0.27	-0.06	2.04	7.55	6.67	12.38	2.72	9.29
2	0.44	5.91	0.34	2.64	2.83	5.92	3.48	8.74
3	1.82	14.97	0.83	8.15	7.46	20.16	7.25	16.76
4	1.72	3.47	4.17	5.66	6.82	9.52	0.00	-1.17
5	0.00	0.83	2.30	3.53	7.69	9.18	11.11	17.02
6	-0.64	3.64	2.81	3.20	8.05	9.44	10.44	15.67
7	0.00	0.97	0.45	2.77	4.65	5.54	5.16	5.40
8	-1.62	0.57	0.70	0.33	4.17	4.50	0.83	2.07
9	-1.16	1.73	5.45	8.49	8.96	12.44	5.62	-0.16

(b) e2e hops

Figure 2.9: Performance improvements over the single-path approach. The values in a table is expressed in terms of the median (m) and the mean (u) performances in percentage (%).

set	multi-path		gradient-field					
	m	u	extrapolated		path distance		locations	
			m	u	m	u	m	u
0	6.59	37.13	-24.78	10.30	-30.40	-14.19	-33.47	-10.87
1	-0.74	16.33	-22.14	16.88	-28.09	42.79	-28.47	113.99
2	-7.72	59.94	-22.28	-13.69	-29.34	6.83	-26.63	108.37
3	0.82	30.80	-27.15	-13.46	-28.00	-5.88	-28.90	-14.20
4	1.54	8.81	-28.67	-10.19	-29.19	-21.59	-37.31	-26.18
5	-0.36	15.44	-26.46	-17.08	-32.33	-26.79	-33.40	-23.62
6	4.88	39.88	-25.50	-19.64	-33.44	-20.85	-31.44	13.79
7	2.16	10.66	-17.36	-7.08	-22.33	-12.43	-21.10	-8.83
8	0.07	28.69	-26.36	-15.37	-25.75	2.96	-30.19	-17.68
9	2.44	19.73	-22.06	-13.97	-33.75	-25.43	-38.11	-32.92

(a) e2e delay

set	multi-path		gradient-field					
	m	u	extrapolated		path distance		locations	
			m	u	m	u	m	u
0	-7.43	6.05	-25.25	-17.75	-35.27	-26.76	-38.87	-26.93
1	-6.40	1.42	-24.67	73.86	-30.31	68.12	-18.34	121.30
2	-5.29	11.94	-16.73	-10.37	-16.05	9.07	-23.04	46.82
3	-2.61	12.70	-25.16	-17.59	-30.75	-10.54	-21.03	0.32
4	-13.92	-12.23	-35.38	-30.10	-35.32	-31.27	-38.73	-22.15
5	-7.71	-0.49	-23.54	-19.85	-30.51	-23.80	-43.09	-36.63
6	-9.87	1.94	-28.15	-19.26	-36.17	-20.24	-37.78	-28.98
7	-9.63	3.72	-13.74	-1.83	-21.58	-10.23	-26.65	-20.42
8	-3.95	6.19	-27.57	-18.27	-31.43	-23.20	-35.43	-25.22
9	-6.08	4.50	-30.08	-24.19	-38.89	-32.96	-34.06	-4.01

(b) overhead

Figure 2.10: Performance improvements over the single-path approach. The values in a table is expressed in terms of the median (m) and the mean (u) performances in percentage (%).

CHAPTER III

ONE-TO-MANY GRADIENT-ASCENDING STATELESS ROUTING

3.1 Introduction

Many-to-one routing is a common routing pattern for many applications in wireless sensor or mesh networks. For example, a typical application in wireless sensor networks is concerned with the collection of data and events from the sensor field through a base station. Also, wireless mesh networks provide a community multiple connections to the wired Internet gateways. Numerous many-to-one routing protocols [14, 25, 57, 64, 66, 69, 70] have been proposed to achieve resilient and efficient packet delivery to a ‘sink’ (i.e., a base station or gateway).

However, communications from many nodes to the sink alone are not enough; the opposite direction (from the sink to a specific node or a set of nodes, i.e., *one-to-many*) communications must also be supported. As illustrated in Figure 3.1, both directional communications can best be described as “gradient-descending” and “gradient-ascending” with respect to the sink. The term “gradient” between two nodes refers to the difference of “height” between them, where a node’s height can be any scalar value such as hop count, delay, or distance between itself and the sink. Examples of gradient-ascending interactions in wireless sensor or mesh

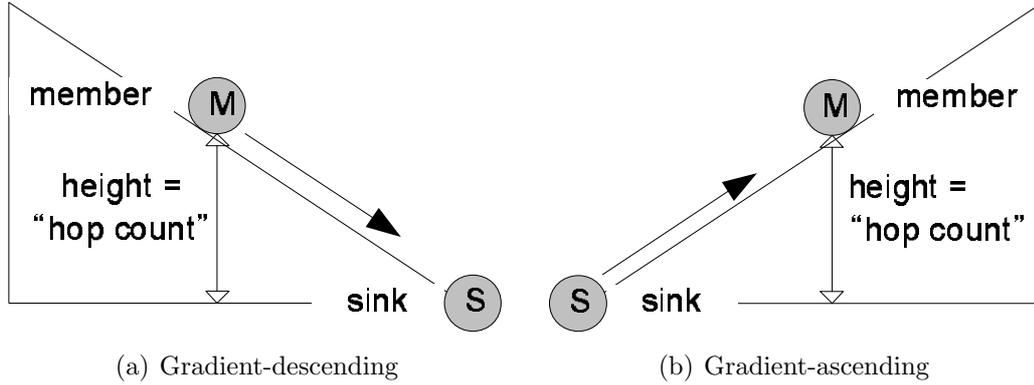


Figure 3.1: Classification of routing and interaction patterns with respect to the sink in a typical many-to-one routing scenario. The ‘height’ of a node in the gradient field can be any scalar value such as hop count, delay, and distance from that node to the sink.

networks are: (i) allocation, change, and migration of queries; (ii) reconfiguration, diagnosis, and summarization; and (iii) other signal-and-response type interactions.

This chapter addresses how to design efficient *one-to-many* gradient-ascending routing, especially when gradient-descending routing doesn’t provide any route information for gradient-ascending packets. For a given unidirectional gradient-descending routing protocol, one-to-many routing protocols should depend on neither on-demand route discovery nor simple flooding for routing packets in the gradient-ascending direction. Specifically, we would like to develop an efficient, stateless one-to-many routing protocol.

We say a routing protocol to be *stateless* if it does not create or maintain any route information. The route information specifies the next hop for each destination with the state indicating whether the next hop is up or down. By eliminating the need to maintain such route information at each node, a routing protocol is less sensitive and thus more resilient to link or node failures. On the other hand, an efficient routing protocol should not incur too much overhead. Any action that incurs

excessive routing overhead may also cause collision and contention, thus decreasing network capacity or data bandwidth. The efficiency becomes more important when the sink needs to interact with multiple (source) nodes.

Either using simple flooding that involves broadcast from every node in a network or running a separate stateful on-demand routing protocol such as AODV [55] or DSR [34] is not considered for the following two reasons. First, simple flooding is not bandwidth-efficient for one-to-many interactions unless one-to-all is required as holistic operations on the network, such as network programming over the air [29, 56]. Second, running a separate stateful routing protocol along with a stateful gradient-descending routing protocol is a daunting task because it is difficult to logically reconcile protocol conflicts between two different stateful routing protocols in case of common node or link failures. Moreover, it is not easy to create or maintain one-to-many connections to multiple nodes.

This chapter presents a novel one-to-many routing protocol called Gradient-Ascending Stateless Routing (GASR), given unidirectional gradient-descending routing such as typical tree-based routing. GASR is built on the novel membership-based broadcast with a Bloom filter on each node. Each node maintains the filter independently of others and hashes into it the source addresses of gradient-descending packets that pass through it. The membership-based broadcast makes the sink specify the recipients of broadcast and is relayed by any intermediate node that finds those recipients “recorded” in its own filter.

GASR has the following salient features.

- It is stateless and complements unidirectional gradient-descending routing protocols without creating any protocol conflicts.¹

¹Some readers may argue that GASR is not stateless since it has a state in the form of a Bloom

- It incurs a small space overhead by using the *Bloom filter* [7].
- It offers efficient one-to-many communications without the need to create and maintain separate routes to multiple recipients.
- It is bandwidth-efficient, thanks to the novel membership-based broadcast that employs gradient-based suppression to reduce the number of broadcasts.
- It is resilient to packet losses over a lossy wireless channel due to its unicast-like retransmission.
- It is resilient to node or link failures due to (likely) existence of multiple receivers in the network, i.e., the false positives associated with the membership test will allow multiple nodes to broadcast.

A location-based stateless routing algorithm such as GPSR [37] could be used, but it requires a location-acquisition component that GASR does not need. Therefore, GASR becomes robust to the absence or failure of a location-acquisition component. All that GASR requires is the accumulation of packet forwarding histories in the gradient-descending direction. Despite its importance in wireless sensor or mesh networks, making GPSR for one-to-many routing, especially with multiple recipients, has never been studied before.

In GASR, forwarding a packet is just performing the membership test of the recipients against networked Bloom filters that keep a dejavu image for other nodes whose packets have been relayed. While going through a series of networked Bloom filters, packets may be dropped or eventually forwarded to their destinations. One or more trails can lead to the recipients, being continually constructed and reinforced

filter and updates the filter. However, the filter does not contain any next-hop information. Thus, GASR is still stateless by our definition.

from gradient-descending packets.

The rest of this chapter is organized as follows. Section 3.2 describes the related work. Section 3.3 describes membership-based broadcast. Section 3.4 details how to design GASR on membership-based broadcast. Section 3.5 evaluates GASR extensively and comparatively. Section 3.6 conclude the paper.

3.2 Related Work

Many-to-one routing protocols have been studied extensively for wireless sensor networks. The authors of [32] proposed a data-centric routing algorithm where a data sink advertises its interest with a list of attribute name and value pairs, and nodes matching the interest subscribe to it. Propagating the interest neighbor-to-neighbor establishes gradients indicating the routing path and data-delivery rate to the sink.

Schurgers and Srivastava [62] proposed a gradient-based routing (GBR), as an extension to the directed diffusion [32], in such a way that the interest message records the number of hops taken during the propagation. The gradient is defined as the difference between a node’s hop count and that of its neighbors. Following the largest gradient, packets will traverse toward the sink via minimum-hop paths.

Poor [57] proposed an idea similar to GBR. Its difference lies in that instead of identifying which node to forward a packet to, a node broadcasts a packet with the “cost-to-destination” information, which was obtained opportunistically when the destination sends out its packet. Of all the nodes that receive the broadcast, only those that can deliver the packet at a lower cost will relay the packet.

Similarly, Ye *et al.* [70] proposed a robust data-delivery protocol, called GRAB, that uses the notion of “credit” for robust packet delivery over a gradient-descending

forwarding mesh. Once a sink broadcasts an advertisement packet (ADV) to set up a gradient cost field, packets from sources carry “credit” information to control a gradient-descending forwarding mesh for robust delivery.

On the other hand, on-demand routing protocols such as AODV [55] or DSR [34] find routes to destination by performing a flood-based extended ring search as needed. Despite potential reduction of the flooding overhead with query localization [11], their stateful routing property makes them incomparable with other stateful gradient-descending protocols. More important, using (on-demand) stateful protocols for gradient-ascending packets is against our design principle.

A Bloom filter is one of the important data structures often used in various areas of computer science to support a membership query with a small amount of storage space. We only examine key extensions to the original Bloom filter.

Li *et al.* [20] proposed *counting Bloom filters* that provide a mechanism to build a summary of the directory of cached documents at each Web proxy. A counting Bloom filter maintains a counter associated with each bit of the filter. Mitzenmacher [52] proposed *compressed Bloom filters* to reduce the number of broadcasts, the false-positive probability, and/or the amount of computation per lookup.

Rhea and Kubiawicz [60] proposed *attenuated Bloom filters*, an array of Bloom filters. Associated with each d^{th} -row Bloom filter is the summarization of replicas on the neighbor at d hops away. They use the proposed Bloom filters to enhance the performance of existing peer-to-peer location mechanisms when a replica for the queried data item exists closer to the query source.

Donnect *et al.* [16] proposed *retouched Bloom filters* that permit the removal of false positives at the expense of generating random false negatives. The proposed filters are used to implement a bandwidth-efficient protocol that sends information

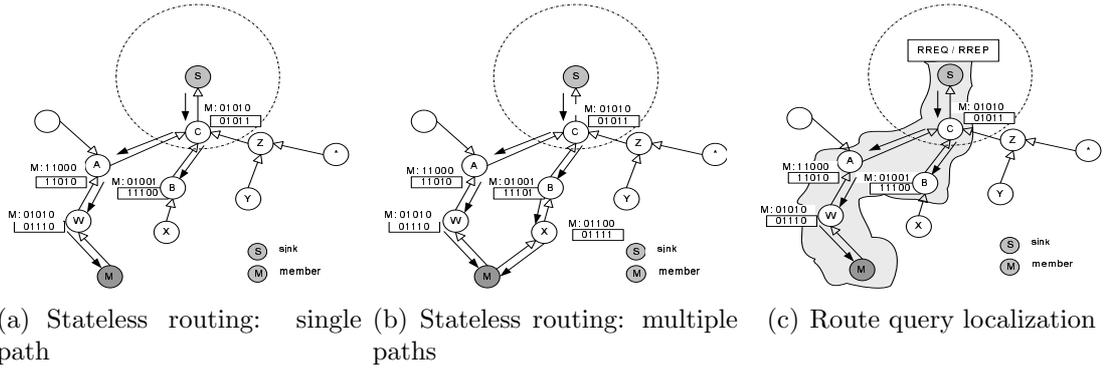


Figure 3.2: FBF can be directly coupled with stateless routing or used as a low-level primitive to localize the route-discovery query. The hashed footprint for a packet en route is shown without a box, whereas the Bloom filter at each node is shown with a box. A membership test of “M” against networked Bloom filters is performed when sink “S” sends data to it. Two independent hash functions and a Bloom filter of length 5 bits are associated with each node. Each node selects its hash functions independently of others.

concerning large sets of IP addresses between measurement points.

3.3 Membership-based Broadcast

3.3.1 Overview

We build membership-based broadcast on top of a novel packet forwarding mechanism called *Footprint-Based Forwarding* (FBF). Representative applications of FBF are illustrated in Figure 3.2. It uses the prior forwarding history cached in a Bloom filter (or bitmap). This forwarding history is called a *footprint* of the node that sent packets. The source addresses of gradient-descending packets en route to the sink are hashed into the filter.

The footprint of a packet is a bit vector of the same length as the bitmap, generated after applying independent hash functions to the packet’s origin address. As a result of this hashing, some positions in the bit vector are set to 1 and others to 0. Several bit-wise operators are used to generate and access the generated footprint in

the bitmap.

In the example of Figure 3.2(a), a packet originating from node M leaves its footprints on the nodes along the trail/path $M \rightarrow W \rightarrow A \rightarrow C \rightarrow S$. Suppose, upon reception of a signal from M, sink S responds to M by returning some information to it. Then, S just broadcasts a packet with any feedback to the signal and M as its recipient.

When the packet is received by S's neighbors, each neighbor performs a membership test of the packet's recipient against its current bitmap. For example, node C will first hash the packet's recipient M and get "01010." The hashed bit string is then compared with its current bitmap, a boxed "01011." If any recipient turns out to be in the bitmap, the received packet is re-broadcast; else it is dropped. This is the way the membership-based broadcast works.

This membership-based broadcast is fundamentally different from other types of broadcast such as probability-based broadcast [26, 36, 47] and location-based broadcast [3, 24, 73] as follows. First, the membership-based broadcast uses the packet-forwarding histories. Second, the membership-based broadcast is deterministic: an intermediate node re-broadcasts only if the membership test is passed. Third, the membership-based broadcast does not require any location information.

There may be multiple paths to a given recipient, as shown in Figure 3.2(b). Note that the false positives associated with a Bloom filter may produce an incorrect path, along which gradient-descending packets are never relayed to the sink. Unfortunately, FBF has no way of figuring this out. A packet under FBF is just re-broadcast as long as it passes the membership test.

Figure 3.2(c) illustrates how FBF can be coupled with on-demand routing protocols to reduce the overhead associated with route discovery. DSR or AODV over FBF could be another solution to the problem, i.e., a low-overhead stateful gradient-

ascending routing protocol. We will not explore this case any further, but it is not difficult to see how FBF will improve the route-discovery overhead.

3.3.2 Data Structure and Hash Functions

Each node has its own Bloom filter which is used to manage its footprint. This is a way of representing a set $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$ of n elements (nodes' addresses) to support membership queries. It uses a bit vector \mathcal{B} of length m with k independent hash functions $\mathcal{H} = \{h_1, h_2, \dots, h_k\}$. For each element $a \in \mathcal{A}$, each hash function $h \in \mathcal{H}$ maps a to an integer in $\{0, 1, \dots, m - 1\}$. Thus, for each element $a \in \mathcal{A}$, the bit positions $\{h_1(a), h_2(a), \dots, h_k(a)\}$ in \mathcal{B} are set to 1. Conversely, given an element $a \in \mathcal{A}$, a membership test of a against \mathcal{B} is a simple procedure that checks the bit positions $\{h_1(a), h_2(a), \dots, h_k(a)\}$ in \mathcal{B} ; if all the relevant bits are set to 1, a is a member; otherwise, it is not.

We use a *counting Bloom filter* [20, 52], an extension to the basic Bloom filter, such that a counter per bit is defined and used to keep track of the number of times the bit is set to 1. For a c -bit counter, the counter ranges from 0 and $C_{max} = 2^c - 1$. The original Bloom filter without any counter is here considered as a special counter Bloom filter with a 1-bit counter.

In a counter Bloom filter, when a counter changes from 0 to 1, the corresponding bit is set to 1. Similarly, when a counter changes from 1 to 0, the corresponding bit is set to 0. For a non-zero counter, the corresponding bit is set to 1. At any time a counter c is non-negative and no larger than C_{max} , regardless of the increment or decrement in c .

A hash function $h \in \mathcal{H}$ is defined as a parameterized function as follows:

$$h(a) \triangleq h(a|p, q, r, m) = ((p \times a + q) \times r) \bmod m$$

where p, q , and r are prime numbers and $r > m$. To make an independent hash function, prime numbers are to be selected independently of other hash functions and other nodes.

3.3.3 Footprint and Operators

Formally, the footprint $\mathcal{F}(a)$ of a node whose address is a , is a vector of m bits with each of the bit positions $\{h_1(a), h_2(a), \dots, h_k(a)\}$ set to 1 and the other bit positions set to 0.

For simplicity, a counting Bloom filter is denoted by \mathcal{B}_c , which consists of a bitmap (or an original Bloom filter) \mathcal{B} and its c -bit counter vector \mathcal{C} . By default, $\mathcal{B} = \mathcal{B}_1$. Using common bitwise-AND (&) and bitwise-OR (|) operators, we define operators to manipulate the footprint of a node over a counting Bloom filter \mathcal{B}_c as follows:

Stamping footprints $\mathbf{S}(\mathcal{B}_c, a)$:

$$\begin{aligned} \mathcal{C}[i] &\leftarrow \mathcal{C}[i] + 1, \quad i \in \{h_1(a), h_2(a), \dots, h_k(a)\} \\ \mathcal{B}[j] &\leftarrow 1, \quad j \in \{i : \mathcal{C}[i] > 0\}. \end{aligned}$$

Erasing footprints $\mathbf{E}(\mathcal{B}_c, a)$:

$$\begin{aligned} \mathcal{C}[i] &\leftarrow \mathcal{C}[i] - 1, \quad i \in \{h_1(a), h_2(a), \dots, h_k(a)\} \\ \mathcal{B}[j] &\leftarrow 0, \quad j \in \{i : \mathcal{C}[i] = 0\}. \end{aligned}$$

Initializing footprints $\mathbf{I}(\mathcal{B}_c)$:

$$\mathcal{C} \leftarrow 0, \mathcal{B} \leftarrow 0.$$

Losing footprints $\mathbf{L}(\mathcal{B}_c)$:

$$\begin{aligned} \mathcal{C}[i] &\leftarrow \mathcal{C}[i] - 1, \quad i \in \{\text{randomly chosen } k \text{ positions}\} \\ \mathcal{B}[j] &\leftarrow 0, \quad j \in \{i : \mathcal{C}[i] = 0\}. \end{aligned}$$

Testing footprints $\mathbf{T}(\mathcal{B}, a)$:

$$\prod_{i=1}^k \mathcal{B}[h_i(a)]$$

where $\mathcal{B}[h_i(a)]$ is the bit at $h_i(a)$ position of \mathcal{B} .

3.3.4 Clocking Footprints $\mathbf{C}(\mathcal{B}_c)$

By definition, counters associated with a footprint indicate the number of times the bits are set to 1. Such counters could be translated to time ticks, representing the elapsed time since the last time the bits were set. In general, the interpretation of counters is application-specific. In the next section, we will describe how GASR makes use of these counters.

3.3.5 Footprint Loss Model

All of the footprints can be erased through the \mathbf{E} operator, or a random footprint can be lost through the \mathbf{L} operator. Of course, it is possible for footprints to remain forever once stamped in a bitmap, i.e., no loss model. Obviously, using a loss model introduces false negatives due to the pigeon hole principle; a bit in the filter represents the membership relationship with several other nodes, not just with a specific node. Usually, the choice of a proper footprint-loss model and its periodic or occasional enforcement are up to the applications that use FBF.

3.4 Stateless One-to-Many Gradient-Ascending Routing

This section describes how the membership-based broadcast is used in GASR. We will also describe how to build the gradient-field around the sink to efficiently suppress redundant broadcasts.

3.4.1 Per-Sink Bloom Filters

A gradient-field is created, destroyed, and maintained by an individual sink independently of other sinks, if any. It is, therefore, natural and convenient to set up a Bloom filter for each sink at a node such that the filter's lifetime is the same as the gradient-field's lifetime. A node manages and uses multiple independent Bloom filters, one for each sink, and records the history of forwarding packets in the filters corresponding to their sinks.

Instead of using a Bloom filter, one might consider using a bitmap of size equal to, or greater than the number of nodes in the network, so every node in the network is assigned statically a particular bit position in the bitmap. This approach has several drawbacks. First, it reduces flexibility since a new bitmap needs to be installed on every node whenever a network grows beyond a pre-allocated bitmap size, or a membership change occurs in the existing node set. Second, it needs a bigger bitmap size than the number of nodes available in the network since the node identifiers are not contiguous. For example, the length of an IPv4 address is 32 bits, and hence, a total of 2^{32} addresses are available. But, only a fraction of them are likely to be seen in the network. Nevertheless, it needs to allocate a large bitmap of size up to 2^{32} unless an additional table is used to map the node identifier to a bit position. Third, false positives cannot be exploited in a single large bitmap since it does not incur any false positive. Note that false positives can be used to implement reliable broadcast by compensating lost broadcasts when a wireless channel is lossy.

3.4.2 Gradient-Field Construction

Recall that GASR requires gradient-descending routing, but does not depend on how gradient-descending routing works. Regardless of how the gradient-field is

constructed, GASR expects each node to have sort of “height” with respect to the sink. For completeness, we assume that a gradient-field is constructed as follows (see Section 2.3.2 for details).

Each node is allowed to maintain its ‘height’ relative to the sink, where the height is the sum of the lengths of links on the minimum-cost path to the sink. Without using nodes’ geographic locations, the length of a link is estimated by periodically measuring the quality of the link (between two neighbors).

The height of node i relative to sink s in the gradient-field, denoted by $H_s(i)$, may, in its simplest form, be the hop count of the minimum-cost path from i to the sink s , but the method in Section 2.3.2 yields a finer-grained scalar value than just a hop count.

3.4.3 Footprints Accumulation and Loss Model

Accumulating footprints involves the invocation of the \mathbf{S} operator on a relevant Bloom filter. However, the accumulation of footprints without erasing them over the lifetime of a Bloom filter makes it useless since too many bits in the filter would have been set. Thus, it may need to reset the filter periodically.

