Analysis and application of hop count in multi-hop wireless ad-hoc networks

Author:
Chen, Quanjun

Publication Date:
2009

DOI:
https://doi.org/10.26190/unsworks/22955

Analysis and Application of Hop Count in Multi-hop Wireless Ad-hoc Networks

by

Quanjun Chen

B.S., Civil Aviation University of China, 1999

THE UNIVERSITY OF
NEW SOUTH WALES

SYDNEY · AUSTRALIA

A thesis submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science and Engineering

University of New South Wales

2009

This thesis entitled:
Analysis and Application of Hop Count in Multi-hop Wireless Ad-hoc Networks
written by Quanjun Chen
has been approved for the Department of Computer Science and Engineering

Supervisor: Prof. Mahbub Hassan

Co-supervisor: Dr. Salil S. Kanhere

Signature ________________ Date ________________

The final copy of this thesis has been examined by the signatory, and I find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Declaration

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

Signature ________________ Date ________________

Abstract

Hop count, i.e., the number of wireless hops a packet has to go through to reach the destination, is a fundamental metric in multi-hop wireless ad-hoc networks. Network performance, such as throughput, end-to-end delay, energy consumption, and so on, depends critically on hop count. Previous work on modeling hop count is limited in making unrealistic simplifying assumptions either at the physical or network, or both layers of the communication protocol stack. A key contribution of this thesis is to present an analytical model to derive the probability distribution of hop count under realistic assumptions at both physical and network layers. Specifically, the model considers a log-normal shadowing radio propagation capable of accommodating the random signal fading observed in most wireless communication environments, and the widely used geographic routing at the network layer. Validation of the model is achieved by a comprehensive set of simulation experiments including a trace driven simulation of a real-word vehicular ad-hoc network. The model reveals that the presence of randomness in radio propagation reduces the required number of hops to reach a given destination significantly. To demonstrate the utility of the proposed hop count model, the thesis proposes three new applications which address some of the key challenges in multi-hop wireless networks. The first application derives the per-node packet forwarding load in multi-hop wireless sensor networks and reveals that the nodes in the vicinity of the base station has a significantly less forwarding load than previously thought under simplifying radio propagation and routing assumptions.

The second application demonstrates that using hop count as a measure of distance traveled by a data packet, geocasting can be achieved in multi-hop wireless networks in situations when some of the network nodes do not have access to reliable location information. Finally, the proposed hop count model is used to evaluate the performance of the third application which demonstrates that the overhead of geographic routing can be reduced significantly by embracing a position update philosophy which adapts to the mobility and communication patterns of the underlying ad-hoc network.

Dedication

To my parents, Dexin Chen and Xiuying Guo, for their love and support.

Acknowledgements

I would like to thank all people who have helped and inspired me during my doctoral study. Firstly, I would like to express my deep and sincere gratitude to my supervisor, Professor Mahbub Hassan, for his excellent guidance during my research study. Mahbub has been an incredible supervisor, who has always challenged me to think in a logical and critical manner. His understanding, encouraging and insightful comments have provided a good basis for my present thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

Next, I am deeply grateful to my co-supervisor, Dr. Salil S. Kanhere, for his continuous support, patience, motivation and immense knowledge. His guidance has helped me through the research and writing of this thesis. In particular, I want to thank him for his very detailed and constructive comments on every draft that I wrote. This would never have been possible without his help and supervision.

My sincere thanks also goes to Dr. Wen Hu and Dr. Peter Corke for offering me the internship opportunity at CSIRO ICT Brisbane and guiding me to work on diverse and exciting projects. I would like to express my gratitude to all the members of Network Research Lab (NRL) at UNSW especially Yuvraj Krishna Rana (kris) and Tuan Le Dinh for their encouragement and support and Junaid Qadir and Nadeem Ahmed for their kind help. I would also like to thank Xin Zhao, Yifei Dong and Hailun Tan, who made my stay more colorful and enjoyable.

I am also indebted to my friends Wei Tang and Sam Zhang. They helped me immensely by giving me encouragement and friendship. Finally, my deepest gratitude goes to my parents and my brothers and my sisters. Without their understanding and support, it would have been impossible for me to finish this work.