When nodes are introduced for the first time in the network, they have to leave their footprints at the sink. Otherwise, the sink cannot send any packet to them. This is not a problem in self-configuring wireless sensor or mesh networks, where every node is required first to find a path to the sink. The deletion of a node may result in the failure of an existing path to the sink. An alternative path will then be explored by the self-configuration process and new footprints will be accumulated along that path.

GASR implements a filter with a random loss model, as well as the one with a

```

1 tick ← (tick+1) mod period;
2 if tick == 1 then
3   L( $\mathcal{B}_c$ );
4   S( $\mathcal{B}_c, a$ );

```

Algorithm 5: Modified stamping operator $S_r(\mathcal{B}_c, a)$ in the random loss model

Input: Received broadcast p from $prev$

```

1 assert p.src == sink and find  $\mathcal{B}_c$  for sink;
2 reicast.suppress(p.src, p.seq, p.prev);
3 if  $p$  is not received before and  $T(\mathcal{B}_c, p.dst) == 1$  then
4   delay =  $D(this, prev, sink) + C(\mathcal{B}_c, p.dst) + 0.11 \times U(0, 1)$ ;
5   broadcast  $p$  after delay;
6   reicast.enqueue(p);
7 else drop p;

```

Algorithm 6: Forwarding gradient-ascending packets

conventional no-loss model. The random loss model requires to modify the stamping operation, as described in Algorithm 5. The `period` parameter in the algorithm determines how often the loss operation can be applied. Currently, the period is arbitrarily set to 10. In Section 3.5, we will study the impact of this footprint loss model on the routing performance.

3.4.4 Forwarding Gradient-Ascending Packets

Given a gradient-field as described above, GASR forwards gradient-ascending packets from the sink as in Algorithm 6. It involves interactions with FBF and MAC upon reception of a packet from one of its neighbors, `prev`.

Line 1 checks if a packet originates from the right sink for which a Bloom filter \mathcal{B}_c is created. This step is necessary since GASR assumes a per-sink Bloom filter. If a Bloom filter matching `sink` is found, the packet forwarding procedure is simple, as shown in line 3.

In line 3, a duplicate broadcast identification method is used to eliminate redun-

dant broadcasts, which is a trivial operation in broadcast-based dissemination. After passing the membership test, the packet is broadcast again with some delay not only to avoid MAC-level broadcast collisions, but also to eliminate redundant broadcasts.

The delay incorporates two important ideas. $D(\mathbf{t}, \mathbf{p}, \mathbf{s})$ is a delay to favor those nodes making farther progress, whereas $\mathbf{C}(\mathcal{B}_c, \mathbf{p}, \mathbf{dst})$ is a delay to favor those nodes being used frequently. The remaining term in the delay calculation is a tie-breaking random delay.

$D(\mathbf{t}, \mathbf{p}, \mathbf{s})$ is defined as follows:

$$D(\mathbf{t}, \mathbf{p}, \mathbf{s}) = e^{(H_s(p) - H_s(t))} \quad (3.1)$$

where $H_s(i)$ represents the height of node i with respect to sink s in the gradient-field. $D(\mathbf{t}, \mathbf{p}, \mathbf{s})$ ensures that those nodes farther from the sink are given a shorter delay.

On the other hand, $\mathbf{C}(\mathcal{B}_c)$ has the following definition, which returns the age normalized with C_{max} :

$$(1 - \bar{C}/C_{max}), \text{ where } \bar{C} = \frac{\sum_{i=1}^k \mathcal{C}[h_i(a)]}{k} \quad (3.2)$$

where $\mathcal{C}[h_i(a)]$ is the current counter associated with $h_i(a)$ bit position of \mathcal{B}_c . Thus, $\mathbf{C}(\mathcal{B}_c)$ becomes close to 0 for a frequently-used node. Note that once a counter reaches C_{max} , it remains there no matter how long ago this limit was reached.

3.4.5 Reliable Membership-based Broadcast

Any routing protocol must provide high end-to-end packet delivery performance, which depends largely on retransmission of in-transit packets in case of failure. Unlike unicast-based delivery that can send a packet multiple times to a single next hop, any broadcast-based packet delivery transmits the packet only once and relies on multiple next-hop nodes for its successful advance toward the destinations.

Unlike the usual broadcast-based delivery, GASR is likely to have only a small fraction of multiple next-hop nodes eligible for relaying packets due to the membership test. We cannot expect to realize the advantage of having multiple next-hops. To get around this, GASR employs a *unicast-like retransmission* scheme. With the unicast-like retransmission, a node re-broadcasts the received packet several times until either it overhears any of the next-hop nodes re-broadcasting the packet (i.e., passive ACK), or it reaches the maximum retry limit.

However, there is a subtle difference between passive ACKs of GASR and a typical unicast. In the latter, a node knows which node's transmission to overhear since it knows where a packet is being forwarded to. But, in the former no forwarding node is specified: any neighboring node that passed the test for the membership of the received packet will re-broadcast it. Thus, we need a rule for the passive ACK in GASR. Specifically, a packet p sent by a node f is said to be *passively acknowledged* by a packet q transmitted by a node n if (i) $p.\text{seq} = q.\text{seq}$ and $p.\text{src} = q.\text{src}$, and (ii) $f.\text{height} \leq n.\text{height}$. Both seq and src in the clause (i) uniquely identify a packet, as used in a typical duplicate packet identification method. The $f.\text{height}$ in the clause (ii) is $H_{src}(f)$ given the sink src . Reducing the number of broadcasts in this way is called a *gradient-based suppression*.

Improving the packet-delivery performance via retransmissions requires several modifications, some of which have already been highlighted in Algorithm 6. First, each node needs to check if there is any outstanding packet (passively un-ACKed). Line 2 in Algorithm 6 shows this; **suppress** looks for any packet that can be passively acknowledged by the received packet and suppresses any further transmission of packets that are passively acknowledged. The field $p.\text{prev}$ in line 2 represents the node that has just broadcast the received packet.

Second, a node needs to save the received packet for later retransmissions even after broadcasting it. Line 6 in Algorithm 6 shows this; the received packet p is enqueued into the buffer `rebcast`. Third, a periodic timer is needed to check if there is any passively-unacknowledged packet in the buffer and if so, triggers retransmissions within the maximum retry limit. Finally, a destination is supposed to perform 1-hop broadcast once whenever it first receives the packet, as a passive ACK.

3.5 Evaluation

This section evaluates the characteristics of membership-based broadcast and GASR with a regular grid topology under realistic simulation settings. The topology is an 11×11 grid covering a $1\text{km} \times 1\text{km}$ square field. A node at each grid point (i, j) is identified as $(i + 11 \times j)$, $0 \leq i, j \leq 10$. The two-ray ground model used in an earlier version of this work [33] is replaced with the shadowing propagation model, which was recommended in [42] as a more realistic propagation model to reflect wireless link characteristics such as the link asymmetry and the time-varying link-quality. The *ns-2* shadowing model and other configurations are set up as in Table 3.1. These settings are derived from our implementation of Proxim Orinoco 11b PCI adapter [58] for a semi-open environment.

The simulation consists of two parts. The first part involves the building of a gradient-field and the accumulation of footprints. In parallel with the first part, the second part is the actual evaluation of GASR. The first part begins with the sink sending ADV once in the beginning. After that, every node sends HELLO periodically with the proposed link-quality probe piggybacked. Every node except for the sink becomes a data source and is continually sending an average MAC-level payload of 1500 bytes, or $1500 + U(-750, 750)$ bytes, to the sink. But, it takes turns in sending

Table 3.1: Configurations for 802.11b with the shadowing model.

Path loss exponent		4	
Shadowing deviation		4	
Reference distance		10	
Frequency	2.4 GHz	Transmit power	15 dBm
CPTresh	10	CSTresh	-105 dBm
RXThresh (11 Mbps) [†]		-79.84 dBm, (95% at 50m)	
RXThresh (5.5 Mbps) [†]		-85.68 dBm, (95% at 70m)	
RXThresh (2 Mbps) [†]		-90.05 dBm, (95% at 90m)	
RXThresh (1 Mbps) [†]		-94.31 dBm, (95% at 115m)	
Data Rate		11 Mbps (w/o auto rate)	
Basic Rate		2 Mbps	
MAC headers		(24,14,14,28) bytes @ Basic Rate	
PLCP length		192 bits @ 1 Mbps	
SIFS	10us	DIFS	50us

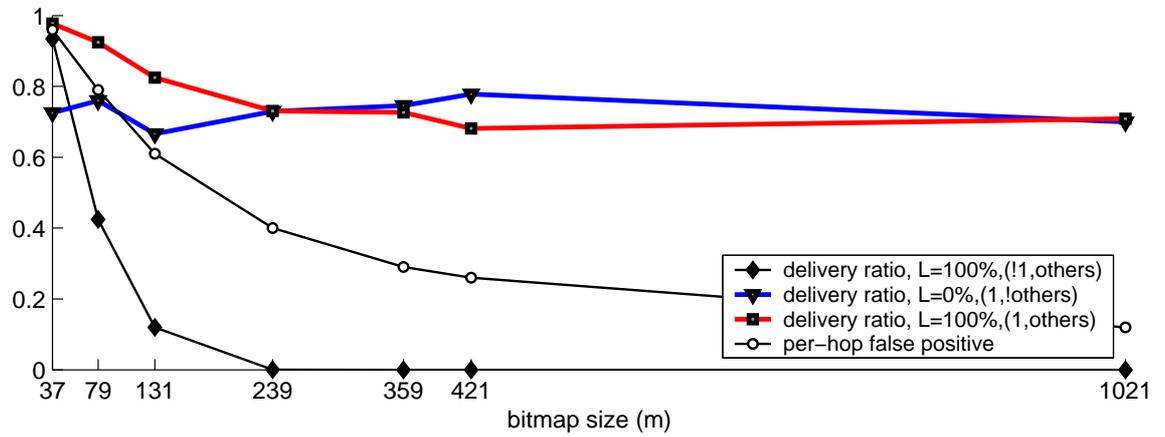
[†] These thresholds are configured for the path-loss exponent 4 and the shadowing deviation 4, and remain unchanged regardless of changes in both parameters.

data to the sink every 4 seconds. In the second part, while running the first part, the sink broadcasts gradient-ascending packets of the same payload size as above every 4 seconds, with node 1 as the recipient. The entire simulation lasts 1511 seconds. Results are averaged over 10 simulation runs, each with a different random seed.

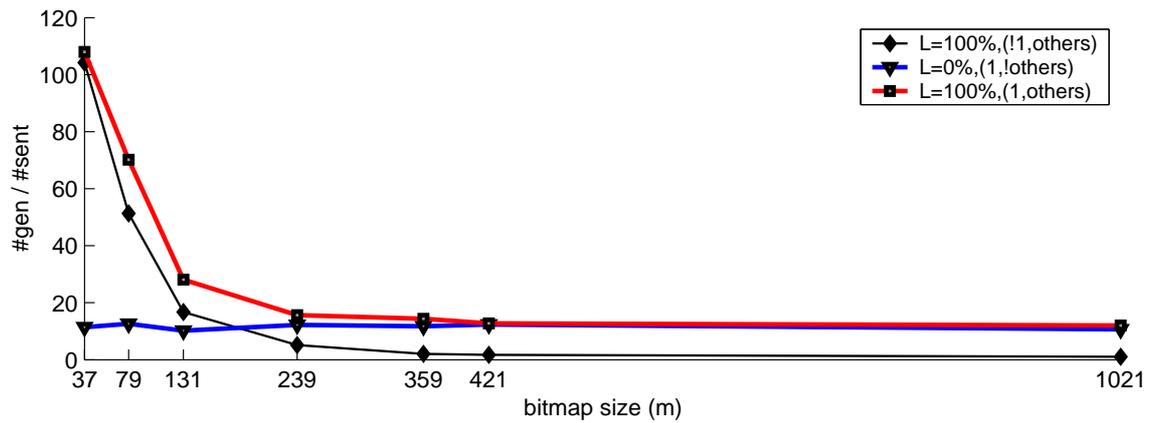
3.5.1 Primitive Membership-based Broadcast

The major objective of this evaluation is to understand the fundamental aspects of membership-based broadcast. To this end, we ran membership-based broadcast without GASR-specific features such as gradient-based suppression or unicast-like re-transmissions. Therefore, those received packets from neighbors will be re-broadcast if they are not duplicate and pass the membership test.

We control the number of false positives as follows. First, a filter is preloaded with a record of randomly-chosen nodes. The parameter L in Figure 3.3 indicates the fraction of nodes in the network preloaded with a bitmap. For example, $L = 100\%$



(a) Delivery ratio and per-hop false positives



(b) Delivery overhead: #gen / #sent

Figure 3.3: Routing performances under the grid topology with $n = 121, k = 1$. L represents the percentage of nodes that are pre-hashed to the filter. (!1, others) means that all but node 1 are sending data to the default sink located at (5,5).

for a network of 121 nodes means that 121 nodes are selected randomly from the network and their identifiers are hashed into the filter. On the other hand, the second method is to have many concurrent flows in the network, thus making more bits likely to be set.

Our first experiment is to allow all but node 1 to send data to the sink, whereas the sink starts membership-based broadcast with node 1 as the recipient. The label “L=100%,(!1,others)” in Figure 3.3(a) represents this. This setup implies that every membership-based broadcast arriving at node 1 is a false positive: node 1 is excluded in the pre-loading step of this experiment. As can be seen in Figure 3.3(a), the delivery ratio quickly decreases as the bitmap gets larger. For example, when the bitmap size is almost as large as the number of nodes in the network, i.e., $m = 131$, the achieved delivery ratio gets below 0.2. By contrast, the per-hop false positives under the same bitmap size are about 60%. This is exactly what the membership-based broadcast tries to achieve by using networked filters: false positives die down quickly during hop-by-hop movements.

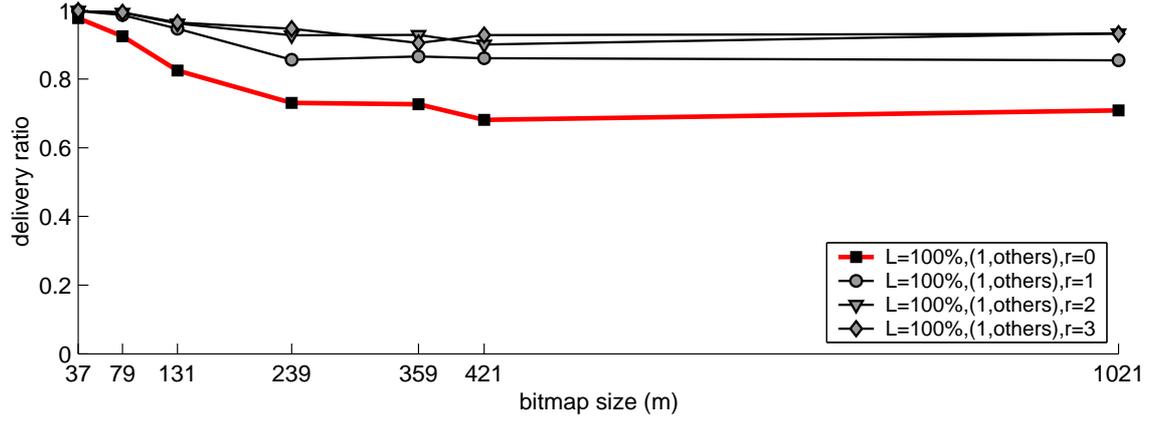
Our second experiment is to remove false positives by allowing only node 1 to send data to the sink — only node 1 is allowed to leave its footprint in the network. In addition, the pre-loading step is skipped. The label “L=0%,(1,!others)” in Figure 3.3(a) represents this. This setup implies that every membership-based broadcast results from the true membership. As can be seen in this figure, the delivery ratio remains about 0.7 regardless of the bitmap size. This is because the true membership is not affected by the size of the bitmap. The reason for the delivery ratio falling short of 100% is the lossy broadcast in wireless networks.

Our third experiment is to create a natural mix of true membership and false positives by allowing all nodes to send data to the sink. The label “L=100%,(1,others)”

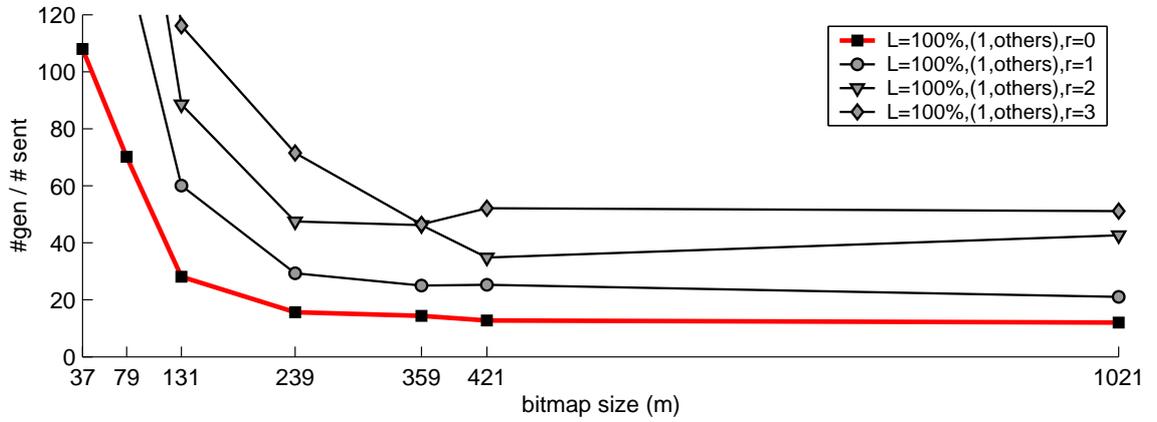
Figure 3.3(a) represents this. This experiment will improve the delivery ratio since many redundant broadcasts will be produced. The result in the figure shows that the delivery ratio benefits much from the increased false positives, but such benefits diminish gradually as the bitmap size gets larger since the increased bitmap size starts reducing the false positives.

The delivery overheads under each experiment are plotted in Figure 3.3(b). The delivery overhead is defined as the ratio of the number of broadcasts generated in the network to the number of broadcasts sent by the sink. One interesting aspect of these results is that redundant broadcasts do not necessarily contribute anything to the increased delivery ratio. For example, the delivery overhead under “L=100%,(!1,others)” is much higher than that under “L=0%,(1,!others)” when the bitmap size is smaller than 131. Despite its high number of broadcasts, the latter achieves a much lower delivery ratio, as seen in Figure 3.3(a).

The results from the above three experiments imply that we have to distinguish two types of cause for the degradation of delivery ratio: *lost broadcast* vs. *collided broadcast*. Lost broadcast refers to the loss of membership-based broadcast over a lossy wireless channel, whereas collided broadcast refers to the collision of membership-based broadcast with other simultaneous transmissions. Lost broadcast is usually caused by a poor wireless channel condition. According to our evaluation results, lost broadcast appears to be a major cause for the degradation of delivery ratio in Figure 3.3(a). This conjecture is based on the result that the delivery ratio for a high traffic volume, i.e., “L=100%,(1,others),” doesn’t make a meaningful difference from that for a low traffic volume, i.e., “L=0%,(1,!others),” when the bitmap size is larger than 239.



(a) Delivery ratio



(b) Delivery overhead: #gen / #sent

Figure 3.4: Improved routing performances under membership-based broadcast with different retry limits $r = 0, 1, 2, 3$.

3.5.2 Impact of Unicast-like Retransmission

The conjecture in previous experiments led us to an investigation of the impact of unicast-like retransmissions. Figure 3.4(a) shows that the proposed retransmission achieves a high delivery ratio in the presence of lost broadcast over lossy wireless channels. However, the improved delivery ratios come at the expense of increased overheads. Figure 3.4(b) shows the delivery overhead. As expected, sending packets multiple times incurs high overhead. Depending on the bitmap size used, the unsuppressed membership-based broadcast incurs by far more overhead than simple

flooding, the overhead of which is 121 since every node broadcasts the received packet once.

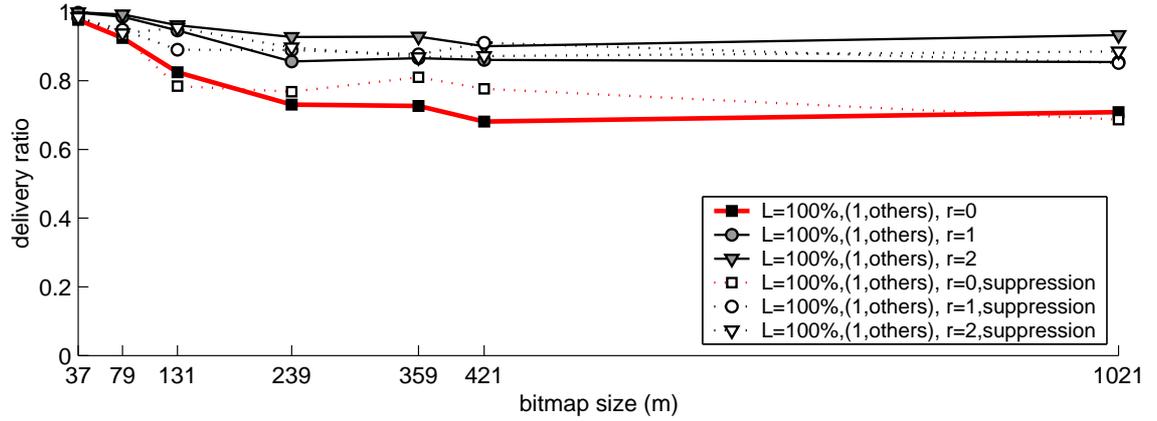
3.5.3 Impact of Gradient-Based Suppression

Figure 3.5 demonstrates that the proposed suppression reduces the delivery overhead significantly while maintaining the delivery performance comparable to that without suppression.

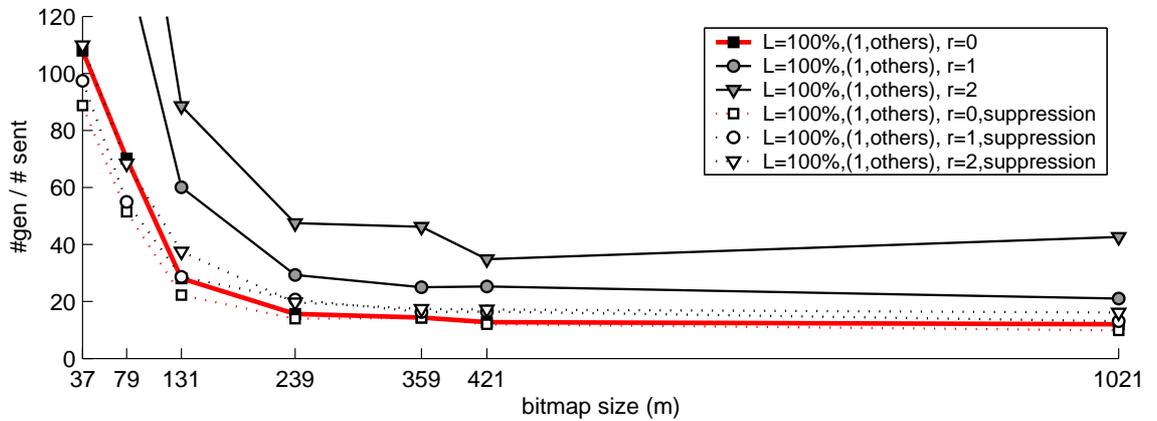
As can be seen in Figure 3.5(a), the delivery ratios with the suppression tend to decrease slightly. However, this observation does not always hold. For example, some of the delivery ratios, especially when $r = 0, m = 239, 359, 421$, are improving despite the suppression. As for the delivery overhead, Figures 3.5(b) and (c) show that about 50% less overhead is incurred when $r = 1$, and that more savings are possible when $r = 2$.

As compared to simple flooding, the results in Figure 3.5 suggest that our membership-based broadcast reduces the delivery overhead by a factor of 5 with comparable delivery ratios, especially when $m \geq 239$. Here, simple flooding is assumed to attain a 100% delivery ratio with every node broadcasting the received packets once.

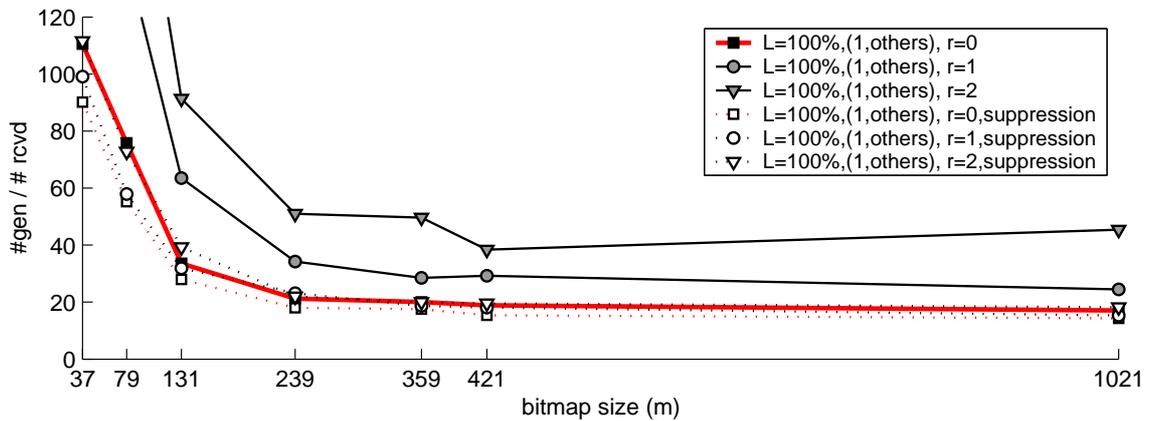
One may be curious about how the change of $H_s(i)$ impacts the routing performance. Figure 3.6 shows the delivery ratio and the delivery overhead when $H_s(i)$ is the hop count from node i to sink s . When the hop count is used, the number of broadcasts seems to decrease (Figure 3.6(b)). However, the delivery ratio also decreases (Figure 3.6(a)). Retransmitting broadcasts with $r = 1$ does not change this conclusion. The delivery overhead when $r = 1$ is very similar to that when $r = 0$, and thus, not plotted in Figure 3.6(b).



(a) Delivery ratio

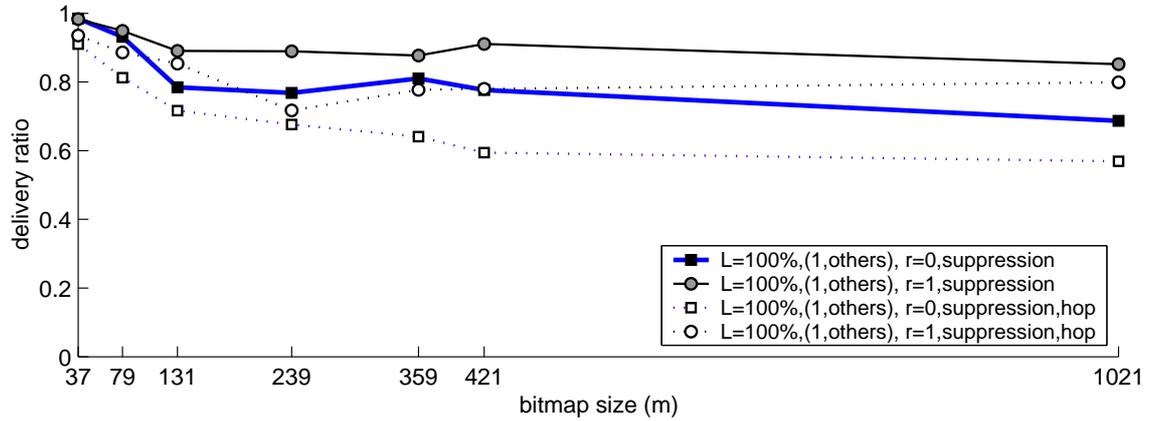


(b) Delivery overhead: #gen / #sent

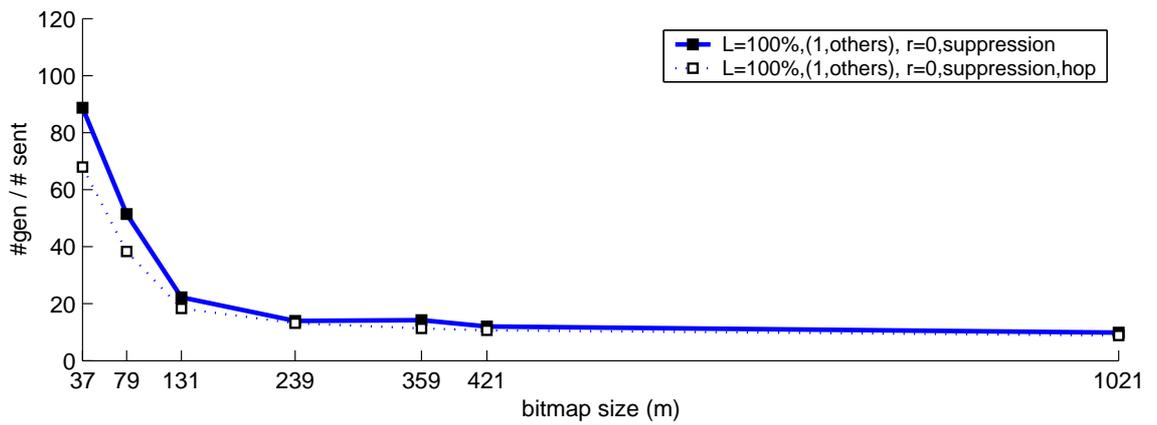


(c) Performance comparisons: #gen / #rcvd

Figure 3.5: Impact of suppression on routing performances when $n = 121, k = 1$.

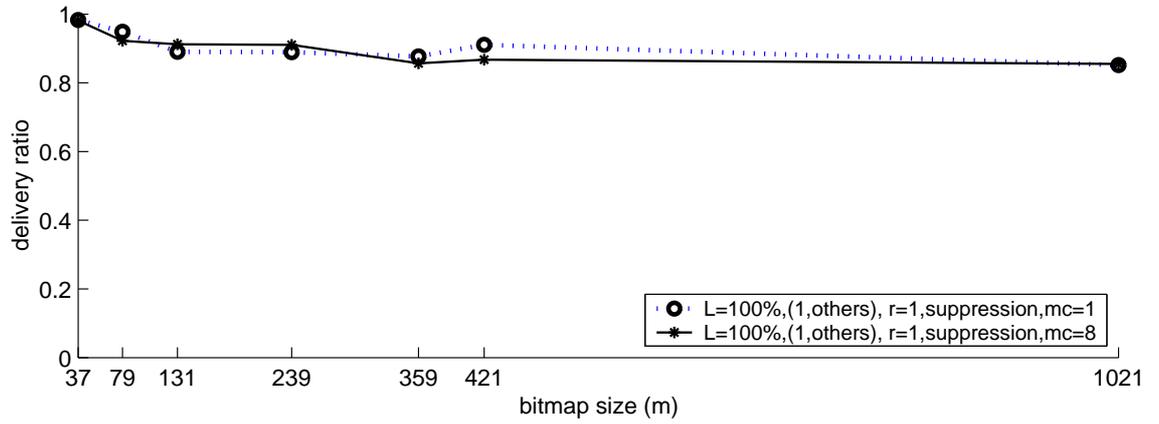


(a) Delivery ratio

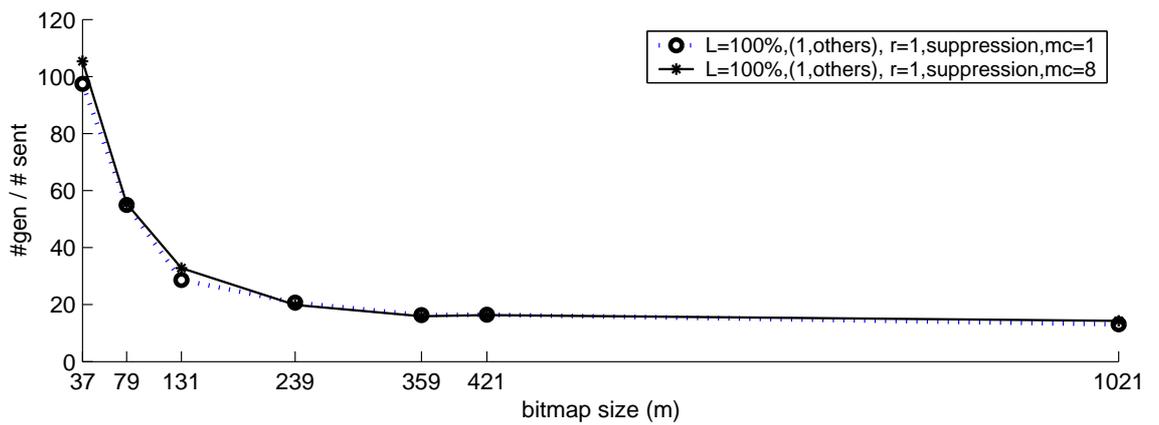


(b) Delivery overhead: #gen / #sent

Figure 3.6: Impact of the suppression on routing performances for $n = 121, k = 1$. $H_s(i)$ is hop count from node i to sink s , which is indicated by “hop”.



(a) Delivery ratio



(b) Delivery overhead: #gen / #sent

Figure 3.7: Impact of the maximum counter (mc) on the routing performances under $n = 121, k = 1$.

3.5.4 Impact of Maximum Counter

The objective of this experiment is to determine if increasing the maximum counter (mc) is advantageous to the delivery ratio or the delivery overhead. So far, we have used the default counting Bloom filter whose maximum counter is 1. However, a counter greater than 1 can be used to tell which bit position is more frequently used. According to (3.2), such a large counter value can differentiate the suppression delay better.

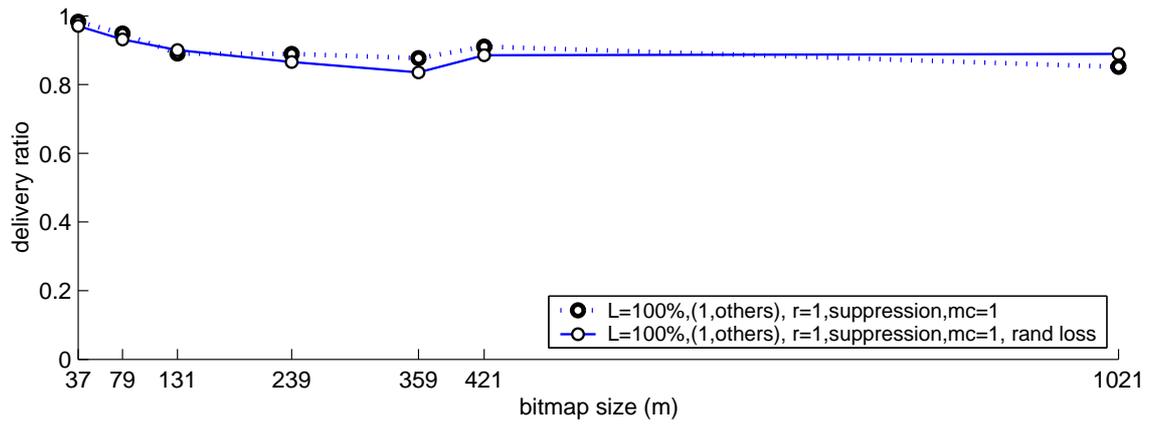
Figure 3.7 gives some answer to this question: it is difficult to say that a large counter substantially improves the delivery ratio or the delivery overhead. Similar observations are also made when $r = 0, 2, 3$, but are not plotted here.

3.5.5 Impact of Loss Model

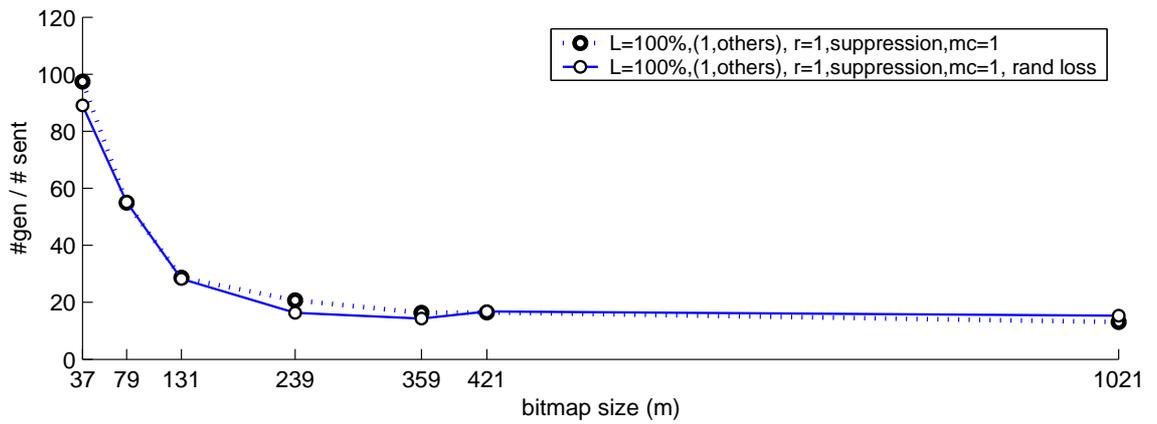
Figure 3.8 presents the results regarding the impact of the random loss model on the routing performances. As can be seen in the figure, both the delivery ratio and the delivery overhead under the random loss model are almost the same as those under the no-loss model. Another comparison with $mc = 8$ is not presented here since it shows almost the same trend as above.

3.6 Conclusion

In this chapter we addressed the problem of designing gradient-ascending routing given unidirectional gradient-descending routing such as a spanning tree rooted at the sink in typical many-to-one communications. Gradient-ascending routing is required not to hinge on any on-demand route discovery nor on simple flooding. The proposed gradient-ascending routing, called GASR, relies only on the packet forwarding histories accumulated by the underlying gradient-descending routing. Due



(a) Delivery ratio



(b) Delivery overhead: #gen / #sent

Figure 3.8: Impact of the random loss model on the routing performances under $n = 121, k = 1$.

to its stateless property, GASR does not create any protocol conflicts with the underlying gradient-descending routing. As compared to simple flooding, it significantly reduces the number of broadcasts while achieving a high delivery ratio, thanks to its unicast-like retransmission and gradient-based suppression. GASR can perform any unidirectional gradient-descending routing with a slight storage overhead to manipulate the forwarding histories via a Bloom filter.

CHAPTER IV

ONE-TO-ONE GRADIENT-HEIGHT VECTOR ROUTING

4.1 Introduction

Although communications from many nodes to a wired Internet gateway represent a typical interaction pattern for wireless mesh networks (WMNs), other types of communication scenarios are also possible. For example, users in a community-based WMN may want to access data or video files owned and exported by other users, or exchange shared secrets with each other, thereby requiring efficient one-to-one routing.

This chapter explores how to design such a one-to-one routing protocol for WMNs. One approach to this problem is to use conventional proactive distance-vector routing [54] or link-state routing [30, 31, 61], or reactive on-demand routing [55]. However, one-to-one routing in WMNs tends to be occasional. Thus, keeping a proactive routing protocol running for all destinations is not cost-effective. Likewise, running a reactive routing protocol becomes expensive and suffers the scalability problem as the number of one-to-one communication sessions increases.

We present a novel routing protocol called *Gradient-Height Vector Routing* (GHVR) for efficient one-to-one communications in WMNs. It does not flood route-discovery

requests nor does it maintain route information between nodes. It only requires information on routes to a set of pre-selected gateways. We eliminate the route request step by running geographic routing [9, 37, 46] over *virtual coordinates* extracted from the routes to the set of pre-selected gateways; we refer to the set of pre-selected gateways used to construct virtual coordinates as *landmarks*. Running geographic routing with the virtual coordinates is then realized by making the routing protocol treat the virtual coordinates as if they were real, geographic locations.

GHVR is a virtual coordinate routing protocol designed to achieve the flexibility, scalability, and simplicity of geographic routing even when the true geographic locations of nodes are not available. Other routing protocols [10, 21, 23] are similar to GHVR in the spirit: they are all landmark-based (or gradient-based) coordinate routing with coordinates represented as a set of scalar values such as hop counts or distances to the landmarks. The type of scalar value indicates if coordinates are *discrete* or *continuous*. For example, the coordinates defined in terms of hop count are discrete, whereas the coordinates using distance are continuous. GHVR differs from other protocols [10, 21, 23] in that GHVR is built on continuous coordinates while the others on discrete coordinates.

Continuous coordinates are finer-grained than the discrete coordinates, and make a significant improvement of routing performance over the discrete counterpart. Defining continuous landmark-based coordinates in networked systems is not new. For example, several techniques to produce fine-grained and accurate landmark-based coordinates built on a routing protocol have already been proposed in [13, 48].

However, a clear distinction should be made between coordinates *built on* a routing protocol and coordinates *for building* a routing protocol itself. Coordinates built on a routing protocol are mainly used to infer the end-to-end delay between two

arbitrarily-chosen nodes before they communicate with each other. A global virtual coordinate is built from actual round-trip time measurements with the selected landmarks; these landmarks also measure round-trip times among themselves. Measuring the round-trip time requires routing to be implemented first between landmarks, and between landmarks and nodes. It is also assumed that the same routing algorithm is used between any pair of nodes; otherwise, any inference from measurements becomes meaningless. Unfortunately, all of existing continuous coordinates [13, 48] fall into this category.

Coordinates for building a routing protocol are used to route packets between a pair of nodes when they need to communicate with each other. A global virtual coordinate is built using scalar values, typically hop count [10, 21, 23], from nodes to each landmark that periodically broadcasts beacons. How to route packets between nodes has not yet been decided and therefore, needs to be designed differently from the broadcast. If there are d landmarks in the network, the “location” of each node will be represented by a d -dimensional vector. Routing packets from one node to another is then becomes a simple matter of running the standard geographic routing with the thus-obtained global virtual coordinate.

Because of differences in both the underlying assumptions and the intended application context, the same coordinate determination techniques developed on an existing routing protocol cannot be used for building a routing protocol itself. This chapter elaborates on the latter. Even if we limit ourselves to landmark-based coordinates for routing, there are several design factors that affect routing performance, including the coordinate type and the definition of distance function over the virtual coordinate space.

This chapter makes the following main contributions:

- It presents a one-to-one routing protocol, **GHVR**, that routes packets with landmark-based virtual coordinates. The proposed landmark-based virtual coordinates are based on the estimation of each link’s length on the minimum-cost paths to the landmarks, and are finer-grained than the existing discrete coordinates that use hop counts.
- It presents comprehensive and comparative results as to how (i) different types of landmark-based coordinates and (ii) different types of the distance functions over landmark-based coordinates affect routing performance.

The rest of this chapter is organized as follows. Section 4.2 describes the related work. Section 4.3 details **GHVR**. Section 4.4 provides extensive evaluation of the proposed scheme under realistic wireless network models. Section 4.5 concludes the chapter.

4.2 Related Work

Defining a continuous virtual coordinate in networked systems is not new, but has never been tried for designing a routing protocol. For example, Vivaldi [13] is a decentralized network coordinate system that assigns hosts synthetic coordinates (expressed in real numbers) such that the distance between two hosts in the virtual coordinate space accurately predicts the actual communication latency between them. In other words, Vivaldi is designed to predict the latency prior to communication between nodes.

Unfortunately, Vivaldi and most of the work related to Vivaldi require that routing be implemented before coordinates are established. However, **GHVR** and other existing work [10, 21, 23] define gradient-based coordinates first, then use the coordinates to route packets between a pair of nodes. Thus, the assumptions and the design

requirements for the gradient-based coordinates are totally different from those of Vivaldi.

Several routing protocols, such as LCR [10], BVR [23], and GLIDER [21], define synthetic coordinates with respect to a set of pre-selected nodes called *landmarks* [10, 21] or *beacons* [23].¹ They use hop count along the minimal-hop path (e.g., LCR or GLIDER), or the minimal-cost path to the set of selected landmarks (e.g., BVR).

LCR identified delivery-failure problems, such as routing loops or greedy routing failures, associated with the hop-based discrete coordinate. Instead of seeking to improve the granularity of a coordinate, LCR uses a loop-avoidance mechanism and time-to-live (TTL)-based packet dropping. These mechanisms only address how to avoid wasting wireless bandwidths rather than offering a reliable mechanism to deal with the problem associated with the discrete coordinate.

The original geographic routing protocols [8, 9, 37] did not take link-quality into account. Later, two similar geographic link metrics such as packet-reception-rate (PRR) \times DISTANCE [63] and normalized advance (NADV) [46] were proposed to make the original geographic routing aware of link-quality. The greedy forward definition used in GHVR is NADV in the gradient-based coordinate space.

Research over the last few years has focused on understanding and characterizing link-quality in wireless sensor networks [72] or mesh networks [1, 6, 12, 17, 18]. New link or path metrics of wireless link-quality have been proposed, including the expected transmission count (ETX) [12], the expected transmission time (ETT), the weighted cumulative ETT (WCETT)[18], and the effective number of transmissions (ENT) [41].

Even though these metrics have proved to be effective in improving throughput,

¹GLIDER only uses virtual coordinate routing as intra-tile routing in its entire routing framework.

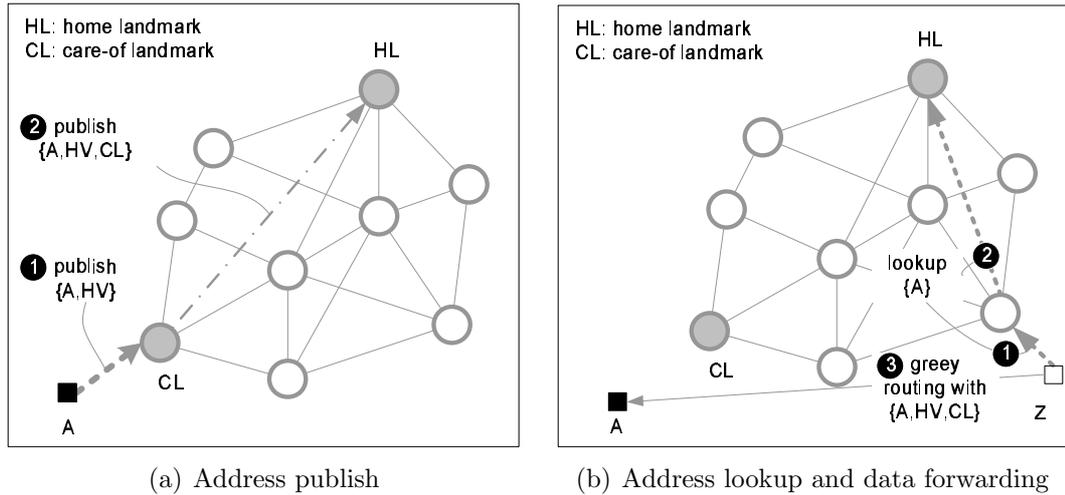


Figure 4.1: Illustration of one-to-one GHVR

their broadcast-based measurements create mismatches between broadcast-based and unicast-based neighbor sets. For example, in 802.11-based wireless networks, broadcast is sent at a default rate of 2Mbps, whereas unicast uses a default rate of 11Mbps. This difference in data rate makes the range of broadcast much greater than that of unicast. The hybrid link-quality measurement scheme in GHVR recognizes such disparity and incorporates it in the calculation of link-quality. More recent work has addressed accurate link measurement, e.g., efficient and accurate link-quality monitor (EAR) [38], which also uses unicast- and broadcast-based link-quality measurements.

4.3 Gradient-Height Vector Routing

4.3.1 Overview

Figure 4.1 illustrates how GHVR works in general. Unlike the Internet routing where an IP address is used not only to identify a node, but also to route packets, the landmark-based routing uses the coordinates only for routing packets. Thus, it needs a location service that maps the node identifier to the landmark-based coordinate.

In its simplest form, a location service can be implemented by using the “home”

and “care-of” landmarks (HL, CL). The home landmark is uniquely determined by hashing a node ID, whereas the care-of landmark is the closest (e.g., in number of hops or estimated distance) landmark to the node. In Figure 4.1(a), node A publishes periodically to CL, its ID and landmark-based coordinates. The landmark-based coordinates of a node are also called the height vector (HV) of that node. Also, CL periodically publishes a list of nodes registered with itself to their respective HLs. For a member node A, the information reported by CL to A’s HL is a triplet $\langle A, HV, CL \rangle$. CL may aggregate the information for the same HL and minimize the report traffic.