Author's Publications

1. **Q. Chen**, S. S. Kanhere and M. Hassan. "Modeling Hop count In Wireless Ad-hoc Network". Submitted to International Journal on Wireless & Optical Communications (IJWOC), Jun. 2009.
   This publication forms the part of Chapter 3 of this thesis.

2. **Q. Chen**, S. S. Kanhere and M. Hassan. "Analysis of Per-node Traffic Load in Multi-hop Wireless Sensor Networks". IEEE Transaction on Wireless Communication, Volume 8, Issue 2. Feb. 2009.
   This publication forms the part of Chapter 4 of this thesis.

3. **Q. Chen**, S. S. Kanhere and M. Hassan. "Geocasting Without Location Information". Submitted to Computer Communications, Sep. 2009.
   This publication forms the part of Chapter 5 of this thesis.

4. **Q. Chen**, S. S. Kanhere and M. Hassan. "Network Mobility and Traffic Load Aware Beacon Update in Geographic Routing". Submitted to IEEE Transaction on Mobile Computing, Dec. 2008.
   This publication forms the part of Chapter 6 of this thesis.

5. **Q. Chen**, S. S. Kanhere, M. Hassan and Y. K. Rana. "Distance-based local geocasting in multi-hop wireless networks". Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), Hong Kong, China, Mar. 2007.
   This publication is the preliminary work of Chapter 5 of this thesis.

6. **Q. Chen** and S. S. Salil and M. Hassan and K.-C. Lan, "Adaptive Position Update in Geographic routing". Proceedings of IEEE International Conference on Communications (ICC), Istanbul, Turkey, Jan. 2006.
This publication is the preliminary work of Chapter 6 of this thesis.

Acronyms

AODV Ad hoc On-Demand Distance Vector routing

CBR Constant Bit Rate

DSR Dynamic Source Routing

DSRC Dedicated Short-Range Communications

GPS Global Positioning System

GPSR Greedy Perimeter Stateless Routing

MAC Media Access Control

MANET Mobile Ad Hoc Network

NS2 Network Simulator 2

RWP Random Way Point

TTL Time-To-Live

VANET Vehicular Ad-Hoc Network

WSN Wireless Sensor Network

Symbols

R	Average radio range of nodes
ρ	Node density, i.e. the number of nodes per unit area
ε	Quantization interval for discrete distance space
i, j, d	Euclidean distance from a node to a destination
ξ	Signal randomness parameter in log-normal shadowing radio model
$P_{\wedge}(s)$	The probability that two nodes separated by distance s can communicate with each other
$P_{i,j}$	Forwarding probability. i.e. the probability that a sensor at distance i from the sink can forward its packets to the sensor at distance j
H, h	Number of hops from a source to a destination
$\overline{H}$	The mean hop count from a source to a destination

Contents

List of Tables

Table

List of Figures

Figure

Chapter 1

Introduction

1.1 Motivations

Wireless ad-hoc networks have received significant attention in recent years due to technological advances and the success of building small-sized and energy-efficient reliable wireless devices [1, 2, 3, 4]. Wireless ad-hoc networks are decentralized, self-organizing communication networks, which do not rely on any fixed infrastructure. Such networks often operate in a peer-to-peer fashion. Each wireless device can work as a terminal station and a router which helps other nodes to relay their packets.

Due to the decentralized nature and ease of deployment, wireless ad-hoc networks are suitable for a variety of applications. Examples include disaster recovery, military operations, vehicular network, health care assistance, environment and habitat monitoring [5, 6]. The autonomous and spontaneous nature of wireless ad-hoc networks also impose many challenges. For example, the dynamical changes in topology require fast, adaptive routing protocols in order to achieve reliable communication. Many applications, e.g. environment monitoring, often need a large number of nodes deployed, which require scalable routing protocols and MAC protocols. The energy-constrained wireless devices, e.g. sensor nodes or personal wireless devices, also necessitate energy-efficient designs at each layer of network protocol stacks.