When a node Z in Figure 4.1(b) wants to send data to node A without knowing A’s HV, it has to take several steps as follows. First, it issues a query for A’s HV to the nearest landmark. If that nearest landmark (or Z’s CL) has no entry for A’s HV, it relays the received query to A’s HL, which always knows A’s HV. After obtaining A’s HV from either Z’s CL or A’s HL, node Z will send packets to A by using $\langle A, HV, CL \rangle$ as the “address” for routing packets. Note that CL in this routing address is the nearest landmark to A.

Optimizing the location service is beyond the scope of this chapter. In fact, the problem of designing a scalable location service in the landmark-based coordinate space itself is extensive enough to constitute a chapter on its own. Our description here is to show how GHVR works for one-to-one communications.

4.3.2 Type of Coordinates

The landmark-based coordinate used by GHVR is called the *extrapolated* coordinate. The reference model in Figure 4.2 illustrates it as well as other types of the landmark-based coordinates. The absolute coordinate represents true geographic

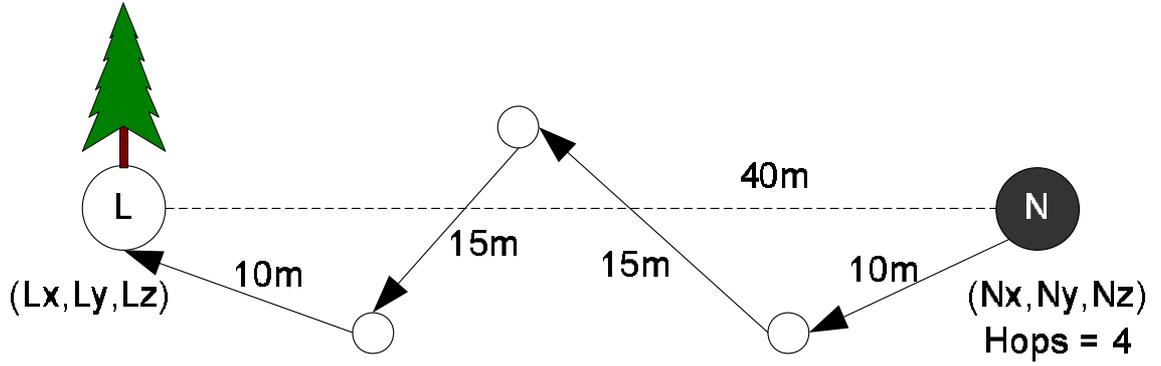


Figure 4.2: A reference model for the various landmark-based virtual coordinates.

locations (N_x, N_y, N_z) , which are in fact irrelevant to the landmark-based coordinates. The *straightline* coordinate is the Euclidean distance between node N and the landmark L, i.e., 40m. The *hops* coordinate is hop count (4) along the minimum-cost path. The *path-distance* coordinate is 50m along the minimum-cost path, which is the sum of the lengths of links along the path. In a line topology, *path-distance* and *straightline* are identical, and represent the ideal landmark-based coordinates, regardless of the method used to estimate the length of a link. When nodes' geographic locations are unknown, all but the hops coordinate are hard to obtain.

The *extrapolated* coordinate is the *path-distance* coordinate with the estimation of a link's length (*per-link length*) without relying on any geographic location. One such technique for estimating the length of a link has already been presented in Section 2.3.2. In fact, GHVR treats a landmark like the sink in Section 2.3.2, and obtains the extrapolated coordinate with respect to that landmark by using the same method as described in Section 2.3.2.

For ease of exposition, we redescribe briefly how to obtain the extrapolated height of a node with respect to the landmark. In the method proposed in Section 2.3.2, each node runs the link-quality measurement scheme based on both the periodic

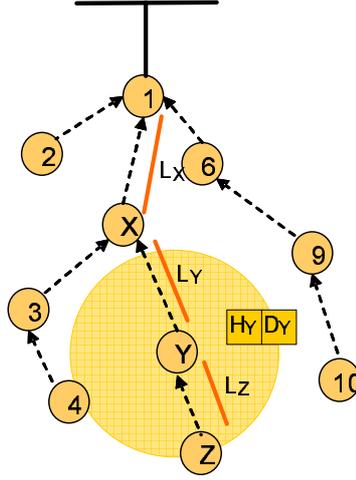


Figure 4.3: After overhearing that node Y advertises its current height (H_Y) and height-determinant (D_Y), node Z updates its own H_Z and D_Z based on both the measured length-estimate (L_Z) and the advertised information (H_Y and D_Y).

exchange of broadcasts and the result of unicast transfers to neighbors. Given such link-quality (dlvrate), the length-estimate is estimated as

$$\sqrt{\text{dlvrate}^{-1}} \quad (4.1)$$

The heuristic in Section 2.3.2 then takes an induction step over the minimum-cost path.² For immediate 1-hop neighbors of the landmark, their initial height is their initial length-estimate to the landmark. Referring to Figure 4.3, we assume that Z has Y as the next hop on the minimum-cost path to the landmark (node 1), and that Y periodically broadcasts its current height (H_Y) and height-determinant (D_Y). Every neighbor around Y then updates the link-quality between Y and itself.

Besides updating the link-quality, Z updates H_Z and D_Z of its own since Y is the next hop on the minimum-cost path from Z to the landmark:

$$H_Z \leftarrow H_Y + L_Z H_Y / D_Y \text{ and } D_Z \leftarrow D_Y + L_Z \quad (4.2)$$

²In our *ns-2* simulation, the link cost is dlvrate^{-1} , even though other definitions of the link cost can be used.

where L_Z is the length-estimate measured by Z for the link between Y and itself. This final H_Z becomes the extrapolated coordinate of node Z with respect to that landmark.

4.3.3 Distance Function

The distance function used by GHVR is the standard Euclidean distance function over the landmark-based coordinate space. However, since the landmark-based coordinates are not necessarily orthogonal to each other, one might wonder whether the standard Euclidean distance function can still serve as a good distance metric to represent the proximity among nodes in the landmark-based coordinate space.

Given two l -dimensional virtual coordinates $\underline{u} = (u_1, u_2, \dots, u_l)$ and $\underline{v} = (v_1, v_2, \dots, v_l)$, the following distance functions $\delta(\underline{u}, \underline{v})$ have been introduced in the literature: (1) the standard Euclidean distance, (2) the *Centered* distance [21] $\delta(\underline{u}, \underline{v}) = \sqrt{\sum((u_i^2 - \bar{u}) - (v_i^2 - \bar{v}))^2}$ where $\bar{u} = \sum u_i^2 / l$, $\bar{v} = \sum v_i^2 / l$, (3) the *Weighted* Manhattan distance [23] $\delta(\underline{u}, \underline{v}) = 10 \times \sum_{i \in C_k(v)} \max(u_i - v_i, 0) + \sum_{i \in C_k(v)} \max(v_i - u_i, 0)$, where $C_k(v)$ is the set of the k closest (in number of hops) landmarks to node v .

The term Weighted Manhattan distance is defined in this study, but not in [23]. Also, note that it violates the definition of the metric since it is not symmetric, i.e., $\delta(\underline{u}, \underline{v}) \neq \delta(\underline{v}, \underline{u})$. For non-Euclidean distance definitions, see the original papers. Unfortunately, they have not been evaluated against each other.

4.3.4 Greedy Forward

Packets are forwarded greedily to one of nodes in the neighborhood. Formally represented in a graph (V, E) with a given distance function $\delta : V \times V \rightarrow \mathbb{R}^+$ in the landmark-based coordinate space, node c forwards packets *greedily* to its neighbor $i \in N(c)$ such that $i = \arg \max_{x \in N(c)} (\delta(c, d) - \delta(x, d)) / C_{cx}$ and $\delta(c, d) - \delta(i, d) > 0$,

i.e., making positive forward progress. Here, C_{cx} is the link cost between nodes c and x .

The above greedy forwarding strategy is in fact NADV [46] over landmark-based coordinates, which is known to be the optimal forwarding for geographic routing with lossy wireless links. If the positive forward progress condition does not exist, the greedy forwarding cannot guarantee loop-freedom in the path.

4.3.5 Recovery from Greedy Failure

GHVR can suffer from the same void problem [9, 37] as geographic routing, although WMNs are likely to be densely populated. The void problem refers to a situation where no neighbors are geographically closer to the destination than the current node. In such a case, the greedy forward fails since there is no next hop that makes positive progress toward the destination.

Several recovery procedures against the void problem in the landmark-based coordinate space have been proposed in the literature. LCR [10] backtracks to the best upstream node, whereas BVR [23] backtracks to the parent toward the landmark closest to the destination and makes that landmark flood the route request whenever necessary. Unlike LCR or BVR, GLIDER [21] never backtracks or falls back on the closest landmark; it always floods packets whenever it encounters a greedy failure.

The recovery mechanism used by GHVR is based on BVR's solution. Figure 4.4 illustrates this recovery. While backtracking to the destination's CL, greedy forwarding will be resumed whenever there is a greedy next hop toward the destination. It is possible that packets get to CL before reaching the destination. In such a case, CL of the destination performs efficient flooding based on the history of packet forwarding to CL. Note that a history of packet forwarding to CL can be accumulated since

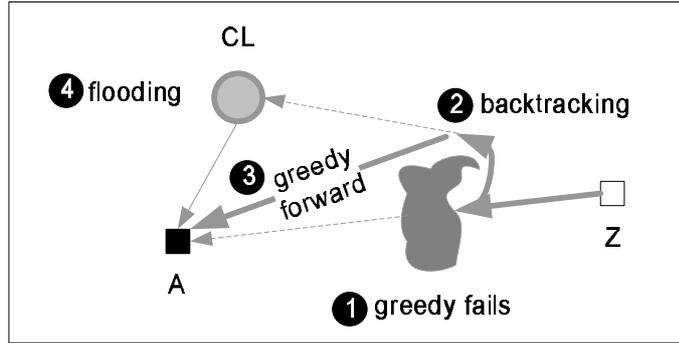


Figure 4.4: Illustration of recovery from a greedy failure.

every node periodically publishes its identifier and landmark-based coordinate to the nearest landmark, as illustrated in Figure 4.1(a).

4.4 Evaluation

4.4.1 Wireless Channel

To make our *ns-2* simulation as realistic as possible, we followed several recommendations in [42], one of which is to use the shadowing propagation model, because it can simulate wireless link characteristics such as link asymmetry and time-varying link-quality. The *ns-2* shadowing model and other configurations are set up as in Table 4.1; these settings are derived from our implementation of the Proxim Orinoco adapter [58] for a semi-open environment.

4.4.2 Topologies and Flows

An MIT indoor wireless sensor network testbed called MistLab [51] is reproduced as an unplanned topology in our *ns-2* simulation. The simulated MistLab, simply called nsMistLab, has sixty-one 802.11b nodes spread non-uniformly over a square field as shown in Figure 4.5. Besides this basic topology, 100 nodes are randomly added in the field. A landmark is also installed at each of the four corners of the

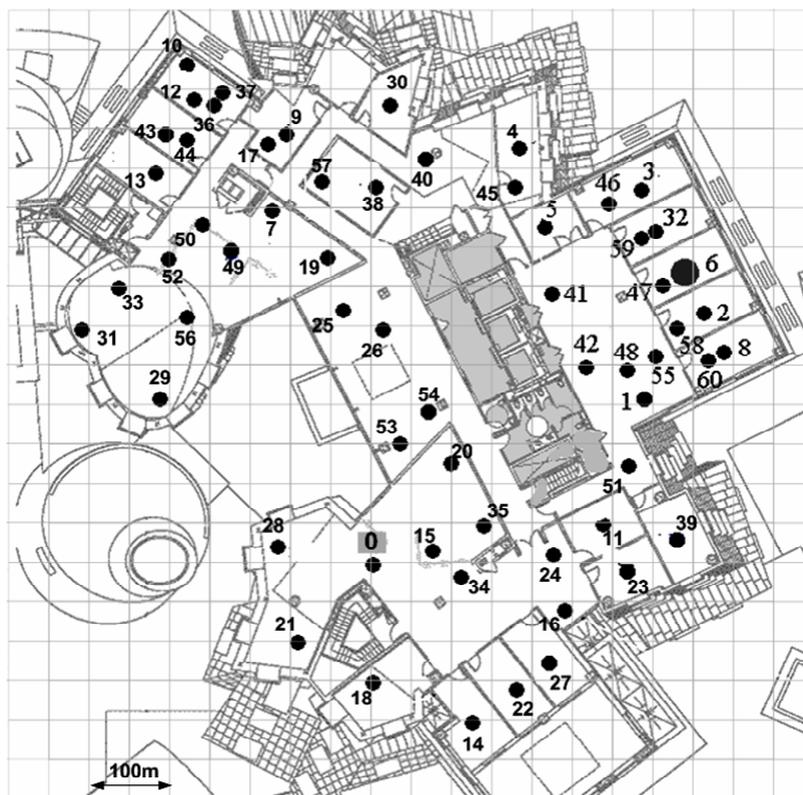


Figure 4.5: MistLab [51], a non-uniform real-world wireless sensor network testbed with 61 nodes over a field of 16,000 square feet, is reproduced in our *ns-2* simulation as an nsMistLab 802.11b wireless network over a field of $1\text{km} \times 1\text{km}$.

Table 4.1: Configurations for 802.11b with the shadowing model.

Path loss exponent		4	
Shadowing deviation		4,5,7,9	
Reference distance		10	
Frequency	2.4 GHz	Transmit power	15 dBm
CPTthresh	10	CSTthresh	-105 dBm
RXThresh (11 Mbps) [†]		-79.84 dBm, (95% at 50m)	
RXThresh (5.5 Mbps) [†]		-85.68 dBm, (95% at 70m)	
RXThresh (2 Mbps) [†]		-90.05 dBm, (95% at 90m)	
RXThresh (1 Mbps) [†]		-94.31 dBm, (95% at 115m)	
Data Rate		11 Mbps (w/o auto rate)	
Basic Rate		2 Mbps	
MAC headers		(24,14,14,28) bytes @ Basic Rate	
PLCP length		192 bits @ 1 Mbps	
SIFS	10us	DIFS	50us

[†] These thresholds are configured for the path loss exponent 4 and the shadowing deviation 4, and remain unchanged regardless of changes in both parameters.

field.

Every node except node 0 becomes a data sink. Node 0 sends continually an average MAC-level payload of 1500 bytes, or $1500+U(-750, 750)$ bytes, to every other node in the network. Data is sent every 4 seconds while changing the destination in a round-robin manner.

4.4.3 Performance Metrics and Questions

Since the primary goal of this chapter is to compare continuous and discrete coordinates, only the greedy forwarding defined in Section 4.3.4 is considered because recovery from a greedy failure has nothing to do with the type of coordinate used. We also limit the maximum number of transmission attempts per packet at each hop. Thus, packets suffering from too many transmission failures at each hop will be dropped after reaching a pre-defined retry limit. Currently, the retry limit is arbitrarily set to 40 per packet at each hop. Throughout the simulation, we refer to

this bounded greedy forwarding as simply *greedy routing*.

Under this greedy routing, the main performance metric to evaluate is the delivery ratio. All the virtual coordinates and the absolute geographic coordinate described in Figure 4.2 are evaluated. The delivery ratio under each type of coordinate is compared to that under the extrapolated coordinate, and relative performance improvements in terms of median and mean are presented.

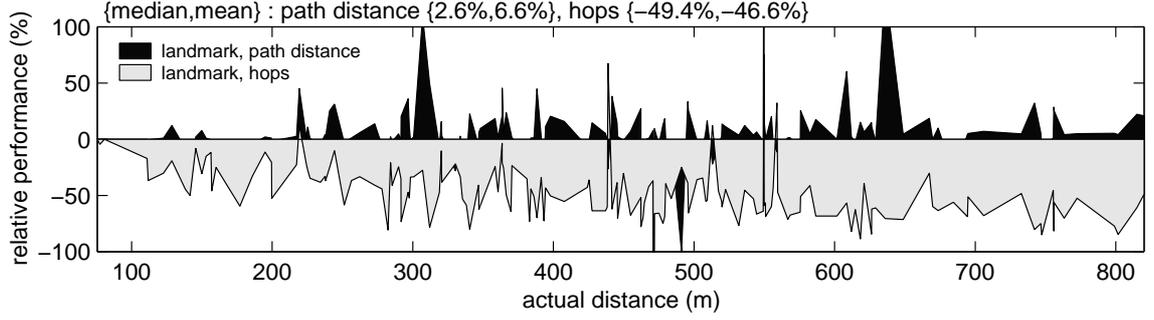
Our evaluation will primarily answer the following questions:

- By how much does the extrapolated coordinate outperform the hops coordinate?
- By how much does the ideal path-distance coordinate outperform the extrapolated coordinate?
- Is the standard Euclidean distance function still a good distance metric in the landmark-based virtual coordinate space?

4.4.4 Main Results

Figure 4.6(a) shows the node-by-node comparisons obtained by averaging the results of five independent runs with different random seeds for the same topology set 0. The x-axis is sorted according to the distance between node 0 and each destination. Figure 4.6(a) indicates that the delivery ratio under the extrapolated coordinate is improved by a factor of almost 2 over the delivery ratio under the hops coordinate in both median and mean numbers.

Evaluation against additional 9 different topologies did not change this conclusion. Figure 4.6(b) suggests that the extrapolated coordinate consistently outperforms the hops coordinate over these different topologies. The performance gains



(a)

set	absolute		landmark-based virtual coordinate					
	geo. locations		straightline		hops		path distance	
	m	u	m	u	m	u	m	u
0	9.72	21.93	5.97	14.71	-49.38	-46.56	2.61	6.58
1	0.00	45.28	-4.14	8.51	-57.99	-53.70	-2.44	12.27
2	13.90	27.06	11.11	22.81	-76.09	-70.85	17.80	304.41
3	24.62	111.54	25.00	56.91	-71.47	-67.89	21.00	304.19
4	0.00	7.39	0.00	-6.09	-56.82	-46.80	0.00	4.33
5	8.11	20.01	9.84	19.27	-53.64	-50.59	4.36	10.59
6	0.00	-14.12	0.00	-19.81	-58.90	-58.07	2.90	9.94
7	1.22	17.75	6.07	18.47	-56.10	-52.93	2.38	14.69
8	10.82	42.70	10.82	50.47	-64.49	-57.60	2.60	15.90
9	-19.05	-32.69	-22.95	-31.22	-51.95	-49.67	2.56	8.83

(b)

Figure 4.6: The improved(+) or reduced(-) delivery ratios (%) of the greedy routing relative to the extrapolated coordinate over 10 different topologies. The m and u represent the median and the mean of comparison results. Node-by-node comparison results for the set 0 in (b) are plotted in (a).

from the extrapolated coordinate over the hops coordinate are greater than a factor of 2 in most cases. Moreover, the extrapolated coordinate loses only 2~4% of performance over the ideal path-distance coordinate all cases but sets 2 and 3.

It is interesting to see that the greedy routing seems to perform slightly better under the path-distance coordinate than under the absolute coordinate. This might open a new design space for improving the standard geographic routing even if the nodes' geographic locations are easily obtainable.

Note that the hops coordinate can also be defined over the minimum-hop path [10, 21]. Related simulation results, not presented in this chapter, show that using the minimum-hop path is not as good as using the minimum-cost path. Thus, defining landmark-based coordinates along the minimum-hop path will not be considered.³

4.4.5 Impact of Wireless Channel Variations

Figure 4.7(a) and (b) suggest that the extrapolated coordinate still outperforms the hops coordinate even in the face of variations in wireless channel quality and the random packet drops at some nodes. Interestingly, the extrapolated coordinate is found to be comparable to the ideal path-distance coordinate in this experiment.

These results are obtained with the wireless channel varied as follows. A total of 12 scenarios, each lasting 500 seconds, are considered on top of the basic wireless channel setting specified in Table 4.1. Each scenario, denoted by the labels (S0, S1, ..., S12) in Figure 4.7(c), is rolled out over the entire simulation of length 6,000 seconds. Changes are made in the wireless channel parameter, the node parameter, and the random packet-loss parameter at the physical layer.

In Figure 4.7(c), a large path loss (exponent) indicates a sharp drop in the average received signal strength over distance. A large (shadowing) deviation in the figure means more smoothed exponential decay but more variations in the received signal strength. The node constraint “_” means that any change will be global across all the nodes, whereas two node constraints G0 and G1 mean that a change will be applied only to those nodes satisfying G0 or G1.

Selection of nodes subject to both constraints is arbitrary. For example, the node

³For interested readers, the hops coordinate over the minimum hop count yields the following relative performance as compared to the extrapolated coordinate for 10 different sets; $\{(m\%,u\%)\}=\{(-71.43, -64.60), (-85.71, -78.97), (-85.62, -73.86), (-92.85, -85.08), (-85.53, -76.07), (-81.23, -73.24), (-85.14, -76.54), (-82.91, -74.73), (-87.33, -78.57), (-87.49, -76.15)\}$.

constraint G0 that changes the shadowing deviation parameter will be applied to those nodes whose IDs are 1 (modulo 3). Similarly, the node constraint G1, which drops the received packets randomly at the physical layer with a probability of 0.3, will be applied to the nodes whose IDs are 54, 20, 35, 24, 11, 51, 42, 41, 45, and 26, regardless of the type of topologies used. Note that the node constraint G1 is not applied in the last scenario S12.

4.4.6 Impact of Greedy Routing Variations

We define an enhanced version of the greedy routing, called *connectivity-first greedy routing* (CFGR), which trades optimality for reliability in link selection. CFGR maintains and uses a routing table that consists of neighbors whose measured bidirectional link-quality is known to be above some threshold. Thus, unlike the original greedy routing, it attempts to directly transmit packets if the destination is found in the routing table. Otherwise, it follows the original greedy routing. Note that evaluation under the original greedy routing represents the lowest performance that could be attained without modifying the optimal link-selection strategy described in Section 4.3.4.

Using CFGR under the hops coordinate turns out to significantly improve the delivery ratios for immediate neighbors, as seen in Figure 4.8(a). This is because nodes keep transmitting packets within the maximum retry limit until the transmission becomes successful. However, we still observe performance decline over distance. Such a decline is probably due to the disparities in transmission range between broadcast and unicast. Recall that the hops coordinate between nodes is likely to differ when nodes are separated by the broadcast coverage. However, the broadcast coverage is much wider than that of unicast in the IEEE 802.11-based radio. Thus, the next

hop in the hops coordinate is likely to be situated near the perimeter of broadcast coverage of the current node. This, in turn, increases the chance of dropping unicast packets.

Using additional landmarks mitigates this problem to some degree, because it produces finer-grained coordinates even under the hops coordinate. Nevertheless, performance under the hops coordinate is still the poorest among the virtual coordinates studied, as shown in Figure 4.8(b). This is also confirmed by [23]; according to the simulation results in Section 4.2 of this paper, 10 destination-dependent landmarks out of 30 randomly-elected landmarks achieved the greedy routing performance comparable to that of using true positions.

Additional simulations over nsMistLab support our claim that the extrapolated coordinate is better than the hops coordinate. As shown in Figure 4.9, the hops coordinate consistently yields about 30% less performance than the extrapolated coordinate.

4.4.7 Impact of Auto Rate Algorithms

To eliminate potential packet drops due to the disparity between the ranges of broadcast and unicast in IEEE 802.11, we implemented a variant of the receiver-based auto rate selection algorithm described in [28]. In the receiver-based rate selection algorithm, a receiver determines the best rate for each neighbor by considering the channel conditions and informs a sender of the best rate. The original auto rate selection in [28] takes steps to reserve the channel before sending data, which allows the receiver to inform the sender of the best rate right before actual data delivery. On the other hand, our implementation requires every node to periodically advertise the best unicast rate for each of its neighbors by using the same logic as in [28], but

based on recently-measured channel conditions.

Figure 4.10 shows that the chosen auto rate algorithm improves the delivery ratio across all virtual coordinates. Nevertheless, there are still noticeable differences between the hops and extrapolated coordinates. At the same time, the differences between the ideal path-distance and the extrapolated coordinates become indiscernible, because only the ideal path-distance performs better (by about 2%) than the extrapolated, and both are the same in terms of the median.

Running the connectivity-first greedy routing with the auto rate feature enabled is shown (Figure 4.11) to reduce the performance gap between the other coordinates and the extrapolated coordinate. However, as already seen in Figure 4.10, the performance under the hops coordinate is still consistently inferior to that under the extrapolated coordinate, whereas the performance of the extrapolated coordinate approaches that of the ideal path-distance coordinate.

Since using either the connectivity-first greedy routing or the auto-rate feature consistently demonstrates the performance superiority of the extrapolated coordinate, our remaining evaluation is done with the original greedy routing without enabling the auto rate feature.

4.4.8 Impact of Length-estimate

Given the same wireless channel variations as in Figure 4.7(c), different length-estimates are evaluated in Figure 4.12. Note that given the link-quality (`dlvrate`), the default length-estimate is $\sqrt{\text{dlvrate}^{-1}}$ as defined in Section 4.3.2. The results in this figure suggest that $\sqrt{\text{dlvrate}^{-1}}$ yields slightly better performance than dlvrate^{-1} . However, using dlvrate^{-2} turns out to widen the performance gap between the ideal path-distance and the extrapolated. Note that this experiment affects

performance only under the extrapolated coordinate.

4.4.9 Impact of Placement and Number of Landmarks

The simulation result (Sec. 4.2 in [23]) suggests that there is a trade-off between the number of landmarks and routing performance as well as per-packet overhead. For example, it is shown in [23] that about 10 beacons need to be installed when landmarks are randomly elected. However, the planned installation of landmarks is shown to be much better than a large number of randomly-elected landmarks. Sec. 4.1.2 in [10] shows that a landmark at each of the four corners of a rectangle is enough to achieve high routing performance, and that having more than 4 regularly-planned landmarks makes no difference in improving routing performance.

Based on these observations, a landmark is pre-established at each of the four corners of the field, which is the default in our simulation, and then up to 8 additional landmarks are randomly placed. This mixed configuration helps us assess the impact of the placement of landmarks as well as the impact of the number of landmarks on the routing performance.

Figures 4.13 and 4.14 summarize the results: the extrapolated coordinate works much the same as the path-distance coordinate. The median performance comparison shows that the difference between the two is indiscernible, even with random installation of additional landmarks. Although the path-distance performs 2.6% better than the extrapolated when 4 landmarks are used, the two are otherwise the same. As for the mean performance, the path-distance shows slightly improved performance as the number of landmarks increases from 4 to 12 by 2: (6.58%, 14.71%, 10.83%, 12.44%, 10.15%).

As expected in [23], the hops coordinate performs better as the number of land-

marks increases. Two points are noticeable from Figures 4.13 and 4.14. First, installing additional random landmarks does not appear helpful even for the hops coordinate; installing landmarks regularly seems more important than installing a large number of them randomly. Second, in spite of additional landmarks the hops coordinate yields poorer performance than the extrapolated coordinate; its median and mean are $(-49.38\%, -40.00\%, -25.00\%, -25.00\%, -21.82\%)$ and $(-46.56\%, -34.72\%, -21.09\%, -19.80\%, -23.28\%)$ as the number of landmarks increases from 4 to 12 by 2, respectively.

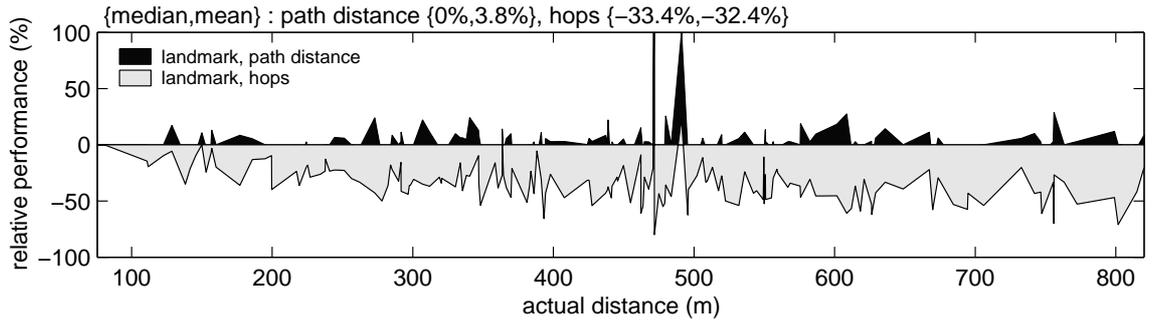
4.4.10 Impact of Distance Function

The authors of [10] concluded that the Euclidean distance is the better metric by comparing it with the usual Manhattan distance on the unit disc radio model. Our comparison results for the distance functions are plotted in Figure 4.15 and indicate that the Euclidean distance improves routing performance more than Weighted Manhattan or Centered distance. Recall that this distance function comes into play when selecting a next hop that makes the most efficient forward progress as defined in Section 4.3.4. The poor performance of the “Weighted Manhattan” function in BVR seems to be related to the asymmetric weight factor in its distance computation. The median and mean performances of the “Centered distance” in Figure 4.15 indicate that its theoretical assumption of “achieving the non-local minima at all (Sec. V.B [21])” does not hold for a realistic wireless channel.

4.5 Conclusion

We first addressed the lack of a fine-grained landmark-based coordinate for routing packets in wireless networks because the existing work uses the hop count along the minimum hop or cost path to the elected landmarks. We then presented a one-to-

one routing protocol based on a novel continuous landmark-based coordinate. The proposed coordinate is called the *extrapolated coordinate* and measures the length along the minimum-cost path to the elected landmarks. We further defined the ideal path-distance coordinate and comparatively evaluated three schemes — the traditional hop count, the proposed extrapolated, and the ideal path-distance — with realistic wireless network settings. Our simulation results demonstrated that the extrapolated coordinate improved routing performance by a factor of almost 2, over the traditional hop count coordinate, and performs nearly as well as the ideal path-distance coordinate. We also investigated the different types of distance functions over the landmark-based coordinate space. Our evaluation indicated that the standard Euclidean distance definition still works best even in the virtual coordinate space.



(a)

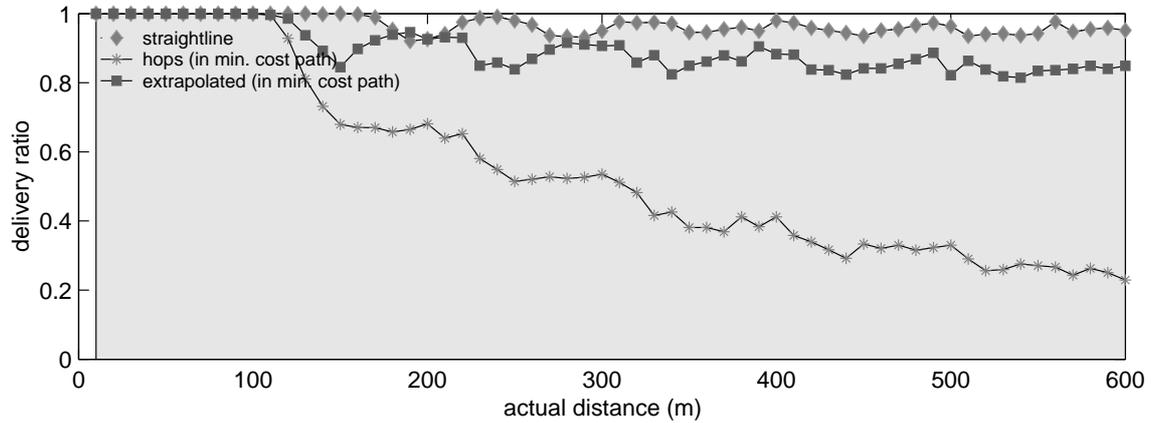
set	absolute		landmark-based virtual coordinate					
	geo. locations		straightline		hops		path distance	
	m	u	m	u	m	u	m	u
0	2.67	7.87	3.65	8.16	-33.42	-32.37	0.00	3.73
1	2.50	11.44	0.00	4.82	-47.66	-46.64	0.00	1.65
2	5.13	2.59	5.26	2.27	-37.14	-36.23	2.60	5.15
3	2.85	12.12	5.13	13.24	-30.49	-29.32	0.00	9.23
4	0.00	-3.52	0.00	-3.41	-39.43	-38.04	0.00	-0.15
5	2.74	7.70	5.13	8.31	-38.35	-37.09	0.00	3.06
6	0.00	-4.79	0.00	-6.79	-42.87	-41.25	2.50	3.17
7	5.72	6.10	3.08	5.95	-35.51	-34.62	0.00	1.41
8	5.79	10.96	7.40	10.19	-44.44	-43.85	2.63	3.60
9	-7.69	-23.81	-6.78	-23.04	-36.99	-38.64	0.00	2.59

(b)

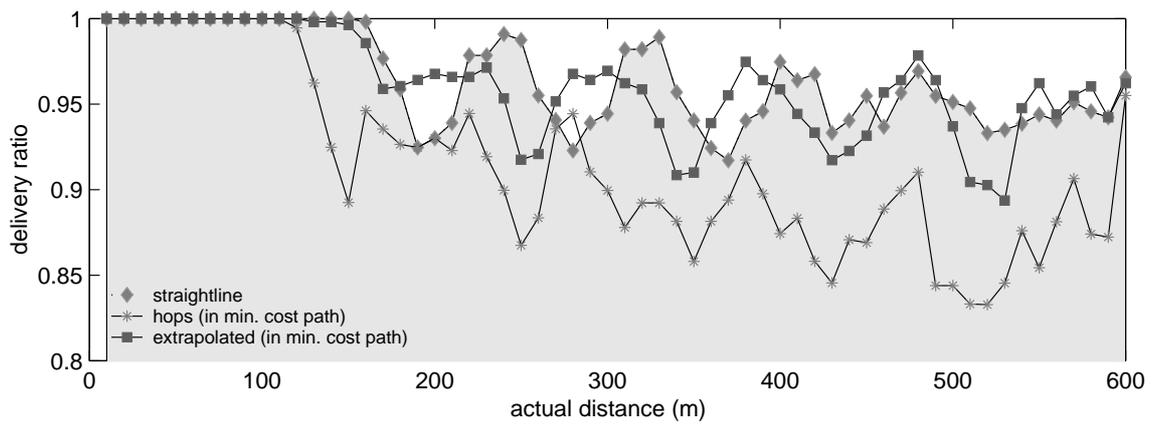
	S1	S2	S3	S4	S5	S6
path loss	4	4	4	4	4	4
deviation	4	9	9	7	7	7
node	-	-	-	G0+G1	-	-
	S7	S8	S9	S10	S11	S12
path loss	4	4	4	4	4	4
deviation	7	5	5	5	5	4
node	-	G0+G1	-	-	-	G0

(c)

Figure 4.7: The improved(+) or reduced(-) delivery ratios (%) of the greedy routing relative to the extrapolated coordinate over 10 different topologies. Node-by-node comparison results for the set 0 in (b) are plotted in (a). The wireless channels are changed according to (c).



(a) delivery ratio under one landmark at 0



(b) delivery ratio under two landmarks at 0 and 600

Figure 4.8: Performances of the *connectivity-first greedy routing* with a line topology under the straightline, hops, and extrapolated coordinates. Note the scale of Y axis in (b).

set	absolute		landmark-based virtual coordinate					
	geo. locations		straightline		hops		path distance	
	m	u	m	u	m	u	m	u
0	2.63	4.88	2.57	4.16	-28.95	-28.50	0.00	1.00
1	5.31	17.85	2.53	8.61	-31.14	-31.31	-1.22	-0.16
2	5.48	4.85	5.13	-0.29	-26.49	-24.11	2.81	5.26
3	4.35	8.57	2.99	6.65	-25.32	-25.18	2.50	3.81
4	2.36	0.61	0.00	0.54	-27.15	-26.62	2.50	4.53
5	3.08	8.98	2.63	8.67	-29.57	-29.15	0.00	2.63
6	0.00	-3.05	0.00	-6.38	-34.22	-33.20	2.53	5.38
7	3.64	6.32	3.59	7.11	-31.14	-29.29	0.39	4.03
8	7.90	10.93	5.56	9.85	-40.61	-36.23	0.13	0.28
9	-2.90	-21.36	-7.91	-22.75	-31.47	-31.88	2.56	3.33

Figure 4.9: The improved(+) or reduced(-) delivery ratios (%) of the *connectivity-first* greedy routing relative to the extrapolated coordinate over 10 different topologies.

set	absolute		landmark-based virtual coordinate					
	geo. locations		straightline		hops		path distance	
	m	u	m	u	m	u	m	u
0	2.50	3.46	2.50	3.98	-22.50	-23.71	0.00	-0.52
1	3.92	17.74	2.67	16.65	-39.15	-38.76	0.00	1.06
2	2.50	2.60	2.53	3.98	-26.60	-27.14	0.00	1.64
3	2.56	6.21	2.56	7.57	-20.00	-21.61	0.00	4.46
4	2.50	8.57	0.00	4.55	-23.73	-25.10	0.00	-0.16
5	2.50	6.75	2.50	6.51	-21.95	-22.92	0.00	1.53
6	2.50	5.64	2.50	4.82	-24.39	-25.90	0.00	2.28
7	2.50	2.24	2.50	3.10	-25.32	-26.40	0.00	0.78
8	2.50	7.07	2.50	5.73	-28.58	-29.68	0.00	1.71
9	0.00	1.07	0.00	2.00	-25.00	-27.72	0.00	0.32

Figure 4.10: The improved(+) or reduced(-) delivery ratios (%) of the greedy routing relative to the extrapolated coordinate when a variant of the best auto rate algorithm [28] is used.

set	absolute		landmark-based virtual coordinate					
	geo. locations		straightline		hops		path distance	
	m	u	m	u	m	u	m	u
0	2.50	3.92	2.50	4.07	-7.88	-11.11	0.00	0.87
1	2.56	33.08	2.50	14.64	-17.58	-20.63	0.00	0.62
2	2.50	5.93	2.50	5.33	-10.25	-11.58	0.00	3.43
3	2.50	5.34	2.50	5.60	-7.50	-10.23	0.00	2.16
4	0.00	1.27	0.00	0.35	-14.81	-15.85	0.00	0.09
5	2.50	6.37	2.50	6.11	-10.58	-11.83	0.00	2.17
6	2.50	3.77	0.00	3.18	-14.63	-17.15	0.00	0.92
7	2.50	2.91	2.50	2.89	-16.44	-17.58	0.00	-1.11
8	2.50	4.58	2.56	4.42	-18.21	-20.74	0.00	0.54
9	0.00	0.18	0.00	0.21	-10.52	-15.08	0.00	1.32

Figure 4.11: The improved(+) or reduced(-) delivery ratios (%) of the *connectivity-first* greedy routing relative to the extrapolated coordinate when a variant of the best auto rate algorithm [28] is used.

set	path-distance vs. extrapolated virtual coordinates					
	sqrt		identity		quadratic	
	m	u	m	u	m	u
0	0.00	3.73	3.23	5.42	63.06	89.25
1	0.00	1.65	0.00	5.19	58.43	175.35
2	2.60	5.15	0.00	6.46	76.72	126.11
3	0.00	9.23	5.98	18.53	46.99	71.78
4	0.00	-0.15	0.00	-0.05	64.08	123.99
5	0.00	3.06	2.67	5.88	46.90	64.12
6	2.50	3.17	5.13	8.07	72.07	96.95
7	0.00	1.41	0.00	1.92	51.72	91.58
8	2.63	3.60	0.00	-0.41	51.00	86.25
9	0.00	2.59	0.00	6.17	77.27	111.59

Figure 4.12: The improved(+) or reduced(-) delivery ratios of the greedy routing relative to the extrapolated coordinate when its length-estimate is changed \sqrt{c} (default), c , c^2 where $c = \text{dlvrate}^{-1}$.

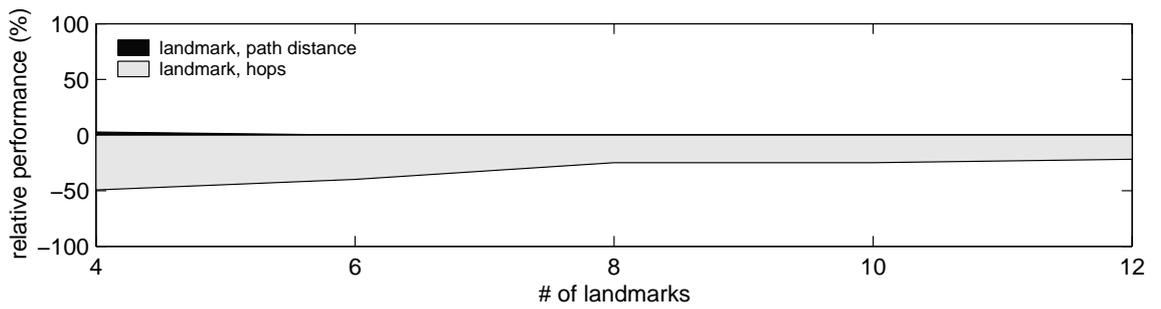


Figure 4.13: The greedy routing performance (in terms of the median value) relative to the extrapolated coordinate with the number of landmarks varied.

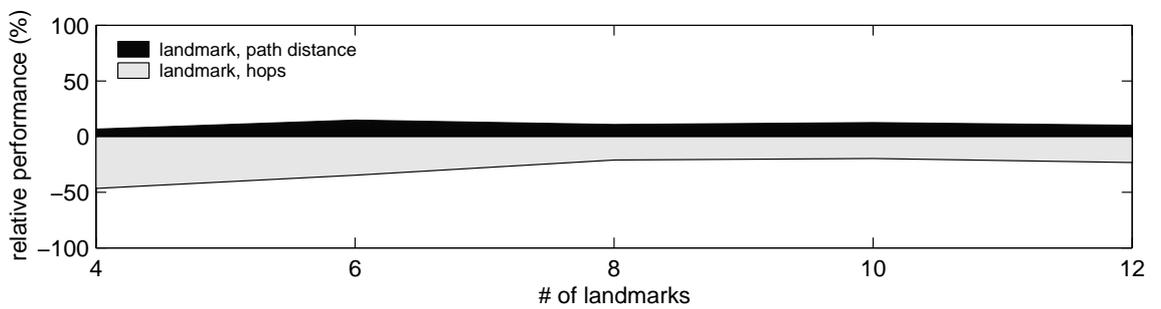


Figure 4.14: The greedy routing performance (in terms of the mean value) relative to the extrapolated coordinate with the number of landmarks varied.

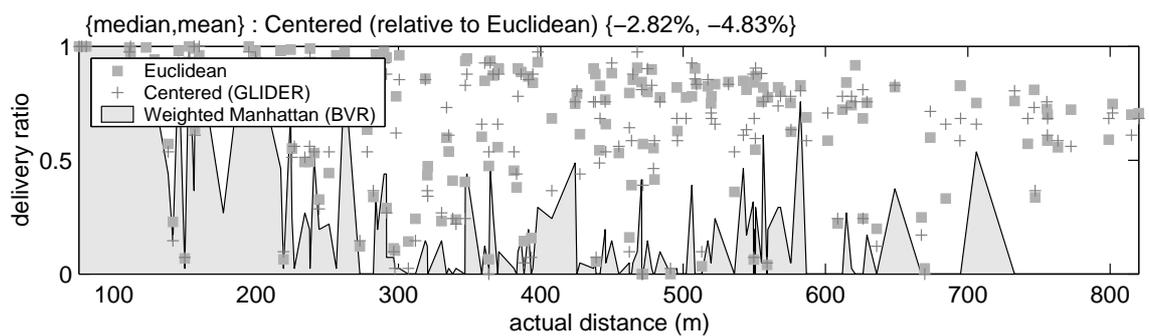


Figure 4.15: The delivery ratios of the greedy routing under different distance functions with the ideal path-distance coordinate.

CHAPTER V

MULTI-RATE OPPORTUNISTIC GREEDY ROUTING

5.1 Introduction

Multi-channel multi-radio IEEE 802.11 wireless devices are being increasingly deployed in many wireless networks, such as wireless mesh networks. These devices can offer different data rates by making a channel/radio switch while taking into account the channel conditions in each device's neighborhood. Even with a single radio, such a dynamic data-rate change based on the wireless channel conditions is possible in most 802.11 wireless devices equipped with the forced or auto rate feature [5, 6, 28, 35]. We will henceforth use a generic term "multi-rate" to describe a radio's capability of providing different data rates via physical features or rate-control algorithms. Whichever is needed, the multi-rate provided by a rate-control mechanisms is called the *soft* multi-rate.

The multi-rate feature of a device can be exploited regardless of whether the device is used in ad-hoc or infra-structured, stationary or mobile networks. The main advantage of a multi-rate radio is its creation of opportunities to efficiently use wireless resources to improve the network capacity or attain the high unicast throughput, primarily because it can either reduce the number of packet drops (due

to channel errors) by using a low data rate or avoid co-channel interferences by switching to different channels.

Recently, numerous routing protocols [6, 17, 18, 59] have exploited this multi-rate feature to improve the network capacity or the unicast throughput. However, they are all topology-based routing protocols that maintain information on routes between nodes. In such protocols, the multi-rate feature of a radio cannot be exploited to the maximum degree possible, because any switch from the current link rate or radio to another affects current link costs in the neighborhood, and such link-cost changes need to be propagated through the network so as to enable discovery of new optimal routes between nodes. Thus, any potential advantage of using multi-rate radios comes with the route update overhead, which, in turn, limits the opportunistic use of the multi-rate feature to improve network capacity or throughput.

In contrast to the use of multi-rate radios in topology-based routing protocols, we propose to enable *geographic routing* to exploit the multi-rate feature so as to improve network capacity or throughput. Geographic routing is chosen since it greatly simplifies the routing operation with nodes' location information. That is, it doesn't require nodes to maintain any route information and thus, need not update route information when the current link rate or radio at a node is changed. If there is no next hop geographically getting closer to a given destination than the current node, packets will be dropped at the current node and/or a recovery mechanism is invoked.¹ Otherwise, packets will be forwarded to the next hop geographically closest to destination. Thanks to this route-less property, geographic routing becomes nothing but a lookup-and-forward operation, allowing for high per-packet adaptability to the network condition.

¹However, recovery from a greedy failure will not be considered any further in this chapter since it is less influential on throughput than the greedy forwarding itself.

The high adaptability of geographic routing on a per-packet basis is very important to multi-rate multi-hop wireless networks. When faced with neighbors using different data rates or channels, one might ask whether there is any packet forwarding strategy that benefits from these diverse neighbors in maximizing throughput while keeping the neighbor-switching cost as small as possible.²

This important question is answered with our main contributions outlined as follows.

- We propose how to make geographic routing exploit the multi-rate feature so as to maximize the throughput. To our best knowledge, this is the first attempt of its kind.
- We devise a novel greedy packet forwarding strategy that determines which neighbor to be attempted first by using the generic reward and penalty model associated with packet transmissions. Since the strategy is expressed in the generic reward and penalty model, it can take various practical forms.
- Two new practical link metrics are developed for geographic routing in multi-rate multi-hop wireless networks, based on the generic reward and penalty strategy. Unlike the existing link metrics that include only a single performance factor, the proposed link metrics combine link-quality, delay, and data rate.
- Of the two new link metrics, the **nmep** link metric is defined when no penalty is incurred for a neighbor-switch.
- Of the two new link metrics, the **pmep** link metric is defined with the cost in terms of the number of transmission attempts reflecting the per-hop transmis-

²Other neighbors will be tried upon failure of a transmission attempt until the packet is successfully forwarded or dropped.

sion delay.

The rest of this chapter is organized as follows. Section 5.2 discusses the related work. Section 5.3 states the problem and derives a generic packet forwarding strategy. Section 5.4 develop two multi-rate link metrics from the generic strategy. Section 5.5 evaluates the metrics using simulation. Finally, Section 5.6 concludes the chapter.

5.2 Related Work

Throughput improvement via multi-radio diversity becomes the problem of reducing interference by using different channels [18, 59]. The data channel can be switched dynamically on a per-packet basis [2, 40, 43] or can be statically assigned for a certain period of time as part of capacity planning [65]. Channel-switching is known to take a few hundred microseconds to a few milliseconds [44].