Multi-hop communications have been an important character in wireless ad-hoc networks. This is because it extends the connectivity of nodes from its immediate neighbors to nodes far beyond their radio range, a facility which helps form large scale wireless networks as required by many applications. Multi-hop communication also comes with its own challenges. For example, the wireless transmission at each hop consumes scarce resources such as bandwidth and energy, increases end-to-end packet delay, as well as the overall probability of packet loss in the network. **Hop count**, i.e., *the number of hops or relay nodes a packet has to go through from the source to reach its destination*, therefore becomes an important measure in the performance evaluation of multi-hop wireless ad-hoc networks [7, 8, 9, 10, 11, 12]. Generally, hop count has a determinative effect on the following performance measures in wireless ad-hoc network:

- **Throughput**: The throughput of a communication pair is inversely proportional to the hop count from the source to the destination, as shown in previous work [9, 10, 13]. This is due to channel sharing and radio interference among neighboring nodes. Also, with the increase of hop count in a routing path, the number of packet successfully received tends to decline due to the increased chances of a routing failure and packet collisions.

- **Energy Consumption**: Energy saving is a critical issue in networks with energy-constraint wireless devices, such as sensor nodes or personal wireless devices. When routing paths from sources to destinations have a larger number of hops, more intermediate nodes are involved in wireless transmitting and receiving, which inevitably consume more energy network-widely.

- **End-to-end Delay**: The wireless transmission at each hop introduces some delay, including packet processing delay, propagation delay, queuing delay and collision back off delay. A longer routing path clearly leads to a larger end-to-end delay.

- **Routing Overhead**: As discussed in [14, 15], a longer routing path is more likely to be disrupted due to the topology dynamics, which results in more routing overhead (generated by the routing recovery mechanisms), especially in reactive routing protocols, e.g. DSR [16, 17], AODV [18].

Despite its importance, hop count has not been adequately investigated in the literature. Most previous work [7, 8, 9] uses a simple and unrealistic approach to estimate hop count. This estimation simplifies hop count as the ratio of source-to-destination Euclidean distance to the radio range. This estimation assumes that nodes can always find the next hop on the border of the radio range, and that the next hop lies on the straight line connecting the source and the destination. These assumptions are valid only when the node density in the network tends to go to infinity, a premise which is never true in reality. Several works [19, 20, 21, 22, 23, 24, 25] have attempted to analyze hop count more accurately. However, these works still make unrealistic over simplifying assumptions either at the physical or network level, or at both layers of the communication protocol stack. Particularly, most of this literature assumed an **ideal radio model**, where radio coverage is a perfect disk with no random fading. This radio model has been proven to be far from realistic [26, 27, 28] and therefore limits its practical implications, leaving the realistic hop count analysis an open problem in need of a solution. Furthermore, the applications that can exploit hop count knowledge are still lagging behind. The existing applications [22, 24] only provided simple models to integrate hop count to estimate end-to-end delay, energy consumption or the nodes coordinates (localization algorithm). More advanced sophisticated applications that fully make use of hop count results in order to address the key challenges in wireless networks are still missing in literature and studies in this area.

1.2 Proposed Work

In the first part of this thesis (**Chapter 3**), we aim to derive the probability distribution and the mean of hop count under realistic assumptions at both network and physical layers. At the network layer, we consider **greedy geographic routing**, a well known concept considered by many routing protocols [29, 30, 31, 32]. The underlying principle used in these protocols involves selecting the next routing hop from amongst a node's neighbors, which is geographically closest to the destination. Since, the forwarding decision is based entirely on local knowledge, it obviates the need to create and maintain routes for each destination. By virtue of these characteristics, geographic routing protocols are highly scalable and particularly robust to frequent changes in the network topology. Furthermore, since the forwarding decision is made on the fly, each node always selects the optimal next hop based on the most current topology. Several studies [29], [33], [34] have shown that these routing protocols offer significant performance improvements over topology-based routing protocols such as DSR [17] and AODV [18].