MuGR ignores the details of how a channel or a radio is switched. It only sees the rate or bandwidth of a resultant channel or radio and the cost required to make such a switch.

Auto rate algorithms can be classified as sender-based [6, 45] or receiver-based [28]. The sender-based approach changes the data rate based on the result of every transmission attempt but fails to accurately reflect the channel quality around the receiver. The receiver-based approach is based on the premise that channel-quality information is best obtained at the receiver. Usually, the receiver informs the sender of the channel condition around itself during the packet exchange to reserve the channel. The receiver determines the best rate by comparing the received signal strength against the preset thresholds.

Compared to MuGR, the original geographic routing [8, 37] only considers the progress toward destination. The expected progress is proposed in [63] and the

normalized advance in [46]. Similarly, metrics making a tradeoff between advanced distance and delay are defined in [27, 71]. All of these metrics make a tradeoff between advanced distance and another single performance measure. Rather than focusing on a single performance measure, the two link metrics **nmep** and **pmep** proposed in MuGR combine link-quality, delay, and neighbor-switch cost.

The main advantage of geographic routing is its scalability, adaptability, and simplicity. In particular, it can switch to a different next hop if the current next hop is unavailable or inadequate for performance reasons, without reconstructing any alternative routes. Practical use of geographic routing, however, must resolve several issues, such as failure of a planar-based recovery on real radios [39] or lack of an efficient location service [22]. MuGR uses the existing results for robust recovery or an efficient location service.

5.3 Generic Opportunistic Greedy Forwarding

We first address how to forward packets in multi-rate multi-hop wireless networks with a generic reward and penalty model. Then we describe the design of MuGR (Multi-rate Opportunistic Greedy Routing) with practical link metrics to maximize the unicast throughput in multi-rate multi-hop wireless networks.

We introduce an abstracted view of multi-rate multi-hop wireless networks, as illustrated in Figure 5.1. Parameters in the abstracted network model are **p** (delivery ratio), **a** (advance), **b** (data rate), and **c** (cost). There exists a link between two nodes with probability **p**. The advance **a** is the distance advanced by a node toward a given destination, e.g., **a** for node 1 is $|SD| - |1D|$. The data rate **b** represents the bandwidth of a node's link. The cost **c** is an arbitrary number, representing the number of transmission attempts, the transmission delay, the channel synchronization latency,

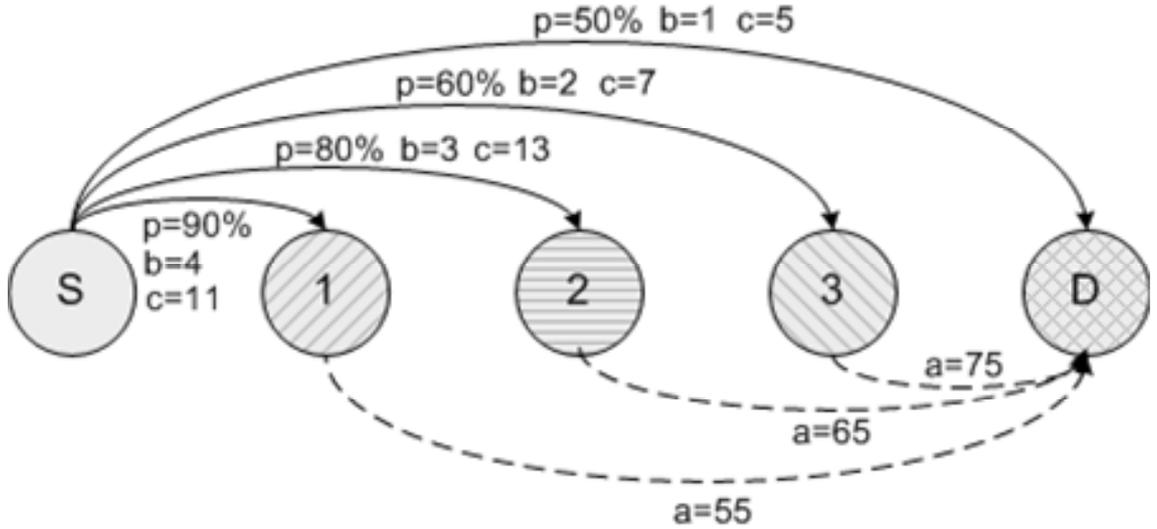


Figure 5.1: Parameters in an abstracted multi-rate multi-hop wireless network.

or any combination thereof.

Unless nodes are mobile, advance a will not change. However, other parameters are all dynamic. For example, loss of a link probe message directly affects link-quality p . Loss of the ACK for data being transmitted causes the underlying auto rate algorithm to decrease the data rate. Similarly, a per-packet or periodic channel-switching algorithm can cause a node to switch to a different channel, which, in turn, affects b . Therefore, we also need on-line estimation techniques for these dynamically-changing parameters. Note that any change in a , b , p , or any combination of them will trigger a change in c .

Suppose we are given a set of N neighbors $(n_1, n_2, \dots, n_{k-1}, n_k, n_{k+1}, n_{k+2}, \dots, n_N)$, a set of packet delivery ratios $(p_1, p_2, \dots, p_{k-1}, p_k, p_{k+1}, p_{k+2}, \dots, p_N)$, a set of current data rates $(b_1, b_2, \dots, b_{k-1}, b_k, b_{k+1}, b_{k+2}, \dots, b_N)$, and a set of advances $(a_1, a_2, \dots, a_{k-1}, a_k, a_{k+1}, a_{k+2}, \dots, a_N)$ made by individual nodes toward a given destination. Then,

we define positive and negative rewards for n_k :

$$\mathcal{R}(k) \triangleq \begin{cases} \mathcal{R}_s(k), & \text{TX to } n_k \text{ ok (positive rewards);} \\ \mathcal{R}_f(k), & \text{otherwise (negative rewards).} \end{cases} \quad (5.1)$$

Our goal is to find a sequence of neighbors that maximizes the combined rewards over all permutations of neighbors. The following theorem on the expected reward for a sequence of neighbors provides a clue for finding such a sequence.

Theorem 1 *Suppose a sequence $\underline{s} \triangleq (n_1, n_2, \dots, n_{k-1}, i = n_k, j = n_{k+1}, n_{k+2}, \dots, n_N)$. The expected reward on this sequence $\mathcal{R}(\underline{s})$ has the following inter-exchange property.*

$$\begin{aligned} E[\mathcal{R}(n_1, \dots, i, j, \dots, n_N)] &\geq E[\mathcal{R}(n_1, \dots, j, i, \dots, n_N)] \\ \Leftrightarrow \frac{1 - P_i}{P_i} \mathcal{R}_f(i) + \mathcal{R}_s(i) &\geq \frac{1 - P_j}{P_j} \mathcal{R}_f(j) + \mathcal{R}_s(j) \end{aligned}$$

where $P_i = 1 - (1 - p_i)^{m_i}$ and m_i is the maximum trial limit for node n_k .

Proof 1 *See Appendix A.*

5.4 Derivation of Practical Multi-rate Metrics

5.4.1 Unpenalized Multi-rate Metric

Since there is no penalty associated with a neighbor-switch upon transmission failure, the reward and penalty for node n_k are defined as:

$$\mathcal{R}(k) \triangleq \begin{cases} \mathcal{R}_s(k) \triangleq a_k p_k / \tau(l, b_k), & \text{TX to } n_k \text{ ok;} \\ \mathcal{R}_f(k) \triangleq 0, & \text{otherwise.} \end{cases} \quad (5.2)$$

where $\tau(l, b_k)$ returns the estimated transmission delay given a packet of length l and a transmission rate b_k .

Using the definitions in (5.2), the expression $\frac{1 - P_k}{P_k} \mathcal{R}_f(k) + \mathcal{R}_s(k)$ reduces to $\mathcal{R}_s(k)$; we define it as *non-penalized multi-rate expected progress (nmep)*.

One may wonder if and how `nmep` is different from NADV (normalized advance) [46], which is defined as \mathbf{a}/\mathbf{c} with the generic cost \mathbf{c} that subsumes link-quality, delay, or anything else. In other words, NADV does not specify the form of cost, and so `nmep` could possibly be called *multi-rate-aware* NADV with $\mathbf{c} = (p_k/\tau(l, b_k))^{-1}$.

According to Theorem 1, MuGR with `nmep` forwards packets as follows: find a candidate n_k with the maximum $\mathcal{R}_s(k)$ and try to forward a packet to that node up to m_k times. If it fails, repeat the procedure with a new candidate node.

5.4.2 Penalized Multi-rate Metric

The penalty for making a neighbor-switch can be defined arbitrarily. Here we consider the neighbor-switch penalty in terms of the estimated number of transmission attempts. The total number of attempts to transmit a packet to node n_k is usually bounded by a maximum try limit, e.g., m_k , and an attempt beyond this limit will trigger the search for a different neighbor. Thus, we modify `nmep` to incorporate this type of penalty as follows. First, we define positive and negative rewards for n_k as:

$$\mathcal{R}(k) \triangleq \begin{cases} \mathcal{R}_s(k) \triangleq a_k p_k - \widehat{c}_k^s, & \text{TX to } n_k \text{ ok;} \\ \mathcal{R}_f(k) \triangleq -\widehat{c}_k^f, & \text{otherwise.} \end{cases} \quad (5.3)$$

where \widehat{c}_k^s and \widehat{c}_k^f are the costs to be incurred as a result of attempting a packet's transmission up to m_k times. Since we are only considering the cost in terms of the number of transmission attempts, the candidate neighbor is "charged" with $\widehat{c}_k^s \triangleq \min\{m_k, 1/p_k\}$ for successful transmission to itself. Otherwise, it is charged with \widehat{c}_k^f , which can be simply m_k after making m_k (predefined) unsuccessful transmission attempts. However, to calculate the penalty for a node that results in consecutive failures, we set \widehat{c}_k^f to $\max\{m_k, \text{tx_failed}_k\}$, where tx_failed_k counts the number of consecutive failures since the last successful transmission to node n_k .

Albeit conceptually intuitive, the reward $\mathcal{R}(k)$ in (5.3) has one problem: a dimension mismatch between its component terms. The first term $a_k p_k$ in $\mathcal{R}_s(k)$, representing the expected progress, is distance, whereas the other terms are simply the number of transmission attempts. To resolve this mismatch, a conversion function between the distance and the number of transmission attempts needs to be established. We define such a conversion function as follows. First, $a_k p_k$ can be interpreted as the geographic advance per transmission attempt since it is $a_k/(1/p_k)$, much like the concept of velocity. Second, every neighbor n_i in the forward region—i.e., nodes closer to the destination than the current node—has this ‘velocity’ of $a_i/(1/p_i)$. Therefore, an average velocity \bar{v} in “distance per transmission attempt” among them is calculated as:

$$\bar{v} = \frac{N}{\sum_N \frac{1}{a_k p_k}} \quad (5.4)$$

where N is the number of nodes in the forward region.

The reward model (5.3) for node n_k can then be transformed to one with distance,

$$\mathcal{R}(k) \triangleq \begin{cases} \mathcal{R}_s(k) \triangleq a_k p_k - \widehat{c}_k^s \bar{v}, & \text{TX to } n_k \text{ ok;} \\ \mathcal{R}_f(k) \triangleq -\widehat{c}_k^f \bar{v}, & \text{otherwise.} \end{cases} \quad (5.5)$$

$\mathcal{R}(k)$ simply states that the reward for a neighbor is just forward or backward progress through that neighbor. Finally, it is made aware of the multi-rate feature:

$$\mathcal{R}(k) \triangleq \begin{cases} \mathcal{R}_s(k) \triangleq (a_k p_k - \widehat{c}_k^s \bar{v})/\tau(l, b_k), & \text{TX to } n_k \text{ ok;} \\ \mathcal{R}_f(k) \triangleq (-\widehat{c}_k^f \bar{v})/\tau(l, b_k), & \text{otherwise.} \end{cases} \quad (5.6)$$

The expression $\frac{1-P_k}{P_k} \mathcal{R}_f(k) + \mathcal{R}_s(k)$ with the definitions in (5.6) is here called the *penalized multi-rate expected progress* (**pmep**). According to Theorem 1, MuGR with **pmep** forwards packets as follows: find a candidate neighbor n_k with the maximum

$\frac{1-P_k}{P_k}\mathcal{R}_f(k) + \mathcal{R}_s(k)$ and try to forward a packet to that node up to m_k times. If it fails, repeat the procedure with a new candidate.

5.4.3 On-line Estimation

In our *ns-2* implementation of MuGR, a module that ranks neighbors based on `nmep` or `pmep` is called `Rank`. Other than `Rank`, the following two modules need to be designed in *ns-2*.

Rate Exposure or Advertisement (`Radv`)

Since how or why the data-rate changes is usually not disclosed by most commercial wireless card vendors, all that MuGR expects from the underlying wireless cards is the current data rate `b`. A module called `Radv` is responsible for learning and advertising it to the node’s neighborhood in the next periodic link-quality probe, as described by `bitrate` in line 6 of Algorithm 7.

Hybrid Link-quality Measurement (`Qlty`)

`Qlty` estimates `dlvrate` for the delivery rate `p`, which combines two distinct link-quality measurement schemes—*unicast-* and *broadcast-based*—in a uniform and consistent way that treats the broadcast-based ETX as a low rate unicast. The algorithm is straightforward except for the failure case. If a transmission attempt fails due to exhaustion of the maximum retry limit, the algorithm treats that failed transmission as a success at the very next transmission attempt. Such an interpretation may not be accurate for the failure case but is sufficient to penalize a link that caused a transmission failure. The variable `tx_failed` counts the number of consecutive failures for each neighbor since the last successful transmission.

To make `Qlty` agile in capturing short-term variations while making it insensitive

Input: an ETX probe received or a transmission result locally observed.
Input: a neighbor `nb` from which an ETX probe is received or to which a transmission is attempted.

Data: a link table \mathcal{L}

```

1  $l \leftarrow \mathcal{L}.lookup(nb);$ 
2 if upon receiving an ETX probe  $e$  then
3    $l.update\_bwdrate();$  /*  $d_r$  */
4    $l.fwdrate \leftarrow e.bwdrate;$  /*  $d_f$  */
5    $dlvrate \leftarrow l.fwdrate \times l.bwdrate;$ 

6 if upon receiving an ACK  $a$  then
7    $l.tx\_failed \leftarrow 0;$ 
8    $dlvrate \leftarrow 1/a.txattempts;$ 

9 if upon receiving a FAILURE event  $f$  then
10   $l.tx\_failed \leftarrow l.tx\_failed + f.txattempts;$ 
11   $dlvrate \leftarrow 1/(f.txattempts + 1);$ 
12   $l.dlvrate \leftarrow \alpha \times dlvrate + (1 - \alpha) \times l.dlvrate;$ 
13  return;

14  $l.dlvrateFilter.put(dlvrate);$ 
15  $l.dlvrate \leftarrow \alpha \times l.dlvrateFilter.get() + (1 - \alpha) \times l.dlvrate;$ 

```

Algorithm 7: Hybrid link-quality measurement

to measurement noises, an instantaneous `dlvrate` undergoes the *median filter* [53] (lines 13 ~ 14 in Algorithm 7) which eliminates a certain span of noises in the input. With the median filter currently set to a 3-span median filter eliminating one peak noise and the smoothing constant α of 0.5, Qlty achieves both agility and stability.

5.5 Evaluation

5.5.1 Wireless Channel

To make the *ns-2* simulation as close to the real-world wireless characteristics as possible, we follow several recommendations made in [42]. One such recommendation is to use the shadowing propagation model to reflect wireless link characteristics such as link asymmetry and time-varying link-quality. The *ns-2* shadowing model and other configurations are set up as in Table 5.1.

Table 5.1: Configurations for 802.11b with the shadowing model.

Path loss exponent		4	
Shadowing deviation		9,7,5,4	
Reference distance		10	
Frequency	2.4 GHz	Transmit power	15 dBm
CPTthresh	10	CSTthresh	-105 dBm
RXThresh (11 Mbps) [†]		-79.84 dBm, (95% at 50m)	
RXThresh (5.5 Mbps) [†]		-85.68 dBm, (95% at 70m)	
RXThresh (2 Mbps) [†]		-90.05 dBm, (95% at 90m)	
RXThresh (1 Mbps) [†]		-94.31 dBm, (95% at 115m)	
Data Rate		11 Mbps (w/ or w/o auto rate)	
Basic Rate		2 Mbps	
MAC headers		(24,14,14,28) bytes @ Basic Rate	
PLCP length		192 bits @ 1 Mbps	
SIFS	10us	DIFS	50us

[†] These thresholds are configured for the path-loss exponent 4 and the shadowing deviation 4 according to the wireless card specifications [58]. They remain unchanged for the rest of simulation.

The plots from a quick dry run of the configured wireless channel model are shown in Figure 5.2. They demonstrate how well our simulation setting reflects the characteristics of a wireless channel as found in many real-world measurements [1, 6, 45, 68, 72].³ For the experiments in Figure 5.2, packets were intentionally broadcast at several different rates than at the default 2Mbps, to illustrate how the disparities in both the coverage and the packet reception probability are developed under the configured receiving thresholds.

A total of 12 scenarios, each lasting 500 seconds, are considered using the basic wireless channel setting specified in Table 5.1. Each scenario, denoted by the labels (S1, S2, ..., S12) in Figure 5.3, will be rolled out one-by-one over the entire simulation of length 6,000 seconds. Changes are made in some or all of the wireless channel

³Data in Figure 5.2 were the average results obtained from a single run of the *ns-2* simulation with the default path-loss exponent 4 and the shadowing deviation 4. Nodes were arranged on a line where one node at the origin keeps broadcasting packets of the MAC-level payload 1500 bytes and the other 30 nodes, evenly spaced 10m apart on the line, measured the received throughput and the delivery rate.

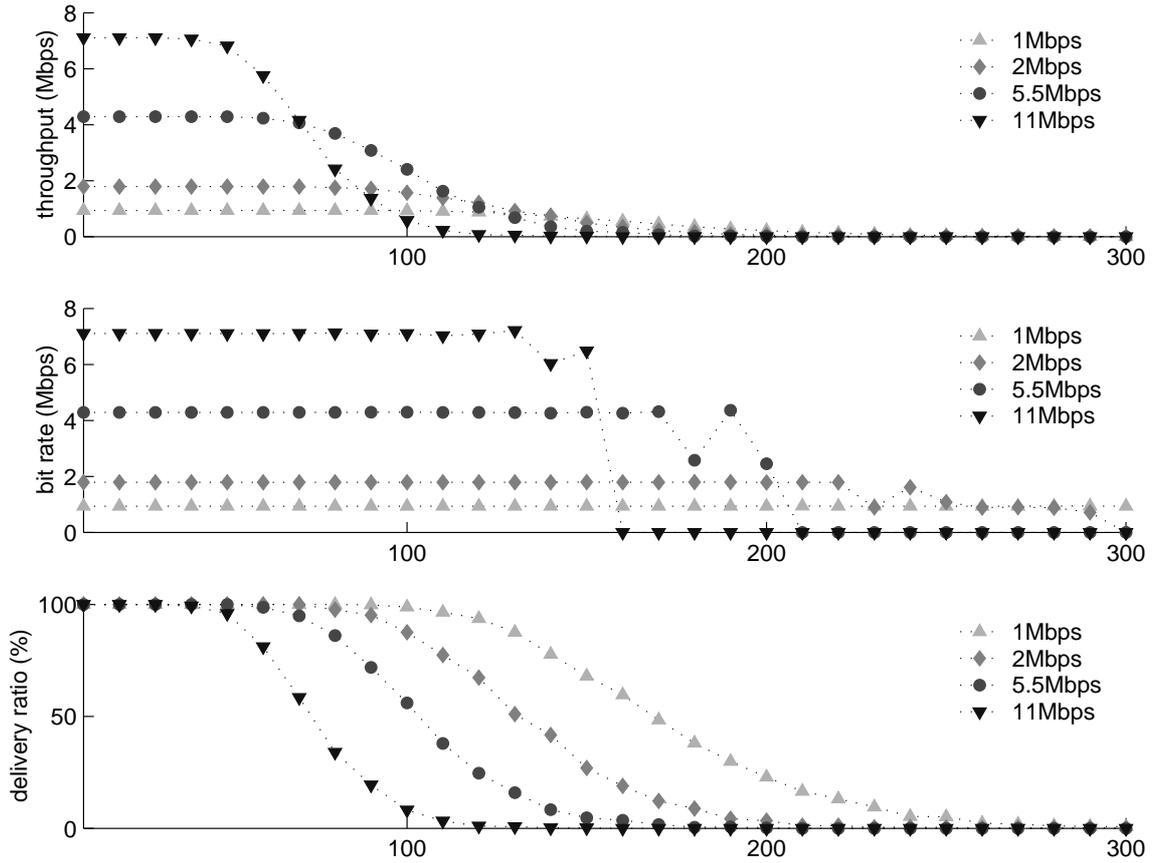


Figure 5.2: Tradeoffs of throughput vs. distance for several bit-rate schemes in the simulated IEEE 802.11b. Labels in the x-axis represent the distance in meters. The plot for throughput vs. distance (1st row) is decomposed into two sub-plots; bit-rate vs. distance (2nd row) and delivery ratio vs. distance (3rd row).

parameters, the traffic and node parameters, and the random packet-loss parameter at the physical layer. The node constraint “–” means that any change will be global across all the nodes, whereas two node constraints G0 and G1 mean that a change will be applied to those nodes satisfying G0 or G1 constraint.

Selection of nodes subject to both constraints is arbitrary. For example, the node constraint G0 that changes the shadowing deviation parameter applies to those nodes whose IDs equal 1 modulo 3. Similarly, the node constraint G1, which drops the received packets randomly at the physical layer with a probability of 0.3, will

	S1	S2	S3	S4	S5	S6
path loss	4	4	4	4	4	4
deviation	9	9	9	7	7	7
flow set	1,	1,2	1,2,3	1,2,3,4	1	1,2
node	-	-	-	G0+G1	-	-
	S7	S8	S9	S10	S11	S12
path loss	4	4	4	4	4	4
deviation	7	5	5	5	5	4
flow set	1,2,3	1,2,3,4	1	1,2	1,2,3	1,2,3,4
node	-	G0+G1	-	-	-	G0

Figure 5.3: “Panoramic” evaluations by staging out different simulation configurations during the entire evaluation.

be applied to the nodes whose IDs are 54, 20, 35, 24, 11, 51, 42, 41, 45, and 26, regardless of the type of topologies used. Note that the node constraint G1 is not applied in the last S12.

Multi-hop flows will be generated using CBR (constant bit rate) over UDP at an average interval of 4 seconds with an average application-level payload of 1500 bytes, each of $1500 + 750 \times U(-1, 1)$ bytes. This packet size distribution implies that the RTS/CTS exchange will not occur in our simulations since the `RTSThreshold` value in the 802.11 standard is set to 2047 bytes. The packet-generation interval is not constant; it is also changed during the stages S11 and S12 such that the interval for S11 is 1.33s and 1s for S12.

Finally, all the simulation results presented here are averages of 10 independent runs of the entire evaluation with a different random seed for each run. Results within a single run are accumulated over stages within that run; the performance results at one stage are accumulated on to the next stage. The purpose of all of these seemingly complex evaluations described so far is to create a dynamic environment that introduces not only a long- or short-term change, but also a global or local

change.

5.5.2 Auto Rate Algorithms

Not only to make the *ns-2* simulation realistic but also to create different bandwidths among neighbors depending on the wireless channel condition, auto rate algorithms are used. Since the auto rate algorithm in a commercial wireless card is usually not disclosed, we use the following algorithms available on a code base: AMRR [45], Onoe [50], SampleRate [6], RBAR [28]. The first three algorithms are implemented in a Linux driver, whereas the last is available only in the *ns-2* simulation. All algorithms are ported into our *ns-2* simulation.

5.5.3 Topologies and Flows

Two types of topologies are used: the line topology and the unplanned non-uniform topology. The line topology is used to assess the effectiveness of individual design components of MuGR by isolating the impact of each component on routing performance. On the other hand, the unplanned non-uniform topology is used to conduct a realistic performance study since many deployed wireless networks are irregular in their node distribution due to terrestrial obstacles or building structures.

With the line topology, 61 nodes numbered from 0 to 60 are located at $(10m \times k, 0)$ where k is a node's ID. Multiple flows on the line topology are as follows: flow set 1 = $\{0 \rightarrow 60\}$, flow set 2 = $\{50 \rightarrow 10\}$, flow set 3 = $\{20 \rightarrow 40\}$, and flow set 4 = $\{35 \rightarrow 25\}$.

With the unplanned topology, an MIT indoor wireless sensor network testbed called MistLab [51] is reproduced in our *ns-2* simulation. The simulated MistLab, simply called here **nsMistLab**, has 61 802.11b nodes spread non-uniformly over a square field as shown in Figure 5.4. Multiple flows on **nsMistLab** are defined as follows:

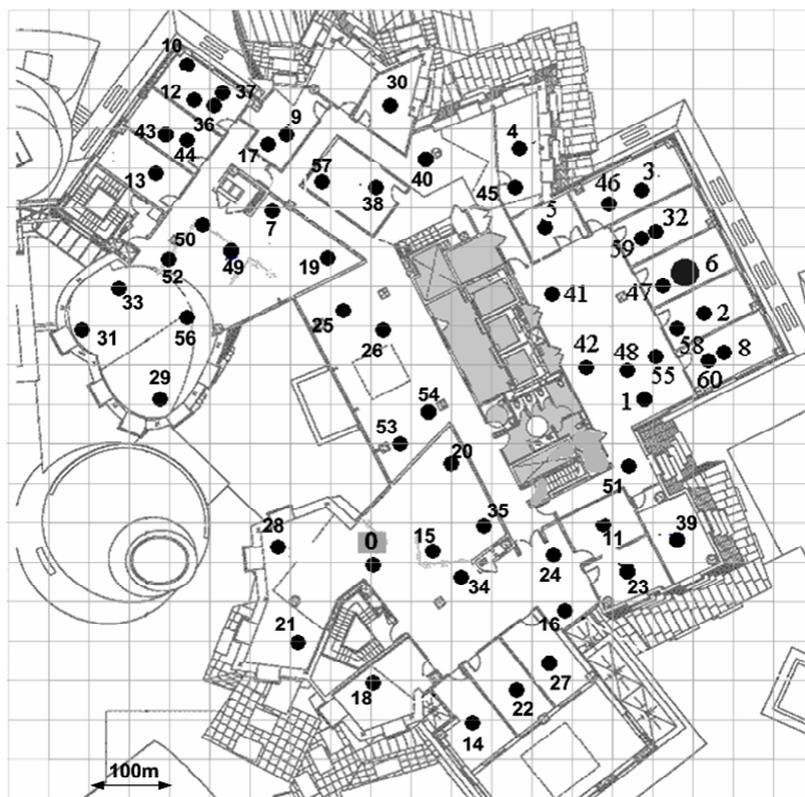


Figure 5.4: MistLab [51], a non-uniform real-world wireless sensor network testbed with 61 nodes over a field of 16,000 square feet, is reproduced in our *ns-2* simulation as an *nsMistLab* 802.11b wireless network over a field of $1\text{km} \times 1\text{km}$.

flow set 1 = {14 → 6}; flow set 2 = {18 → 10}; flow set 3 = {32 → 43, 9 → 21}; and flow set 4 = {23 → 4, 29 → 30, 8 → 13}. For certain evaluations, these destinations are all changed to node 6 to create a typical many-to-one flow in wireless sensor or mesh networks.

5.5.4 Performance Metrics and Questions

In addition to the application-level throughput, we also evaluated the following performance metrics:

- Overhead: the ratio of the total length of *bits generated* in the network to the total length of *data bits received* by receivers. The bits generated in the network count all the bits either in beacons being broadcast or in data being (re)transmitted for its hop-by-hop movement.
- Per-hop overhead: the ratio of the total *number of packets generated* in the network to the total number of *data packets received* by receivers. The per-hop overhead is closely related to the number of per-hop transmission attempts.
- End-to-end (e2e) delay: the average packet-delivery time between senders and receivers, measured in simulation seconds.
- End-to-end (e2e) hops: the average number of hops between senders and receivers.

The main questions we addressed are

- Does MuGR significantly improve the throughput in multi-rate multi-hop wireless networks?
- Which of Radv, Qlty, and Rank improves the throughput most?

protocol	meaning
NADV	use the cost term $c = 1/p$, where $p = \text{fwrate} (d_f) \times \text{bwrate} (d_r)$
NADV, fixed	no rate adaptation is used. Data is always sent at 11Mbps.
NADV+<Auto>	a data rate is set on the sender side by <Auto> algorithm
NADV+Radv	a data rate is automatically changed to the rate a neighbor recommended
NADV+Qty	use the cost term $c = 1/p$, where $p = \text{dlvrate}$ in the unified scheme
MuGR, nmep	Radv+NADV with the cost term $c = t/p$, where $t = \text{delay}$, $p = \text{dlvrate}$

Figure 5.5: Performance evaluations in isolation. See Algorithm 7 for `fwrate`, `bwrate`, and `dlvrate`.

- Is the defined penalty model effectively controlling MuGR?

Throughout the evaluations, MuGR will often be compared against protocols that are augmented on top of NADV.⁴ Recall that ETX [12] is calculated by measuring the forward-delivery ratio (d_f) and the backward-delivery ratio (d_r) using the broadcast-based link probe such that $\text{ETX} = (d_f d_r)^{-1}$. Thus, the base NADV uses the link metric a/c , where $c = \text{ETX}$. Figure 5.5 summarizes the meaning of each addition to the base NADV. Finally, MuGR uses a small retry limit ($R = 3$) to aggressively switch to another neighbor if the current transmission fails, whereas NADV uses the default retry limit ($R = 7$) or a small retry limit ($R = 3$) depending on the evaluation scenario.

5.5.5 Main Results on Line Topology

Figure 5.6(a) shows that MuGR is by far more effective in improving the throughput than NADV with the fixed and auto bit-rate schemes we considered. Compared to NADV with fixed 11 Mbps, MuGR improves the throughput consistently across all of the simulation stages, by 16–54% depending on the channel conditions simulated. Even if NADV is assumed to run atop Radv, MuGR still achieves a throughput gain

⁴The link metric is also used as geographic routing with it.

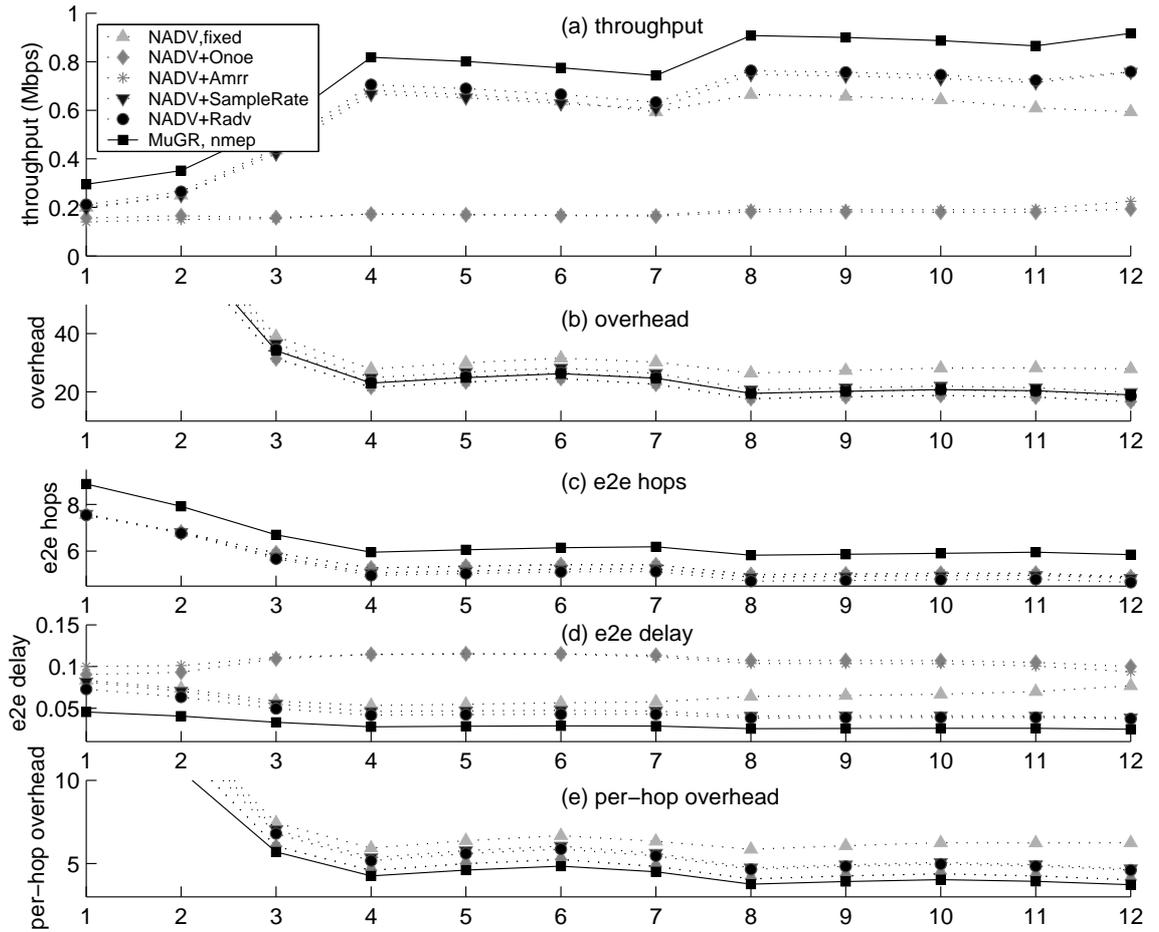


Figure 5.6: Routing performance on the line topology with fixed 11 Mbps, which is the default data rate for 802.11b, and several auto rate algorithms. Labels in the x-axis represent the stage number in Figure 5.3.

of 16–39%.

MuGR outperforms others even if the wireless channel condition gets worse. This is because (i) it is multi-rate-aware, (ii) it uses more accurate link-quality measurements in Algorithm 7, (iii) it quickly switches to another neighbor after attempting fewer transmissions, and (iv) it uses wireless resources efficiently and judiciously; this is explained below. Switching neighbors quickly does not always pay off, as will be shown in the later evaluation with the unplanned topology.

Other performance behaviors of MuGR are also observed in Figures 5.6(c) and

(d), showing why **MuGR** improves the throughput significantly. The e2e delay of **MuGR** (Figure 5.6(c)) turned out to be minimal in spite of a relatively small increase in the e2e hop count (Figure 5.6(d)). This implies that **MuGR** wastes less per-hop transmission time than the other schemes. This is also observed in Figure 5.6(e), which measures the per-hop overhead in terms of the number of attempts until a successful transmission. In terms of the total overhead in Figure 5.6(b), **MuGR** performs better than, or as well as, the other algorithms.

The best way to answer the second question is to examine the effectiveness of a component by isolating it from the other components. Thus, we opt to augment **NADV** with each component of **MuGR** one-by-one.

The first experiment with **NADV** is the addition of the best auto rate selection algorithm. Several auto rate selection algorithms are evaluated with **NADV** and **Radv** turned out to improve the throughput most; see Figure 5.6(a). Thus, our first experiment is to simply augment **NADV** with **Radv** and the result of this augmentation is the improvement of throughput by 5–27%; **NADV+Radv,R=7** vs. **NADV,fixed,R=7** in Figure 5.7. However, since **Radv** can be a common foundation for other routing protocols, this result would rather emphasize the importance of having a proper rate-selection algorithm for each neighbor in multi-rate or multi-radio wireless networks.

Our second experiment is to provide a slight change in **NADV+Radv,R=7** by using the smaller retry limit 3, i.e., **NADV+Radv,R=3** in Figure 5.7. The standard 802.11 allows the MAC-level attributes such as `ShortRetryLimit` or `LongRetryLimit` to be changed on a per-packet basis through the management interface. We use this feature. A slight change of the retry limit increases the opportunistic use of neighbors, although it may cause route oscillations. The result of this experiment indicates that switching neighbors frequently improves the throughput. However,

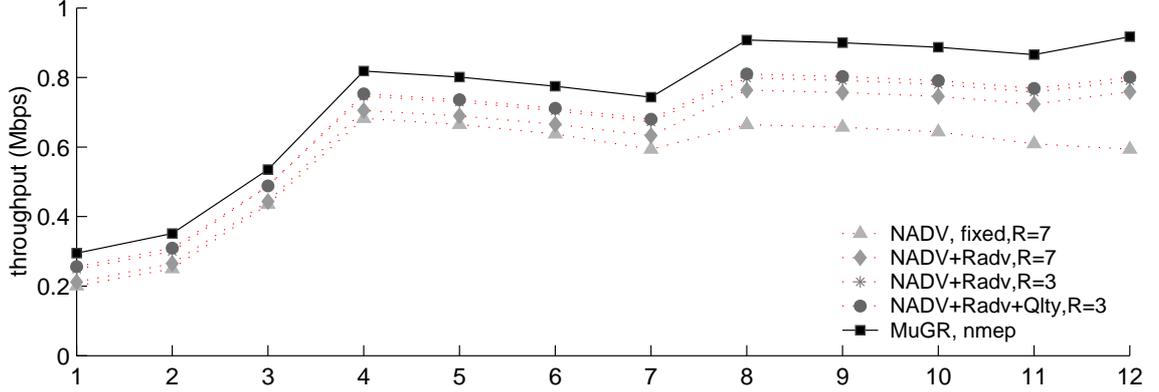


Figure 5.7: Throughput improvement on the line topology by each component of MuGR.

this conclusion on the line topology will later be negated in the same evaluation with nsMistLab.

Our third experiment is to augment NADV+Radv, R=3 with Qlty. By using Qlty, p_k is updated as specified in Algorithm 7. The result of this experiment is an additional slight (1–2%) increase of throughput over the second experiment. However, the same experiment with nsMistLab shows an additional throughput gain of 23–47%, which will be described later.

Our last experiment is to augment NADV+Radv+Qlty, R=3 with Rank. This is almost MuGR under nmep except for no consideration of the transmission delay. The result of this experiment is an additional throughput gain of 8–13% over the third experiment, or of 16–40% over NADV+Radv, R=7, or of 19–54% over NADV, R=7.

5.5.6 Evaluation with Unplanned Non-uniform Topology

Similar conclusions (drawn from the line topology) still hold for both the throughput and the e2e delay in the unplanned non-uniform topology. As seen in Figure 5.8, MuGR consistently achieves 14–26% more throughput than the best NADV, where NADV+Radv, R=7. Recall that NADV+Radv+Qlty is not the original NADV and

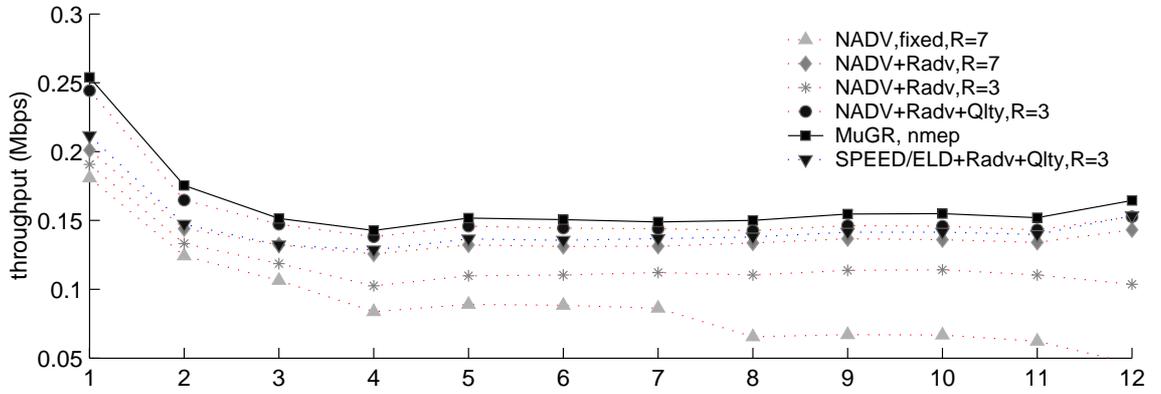


Figure 5.8: Throughput improvement on nsMistLab by each component of MuGR. SPEED/ELD uses a link metric that is updated solely on data being unicast.

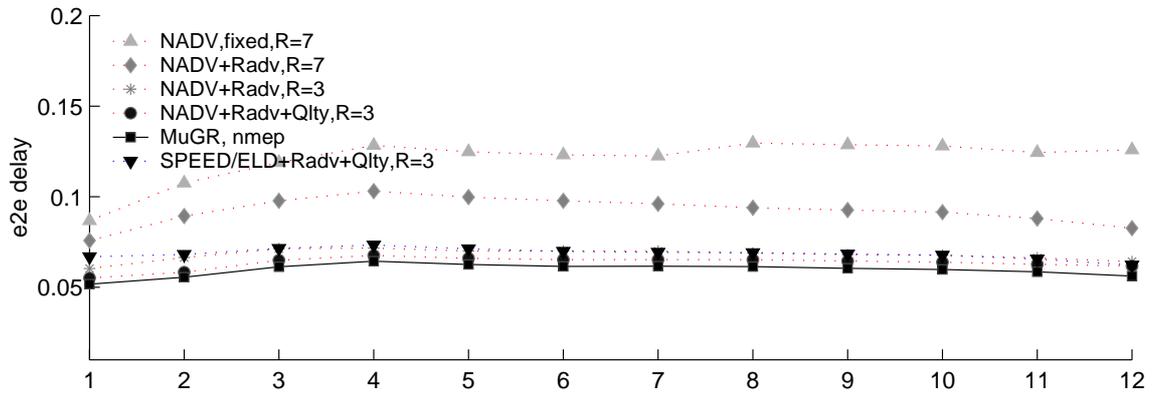


Figure 5.9: E2e delay improvement on nsMistLab by each component of MuGR. SPEED/ELD uses a link metric that is updated solely on data being unicast.

is just augmented to investigate the performance improvement of MuGR on a per-component basis.

In Figure 5.8, the performance of using SPEED[27]/ELD[71] is also presented. SPEED and ELD are very similar to each other in that they use a link metric purely based on data being unicast. They also use the same metric defined as the advanced distance for per-hop delay, and switch the next hop probabilistically. Since the probability function used in SPEED is clearer than the one used in ELD, we use the probability described in SPEED. Our evaluation result shows that geographic routing with the unified link metric (NADV+Radv+Qlty,R=3) yields a better performance than the one with the unicast-based link metric (SPEED/ELD+Radv+Qlty,R=3). However, MuGR still outperforms the others.

We obtained similar results for e2e delay, as shown in Figure 5.9; MuGR achieves the lowest. Interesting results regarding the retry limit in NADV are observed; in NADV the maximum throughput is achieved when the retry limit is 7, whereas the minimum e2e delay is achieved with the retry limit 3. One probable cause for this is the neighbor-deletion policy in NADV, inherited from the original GPSR that deletes a neighbor upon a single link failure. The use of the smaller retry limit makes NADV lose neighbors quickly, thus reducing the number of available paths toward the destination. MuGR avoids this, thanks to its probabilistic neighbor deletion policy; it deletes a neighbor whose `dlvrate` in Algorithm 7 is less than 0.125 or $1 / (802.11 \text{ ShortRetryLimit}(7)+1)$.

5.5.7 Evaluation of Penalty Model: `nmep` vs. `pmep`

So far, we have presented the results for MuGR with `nmep` only. Now, we demonstrate the effectiveness of MuGR with `pmep` in Figures 5.10–5.13. The way we defined

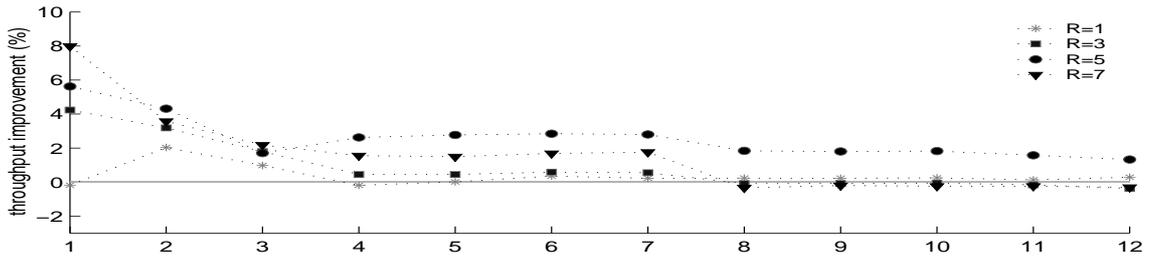


Figure 5.10: Relative throughput improvement by pmep (over nmep under the line topology).

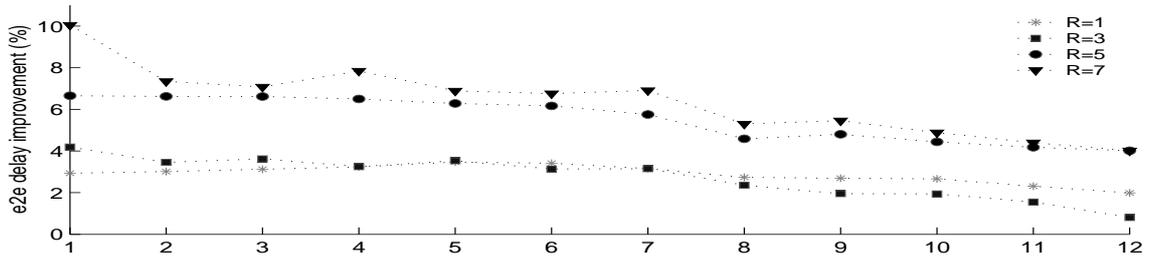


Figure 5.11: Relative e2e delay improvement by pmep (over nmep under the line topology).

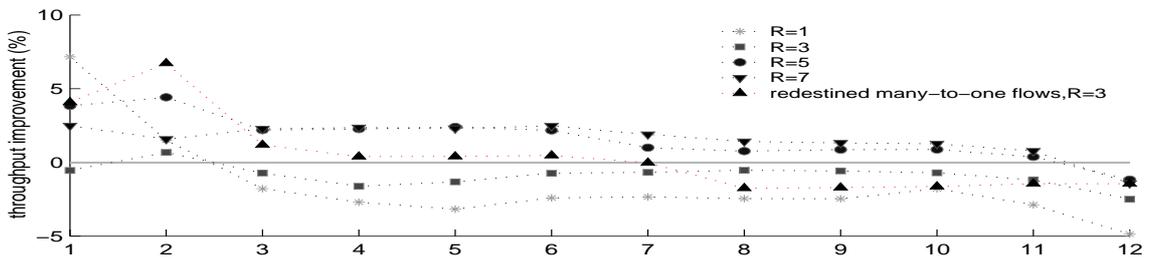


Figure 5.12: Relative throughput improvement by pmep (over nmep under nsMist-Lab).

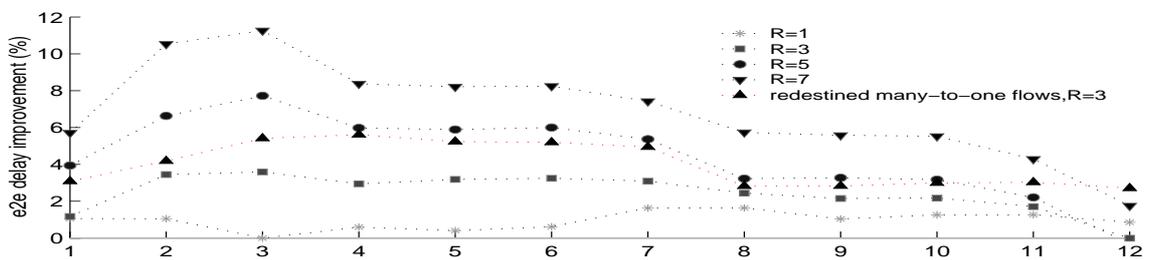


Figure 5.13: Relative e2e delay improvement by pmep (over nmep under nsMistLab).

the penalty model in the previous section discourages **MuGR** from selecting the next hop that is likely to incur a large number of transmission attempts. A large number of per-hop transmission attempts will result in a long per-hop delay. Therefore, **MuGR** with the **pmep** metric is said to be effective if it reduces the e2e delay.