At the physical layer, we consider both idealistic radio model and the realistic **log-normal shadowing model**, thus enabling us to compare the impact of the two on the results. In ideal radio model, the existence of a direct link between two nodes is a binary problem, i.e. either there exists a direct link or there does not, solely based on the distance separating them. Thus, the radio coverage of a node in ideal radio model is a perfect circle. However, under the log-normal shadowing radio model, the existence of direct link between two nodes is a probabilistic problem. The probability depends on the distance between the two nodes and a random signal fading. The radio coverage of each node in this model is irregular, which resembles realistic situations [27, 28]. Therefore, the log-normal shadowing radio model has been accepted as a realistic radio model.

Since the hop count is closely determined by the behavior of packets' progress toward the destination, i.e. how the packet is forwarded towards the destination, we use a discrete Markov chain to model the hop-by-hop progress of a packet from the source to the destination. We firstly identify the state transition probability (i.e. the forwarding probability from a node to another node) in this Markov chain model. Then, based on the state transition probability, we recursively calculate the hop count distribution and the mean value. Further, due to the computation complexity of the recursion, we also propose approximations to ease the calculation of mean hop count. Note that, in each of these steps, we consider both the ideal radio model and the log-normal shadowing radio model. We conduct a rich set of simulations to validate our analytical model. The comparison results well justify our proposed model. The analysis results are further confirmed through a trace driven simulation of a practical vehicular ad-hoc network that exhibits realistic topologies of public transport buses in a metropolitan city.

In the second part of our work, we propose three novel applications that utilize our analysis results of hop count to address some challenging issues in wireless networks. In the first application (**Chapter 4**), we present an analytical model to estimate per-node traffic load in wireless sensor networks. The analysis gives a solid foundation to estimate the energy consumption and the lifetime of the sensor networks. In the second application (**Chapter 5**), we propose TTL-based geocasting for multi-hop wireless network. Our approach eliminates the requirements of nodes' location information compared to the previous geocasting schemes. We estimate the expected hop count that can cover the required geocast region and then use it as TTL value to limit the propagation of a packet within the geocast region. In the last application (**Chapter 6**), we propose Adaptive Position

Update (APU) for geographic routing, which enable nodes to update position adaptively to nodes mobility and traffic load. We then utilize our analysis results of hop count to analyze the performance of APU. We introduce these applications in detail in the next three subsections.

1.2.1 Analysis of Per-node Traffic Load in Multi-hop Wireless Sensor Networks

Wireless Sensor Networks (WSN) are being increasingly used in a variety of applications ranging from health care, environmental monitoring to industrial automation [4]. Ensuring energy efficient operation is critical, especially given that a typical WSN is deployed in remote and unaccessible areas and sensor nodes are equipped with a limited battery source. It has been well accepted that the energy expended in transmission and reception of packets forms a significant component of the total energy budget of a sensor node [4, 35]. Consequently, an analytical model that can accurately estimate the traffic load incurred at each sensor node is instrumental in predicting the energy consumption of the nodes and thus the operational lifetime of the entire WSN. In addition, knowledge of the energy expenditure of nodes can be useful in planning deployment and maintenance of the WSN. The network designer can deploy redundant nodes in regions where nodes are expected to expend their energy at a higher rate.

Analytical models which can accurately characterize the traffic load of nodes in a WSN are rare. The available models, e.g., those presented in [36, 37, 38, 39], only provide for a coarse-grained characterization of the traffic load. In addition, these work have adopted the idealistic circular coverage radio model, which is known to be a poor abstraction of the real communication environment.

In this application, we take a first step towards developing a detailed and precise analytical model for estimating the per-node traffic load in a WSN. The traffic load at a node is a collective results of the packets originated from this node and the packets relayed by this node. The former one is given by the application requirement, while the latter one depends on the packet forwarding behavior of the routing scheme. In our previous hop count analysis (Chapter 3), we have identified this forwarding behavior of greedy geographic routing. Specifically, we apply a Markov chain model to analyze the probability that a node forwards its packets to another node. Having this knowledge, a node can estimate the chance that each of its neighbor uses this node to relay their packets, therefore the node is able to estimate the total number of packets it will relay.