In summary, **MuGR** with **pmep** achieves our goals with the throughput comparable to that of **MuGR** with **nmep** metric; the e2e delays in Figures 5.11 and 5.13 consistently show that **MuGR** with **pmep** has a lower e2e delay regardless of the retry limits used. This indicates that **pmep** with the cost definition in (5.6) gets **MuGR** to successfully avoid those links that require a large number of transmission attempts.

However, we don't know if we can improve further by defining **pmep** with a different cost. Note that the re-destined many-to-one flows in Figures 5.12 and 5.13 are introduced by changing the destinations of all the flows defined in our **nsMistLab** topology to a single node 6, which could represent a sink or gateway.

5.6 Conclusion

In this chapter we presented a novel packet-forwarding strategy and derived two link metrics **nmep** and **pmep** for geographic routing in multi-rate multi-hop wireless networks. An extensive simulation study showed that **nmep** improves the throughput by 16–39% for the given simulation settings as compared to the state-of-the-art link metrics in the literature. We also demonstrated that geographic routing with **pmep** reduces the e2e delay more than routing with **nmep**, while achieving a comparable throughput.

CHAPTER VI

CONCLUSIONS AND FUTURE DIRECTIONS

6.1 Conclusions

This research began by recognizing that both accidental switching against unexpected link failures and opportunistic switching for better performance can be detrimental to the overall network capacity or throughput. In particular, from the standpoint of routing, any link switching can trigger expensive route reconfiguration and rediscovery procedures. Frequent and high volume of route maintenance traffic wastes network capacity and offsets any expected benefits from opportunistic switching below the network layer.

The major contribution of this research is the development of techniques that make routing immune to the underlying link switches insofar as possible. An important foundation for the techniques developed is the elimination of the explicit route state by building a stateless, fine-grained, and continuous gradient-field based on link-quality. Routing over the gradient-field can be translated to standard geographic routing. The greatest advantages of routing packets on the gradient-field are then not only the achievement of high resilience against link failures, but also the significant improvement of performance by dynamically selecting the best next hop to make the most efficient progress toward the destination.

We first proposed two stateless routing protocols, GDSR and GASR, with respect to a single landmark or the Internet gateway in wireless mesh networks. They are complementary in that GDSR offers resilient and efficient unicast in the gradient-descending direction, whereas GASR does so in the gradient-ascending direction. Any many-to-one applications that require lightweight, resilient, stateless, and bidirectional routing can benefit much from both GDSR and GASR.

GHVR extended the idea of GDSR and GASR to build a generalized one-to-one routing protocol, and achieved it by using multiple landmarks. Landmarks can be wired Internet gateways or dedicated beacon nodes. By constructing individual and independent gradient-fields with respect to each of the selected landmarks, nodes in the network can be provided with multiple gradient-based coordinates. Then, they are used to trick standard geographic routing into believing that they are the actual geographic locations in the multi-dimensional space.

Finally, we presented a generic greedy packet forwarding strategy over physical or virtual coordinate space when nodes are equipped with multi-channels, multi-radios. The goal of the strategy is to find a neighbor-switch order among neighbors while keeping the cost caused by the sequence of neighbor-switches as low as possible. In designing MuGR, we first developed a generic reward and penalty model on an abstract wireless network model. From the generic model, we derived two link metrics for multi-channel, multi-radio wireless networks.

6.2 Future Directions

Although the coordinate-based routing protocols developed in this research are heavily verified through *ns-2* simulations, we will need to continue study of the efficacy, strengths and weaknesses of coordinate-based routing, and to widen its

applicability.

6.2.1 Performance Comparisons of Coordinate Routing vs. Topology-based Routing in Wireless Mesh Networks

Currently, IEEE 802.11s Working Group is focused on considering conventional proactive topology-based routing protocols such as distance-vector routing or link-state routing for wireless mesh networks. However, there is no study yet suggesting which type of routing protocol is best for multi-channel, multi-radio wireless mesh networks with stationary and mobile end-users. Furthermore, there is no comparative study of conventional topology-based routing protocols vs. (the proposed) coordinate routing protocols, which would entail answering following questions:

- How do topology-based routing protocols scale with the rate of link-layer failures or switches?
- How much overhead is caused by those switches under each of the protocols investigated?
- How do these protocols perform in various application workloads and scenarios?

Such study could reveal the advantages and disadvantages of each of the protocols investigated and the impact of the decoupling of route state from link-layer failures or switches on network capacity in large-scale wireless mesh networks.

6.2.2 Reward-Maximizing Opportunistic Handover in Wireless Mesh Networks with Mobile Users

To widen the applicability of the proposed routing protocol, we need to address the way in which mobile users access the network. Given multiple access points in densely populated wireless mesh networks, users have to decide which access point to use. This decision becomes more complicated when users are moving, and is

equivalent to determining whether it is advantageous to use the current wireless link or not. Thus, the decision model should be expressed in terms of the reward and penalty associated with the selection of a wireless link to a particular access point. When developing the reward and penalty models, the following design factors need to be incorporated.

- The location of access points in geographic or virtual coordinate space.
- The bandwidth and link-quality offered by each of access points.
- The cost of releasing or establishing a secure association with an access point.
- The bandwidth and duration of an application session.
- The location of a server that an application contacts.
- The speed and intended trajectory of mobile users.

After defining appropriate reward and penalty models, the problem of deciding whether or not to switch to a new access point can be instantiated from the opportunistic switching forwarding problem studied here. Investigation is need to understand how an instantiated decision model with related reward or penalty models affects routing performance or improves an application-level quality of service. Finally, study is also needed of cases where multiple access points from heterogeneous wireless networks become available, offering different bandwidth, error rate, and coverage.

APPENDIX

APPENDIX A

Proof of Theorem 1

Given the positive and negative reward definitions $\mathcal{R}_s(k)$ and $\mathcal{R}_f(k)$ for node n_k , suppose a sequence $\underline{s} \triangleq (n_1, n_2, \dots, n_{k-1}, i = n_k, j = n_{k+1}, n_{k+2}, \dots, n_N)$. Then,

$$E[\mathcal{R}(\underline{s})] = \Pr\{\text{send}\}E[\mathcal{R}(\underline{s})|\text{send}] + \Pr\{\text{fail}\}E[\mathcal{R}(\underline{s})|\text{fail}]$$

$$\Pr\{\text{fail}\} = \prod_{o=1}^N (1 - P_{n_o}), \quad \Pr\{\text{send}\} = 1 - \Pr\{\text{fail}\}$$

where $P_{n_o} \triangleq 1 - (1 - p_{n_o})^{m_{n_o}}$. Moreover,

$$E[\mathcal{R}(\underline{s})|\text{fail}] = \sum_{k=1}^N \mathcal{R}_f(n_k)$$

and

$$\begin{aligned} E[\mathcal{R}(\underline{s})] &= \left(1 - \prod_{o=1}^N (1 - p_{n_o})\right) E[\mathcal{R}(\underline{s})|\text{send}] \\ &+ \prod_{o=1}^N (1 - p_{n_o}) \sum_{o=1}^N \mathcal{R}_f(n_o) \end{aligned}$$

Thus, the following lemma holds:

Lemma 1 *Given two ordered sequences \underline{s} and \underline{t} of neighbors, $E[\mathcal{R}(\underline{s})] \geq E[\mathcal{R}(\underline{t})] \Leftrightarrow E[\mathcal{R}(\underline{s})|\text{send}] \geq E[\mathcal{R}(\underline{t})|\text{send}]$.*

On the other hand,

$$E[\mathcal{R}(\underline{s})|\text{send}] = \sum_{o=1}^N E[\mathcal{R}(\underline{s})|I_{\underline{s}}(o) = 1]\Pr\{I_{\underline{s}}(o) = 1\}$$

where $I_{\underline{s}}(o)$ is an indicator function, i.e., $I_{\underline{s}}(o) = 1$ when TX succeeds at the o^{th} neighbor n_o in the sequence \underline{s} ; 0 otherwise.

Then,

$$\Pr\{I_{\underline{s}}(k) = 1\} = \prod_{o=1}^{k-1} (1 - P_{n_o})P_{n_k}$$

where $P_{n_i} \triangleq 1 - (1 - p_{n_i})^{m_{n_i}}$.

Moreover,

$$\begin{aligned} E[\mathcal{R}(\underline{s})|I_{\underline{s}}(k) = 1] &= \sum_{o=1}^{k-1} \mathcal{R}_f(n_o) + E[\mathcal{R}(n_k)|\text{send}, M = m_k] \\ &= \sum_{o=1}^{k-1} \mathcal{R}_f(n_o) + \mathcal{R}_s(n_k) \end{aligned}$$

where M represents the maximum try limit.

Therefore,

$$\begin{aligned} E[\mathcal{R}(\underline{s})|\text{send}] &= \sum_{o=1}^{k-1} E[\mathcal{R}(\underline{s})|I_{\underline{s}}(o) = 1]\Pr\{I_{\underline{s}}(o) = 1\} + \\ &\quad \left(\sum_{o=1}^{k-1} \mathcal{R}_f(n_o) + \mathcal{R}_s(i) \right) \prod_{o=1}^{k-1} (1 - P_{n_o})P_i + \\ &\quad \left(\sum_{o=1}^{k-1} \mathcal{R}_f(n_o) + \mathcal{R}_f(i) + \mathcal{R}_s(j) \right) \prod_{o=1}^{k-1} (1 - P_{n_o})(1 - P_i)P_j + \\ &\quad \sum_{o=k+2}^N E[\mathcal{R}(\underline{s})|I_{\underline{s}}(o) = 1]\Pr\{I_{\underline{s}}(o) = 1\} \end{aligned}$$

Similarly,

$$\begin{aligned}
E[\mathcal{R}(\underline{t})|\text{send}] = & \\
& \sum_{o=1}^{k-1} E[\mathcal{R}(\underline{t})|I_{\underline{t}}(o) = 1]\Pr\{I_{\underline{t}}(o) = 1\} + \\
& \left(\sum_{o=1}^{k-1} \mathcal{R}_f(n_o) + \mathcal{R}_s(j) \right) \prod_{o=1}^{k-1} (1 - P_{n_o})P_j + \\
& \left(\sum_{o=1}^{k-1} \mathcal{R}_f(n_o) + \mathcal{R}_f(j) + \mathcal{R}_s(i) \right) \prod_{o=1}^{k-1} (1 - P_{n_o})(1 - P_j)P_i + \\
& \sum_{o=k+2}^N E[\mathcal{R}(\underline{t})|I_{\underline{t}}(o) = 1]\Pr\{I_{\underline{t}}(o) = 1\}
\end{aligned}$$

where $\underline{t} \triangleq (n_1, n_2, \dots, n_{k-1}, j, i, n_{k+2}, \dots, n_N)$.

Thus, the following lemma holds:

Lemma 2

$$\begin{aligned}
E[\mathcal{R}(\underline{s})|\text{send}] & \geq E[\mathcal{R}(\underline{t})|\text{send}] \\
& \Leftrightarrow \mathcal{R}_f(i)P_j - \mathcal{R}_f(i)P_iP_j - \mathcal{R}_s(j)P_iP_j \\
& \geq \mathcal{R}_f(j)P_i - \mathcal{R}_f(j)P_iP_j - \mathcal{R}_s(i)P_iP_j \\
& \Leftrightarrow \mathcal{R}_f(i)P_j - \mathcal{R}_f(i)P_iP_j + \mathcal{R}_s(i)P_iP_j \\
& \geq \mathcal{R}_f(j)P_i - \mathcal{R}_f(j)P_iP_j + \mathcal{R}_s(j)P_iP_j \\
& \Leftrightarrow \frac{1 - P_i}{P_i} \mathcal{R}_f(i) + \mathcal{R}_s(i) \\
& \geq \frac{1 - P_j}{P_j} \mathcal{R}_f(j) + \mathcal{R}_s(j)
\end{aligned}$$

By Lemmas 1 and 2, Theorem 1 follows.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level measurements from an 802.11b mesh network. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 121–132, New York, NY, USA, 2004. ACM Press.
- [2] Paramvir Bahl, Ranveer Chandra, and John Dunagan. Ssch: slotted seeded channel hopping for capacity improvement in iee 802.11 ad-hoc wireless networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 216–230, New York, NY, USA, 2004. ACM Press.
- [3] Christopher L. Barrett, Stephan J. Eidenbenz, Lukas Kroc, Madhav Marathe, and James P. Smith. Parametric probabilistic sensor network routing. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pages 122–131. ACM Press, 2003.
- [4] Anindya Basu, Alvin Lin, and Sharad Ramanathan. Routing using potentials: a dynamic traffic-aware routing algorithm. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 37–48, New York, NY, USA, 2003. ACM Press.
- [5] John Bicket. Bit-rate selection in wireless networks, master's thesis. pdos.csail.mit.edu/papers/jbicket-ms.pdf, 2005.
- [6] John Bicket, Daniel Aguayo, Sanjit Biswas, and Robert Morris. Architecture and evaluation of an unplanned 802.11b mesh network. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 31–42, New York, NY, USA, 2005. ACM Press.
- [7] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, 1970.
- [8] Prosenjit Bose, Pat Morin, Ivan Stojmenović, and Jorge Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wirel. Netw.*, 7(6):609–616, 2001.

- [9] Prosenjit Bose, Pat Morin, Ivan Stojmenovic, and Jorge Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *DIALM '99: Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*, pages 48–55. ACM Press, 1999.
- [10] Qing Cao and Tarek Abdelzaher. A scalable logical coordinates framework for routing in wireless sensor networks. In *the IEEE RTSS*, 2004.
- [11] Robert Castaeda and Samir R. Das. Query localization techniques for on-demand routing protocols in ad hoc networks. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 186–194. ACM Press, 1999.
- [12] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 134–146. ACM Press, 2003.
- [13] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. Vivaldi: a decentralized network coordinate system. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 15–26, New York, NY, USA, 2004. ACM Press.
- [14] Swades De, Chunming Qiao, and Hongyi Wu. Meshed multipath routing: An efficient strategy in sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference, 2003*.
- [15] B. Deb, S. Bhatnagar, and B. Nath. Reinform: reliable information forwarding using multiple paths in sensor networks. In *Proceedings. 28th Annual IEEE International Conference on Local Computer Networks (LCN), Vol., Iss., 20-24 Oct. 2003*.
- [16] Benoit Donnet, Bruno Baynat, and Timur Friedman. Retouched bloom filters: Allowing networked applications to trade off selected false positives against false negatives. In *CoNEXT*, 2006.
- [17] Richard Draves, Jitendra Padhye, and Brian Zill. Comparison of routing metrics for static multi-hop wireless networks. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 133–144, New York, NY, USA, 2004. ACM Press.
- [18] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 114–128, New York, NY, USA, 2004. ACM Press.

- [19] Eiman Elnahrawy, Xiaoyan Li, and Richard P. Martin. The limits of localization using signal strength: A comparative study. In *Proceedings of The First IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON 2004)*.
- [20] Li Fan, Pei Cao, Jussara Almeida, and Andrei Z. Broder. Summary cache: a scalable wide-area web cache sharing protocol. *IEEE/ACM Trans. Netw.*, 8(3):281–293, 2000.
- [21] Qing Fang, Jie Gao, Leonidas J. Guibas, Vin de Silva, and Li Zhang. Glider: gradient landmark-based distributed routing for sensor networks. In *IEEE INFOCOM*, 2005.
- [22] Roland Flury and Roger Wattenhofer. MLS: An Efficient Location Service for Mobile Ad Hoc Networks. In *MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, May 2006.
- [23] Rodrigo Fonseca, Sylvia Ratnasamy, Jerry Zhu, Chen Tien Ee, David Culler, Scott Shenker, and Ion Stoica. Beacon vector routing: Scalable point-to-point in wireless sensornets. In *NSDI*, 2005.
- [24] H. Füssler, J. Widemer, M. Käsemann, M. Mauve, and H. Harterstein. Contention-based forwarding for mobile ad hoc networks. In *Journal of Ad Hoc Networks*, pages 351–369, 2003.
- [25] Deepak Ganesan, Ramesh Govindan, Scott Shenker, and Deborah Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(4):11–25, 2001.
- [26] Z.J. Hass, J.Y. Halpern, and L. Li. Gossip-based ad hoc routing. In *IEEE Infocom*, 2002.
- [27] Tian He, John A. Stankovic, Chenyang Lu, and Tarek F. Abdelzaher. A spatiotemporal communication protocol for wireless sensor networks. *IEEE Trans. on Parallel and Distributed Systems*, 16(10), 2005.
- [28] Gavin Holland, Nitin Vaidya, and Paramvir Bahl. A rate-adaptive mac protocol for multi-hop wireless networks. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 236–251, New York, NY, USA, 2001. ACM Press.
- [29] Jonathan W. Hui and David Culler. The dynamic behavior of a data dissemination protocol for network programming at scale. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 81–94. ACM Press, 2004.
- [30] IETF. Optimized link state routing protocol (olsr). <http://tools.ietf.org/html/rfc3626>.

- [31] IETF. Topology dissemination based on reverse-path forwarding (tbrpf). <http://tools.ietf.org/html/rfc3684>.
- [32] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *ACM/IEEE International Conference on Mobile Computing and Networks (MobiCom)*, 2000.
- [33] Jai-Jin Lim and Kang G. Shin. Gradient-Ascending Routing via Footprints in Wireless Sensor Networks. In *Proceedings of the 26th IEEE Real-Time Systems Symposium (RTSS)*, 2005.
- [34] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.
- [35] Ad Kamerman and Leo Monteban. Wavelan ii: A high performance wireless lan for the unlicensed band. In *Bell Labs Technical Journal, Summer 1997*, 1997.
- [36] C. Karlof, Y. Li, and J. Polastre. Arrive: Algorithm for robust routing in volatile environments. In *Technical Report UCB/CSD-03-1233, University of California at Berkeley*, 2002.
- [37] Brad Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254. ACM Press, 2000.
- [38] Kyu-Han Kim and Kang G. Shin. On accurate measurement of link quality in multi-hop wireless mesh networks. In *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 38–49, New York, NY, USA, 2006. ACM Press.
- [39] Young-Jin Kim, Ramesh Govindan, Brad Karp, and Scott Shenker. On the pitfalls of geographic face routing. In *DIALM-POMC '05: Proceedings of the 2005 joint workshop on Foundations of mobile computing*, pages 34–43, New York, NY, USA, 2005. ACM Press.
- [40] Murali Kodialam and Thyaga Nandagopal. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 73–87, New York, NY, USA, 2005. ACM Press.
- [41] Can Emre Koksal and Hari Balakrishnan. Quality-aware routing metrics for time-varying wireless mesh networks. In *IEEE Journal on Selected Areas in Communications, Vol. 24, No. 11*, pages 1984–1994, New York, NY, USA, November, 2006. ACM Press.
- [42] David Kotz, Calvin Newport, Robert S. Gray, Jason Liu, Yougu Yuan, and Chip Elliott. Experimental evaluation of wireless simulation assumptions. In

MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems, pages 78–82, New York, NY, USA, 2004. ACM Press.

- [43] Pradeep Kyasanur and Nitin H. Vaidya. Capacity of multi-channel wireless networks: impact of number of channels and interfaces. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 43–57, New York, NY, USA, 2005. ACM Press.
- [44] Pradeep Kyasanur and Nitin H. Vaidya. Routing and interface assignment in multi-channel multi-interface wireless networks. In *the IEEE Conference on Wireless Communications and Networking Conference (WCNC)*, 2005.
- [45] Mathieu Lacage, Mohammad Hossein Manshaei, and Thierry Turletti. Ieee 802.11 rate adaptation: a practical approach. In *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 126–134, New York, NY, USA, 2004. ACM Press.
- [46] Seungjoon Lee, Bobby Bhattacharjee, and Suman Banerjee. Efficient geographic routing in multihop wireless networks. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 230–241, New York, NY, USA, 2005. ACM Press.
- [47] X. Y. Li, K. Moaveninejad, and O. Frieder. Regional gossip routing for wireless ad hoc networks. In *IEEE Int'l Conference on Local Computer Networks (LCN)*, 2003.
- [48] Hyuk Lim, Jennifer C. Hou, and Chong-Ho Choi. Constructing internet coordinate system based on delay measurement. In *IMC '03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 129–142, New York, NY, USA, 2003. ACM Press.
- [49] Jai-Jin Lim, Daniel L. Kiskis, and Kang G. Shin. System support for management of networked low-power sensors. In *NOMS '06: Proceedings of the 10th IEEE/IFIP conference on Network Operations and Management Symposium*, pages 436–447, 2006.
- [50] Madwifi. Multiband atheros driver for wifi. <http://sourceforge.net/projects/madwifi>.
- [51] MIT. Mistlab. <http://mistlab.csail.mit.edu/>.
- [52] Michael Mitzenmacher. Compressed bloom filters. *IEEE/ACM Trans. Netw.*, 10(5):604–612, 2002.
- [53] Douglas C. Montgomery, Lynwood A. Johnson, and John S. Gardiner. Forecasting and time series analysis. 2nd Edition, McGraw-Hill Book Company, 1990.

- [54] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers. In *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, pages 234–244, New York, NY, USA, 1994. ACM Press.
- [55] Charles E. Perkins and Elizabeth M. Royer. Ad hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, 1999.
- [56] Philip Levis and Neil Patel and David Culler and Scott Shenker. Trickle: A Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks. In *First USENIX/ACM Symposium on Network Systems Design and Implementation (NSDI)*, 2004.
- [57] Robert D. Poor. Gradient routing in ad hoc networks. <http://www.media.mit.edu/pia/Research/ESP/texts/poorieepaper.pdf>.
- [58] Proxim. Orinoco 11b pci adapter. http://www.proxim.com/products/wifi/client/802.11b_pci/.
- [59] Ashish Raniwala and Tzi cker Chiueh. Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network. In *IEEE INFOCOM*, 2005.
- [60] Sean Rhea and John Kubiawicz. Probabilistic location and routing. In *IEEE INFOCOM*, pages 1248–1257, 2002.
- [61] César A. Santiváñez, Ram Ramanathan, and Ioannis Stavrakakis. Making link-state routing scale for ad hoc networks. In *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 22–32, New York, NY, USA, 2001. ACM Press.
- [62] Curt Schurgers and Mani B. Srivastava. Energy efficient routing in wireless sensor networks. In *Military Communications Conference*, 2001.
- [63] Karim Seada, Marco Zuniga, Ahmed Helmy, and Bhaskar Krishnamachari. Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 108–121, New York, NY, USA, 2004. ACM Press.
- [64] R.C. Shah and J.M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference, 2002*.
- [65] Jian Tang, Guoliang Xue, and Weiyi Zhang. Interference-aware topology control and qos routing in multi-channel wireless mesh networks. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 68–77, New York, NY, USA, 2005. ACM Press.

- [66] Di Tian and Nicolas D. Georganas. Energy efficient routing with guaranteed delivery in wireless sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference, 2003*.
- [67] Alvin Valera, Winston Seah, and S.V. Rao. Cooperative packet caching and shortest multipath routing in mobile ad hoc networks. In *IEEE INFOCOM 2003*.
- [68] Alec Woo, Terence Tong, and David Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 14–27, New York, NY, USA, 2003. ACM Press.
- [69] F. Ye, G. Zhong, S. Lu, and L. Zhang. A robust data delivery protocol for large scale sensor networks. In *Information Processing In Sensor Networks (IPSN)*, 2003.
- [70] Fan Ye, Gary Zhong, Songwu Lu, and Lixia Zhang. Gradient broadcast: A robust data delivery protocol for large scale sensor networks. In *ACM Wireless Networks*, 2005.
- [71] Hongwei Zhang, Anish Arora, and Prasun Sinha. Learn on the fly: Data-driven link estimation and routing in sensor network backbones. In *IEEE INFOCOM*, 2006.
- [72] Jerry Zhao and Ramesh Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 1–13, New York, NY, USA, 2003. ACM Press.
- [73] M. Zorzi and R.R. Rao. Geographic random forwarding (gegraf) for ad hoc and sensor networks: Energy and latency performance. In *IEEE Transactions on Mobile Computing*, 2003.