1.2.2 Geocasting Without Location Information

In many wireless ad-hoc networking applications, a wireless node often needs to disseminate information to all other nodes within a target geographical distance from it. For example, In Intelligent Transportation System (ITS), road safety applications require vehicles to flood warning messages to other vehicles within an immediate neighborhood to prevent accidents or to warn of traffic hazards [40]. In a wireless sensor network, the sink node (base station) often need to send various types of query and command messages to all sensor nodes within a predetermined geographic region around it.

To support the aforementioned applications, researchers have proposed a new kind of packet forwarding primitive, called geocasting [41]. While geocasting solves the problem of limiting information propagation within geographic boundaries, it assumes that every node will have access to precise location information at all times. This assumption is valid in general and is well backed up by the falling cost of location sensing hardware, e.g., GPS circuits, and advancements in GPS-less localization [42]. There are, however, many practical situations when a particular (set of) wireless node(s) may not have access to reliable location information. For example, for many tiny wireless sensing devices, e.g., MicaZ and TelosB [43], continuously running the GPS interface would deplete the battery prematurely. For vehicles inside a tunnel or in urban roads, access to one or more satellites may be temporarily blocked (note that GPS requires line-of-sight to four satellites). Because of these, and many other unavoidable circumstances, it is practical to provision for alternate methods of forwarding so that geocasting can proceed in the event that reliable location information is absent.

We propose to use the well-known time-to-live (TTL) forwarding, which limits the propagation of a packet within a specified number of hops (TTL)[1] from the source. A key feature of TTL forwarding, which is of particular interest to us, is that it does not require a node to know its location coordinates.

The main challenge in achieving geocasting with TTL forwarding is how to select the right TTL. The intent of this work is to conduct a systematic study exploring the suitability of using TTL forwarding to achieve geocasting in situations when some (or all) nodes in the geocast area do not have access to reliable location information. Our objective is to quantify, as a function of the TTL, the achievable coverage, i.e., the percentage of nodes in the geocast area that receive a copy of the original broadcast from the source, and the broadcast overhead, i.e., the total number of transmissions needed for each message to achieve the geocast.

The achievable coverage and broadcast overhead given a TTL depends on the geographical distance covered by each hop (referred as hop distance). In our first part of work, we have analyzed the hop distance under the consideration of both ideal radio model and realistic radio model. Based on those results, we develop an analytical model to accurate quantify coverage and broadcast overhead. Given the geocasting performance for each TTL, we are able to select the appropriate TTL value for each network setting.

[1] TTL is a misnomer in the sense that it actually specifies the number of hops, not the time, a packet will live.

1.2.3 Mobility and Traffic Adaptive Position Update for Geographic Routing Protocols

With the growing popularity of positioning devices (e.g. GPS) and other localization schemes [44], geographic routing protocols are becoming an attractive choice for use in wireless ad hoc networks [29], [45, 46, 47, 48]. The forwarding strategy employed in the geographic routing protocols requires the following information: (i) the position of the final destination of the packet and (ii) the position of a node's neighbors. The former can be obtained by querying a *location service* such as the Grid Location System (GLS) [49] or Quorum [50]. To obtain the latter, each node exchanges its own location information (piggybacked in beacons) with its neighboring nodes. This allows each node to build a local map of the nodes within its vicinity, referred to as the **local topology**. In most geographic routing protocols (e.g. GPSR [29], GeoCast [41], [51], [52]), beacons are broadcast periodically for maintaining an accurate neighbor list at each node.

Position updates are costly in many ways. Each update consumes node energy, wireless bandwidth, and increases the risk of packet collision at the medium access control (MAC) layer. Clearly, given the cost associated with transmitting beacons, it makes sense to adapt the frequency of beacon updates to the node mobility and the traffic conditions within the network, rather than employing a static periodic update policy. For example, if certain nodes are frequently changing their mobility characteristics (speed and/or heading), it makes sense to frequently broadcast their updated position. However, for nodes that do not exhibit significant dynamism, periodic broadcasting of beacons is wasteful. Further, if only a small percentage of the nodes are involved in forwarding packets, it is unnecessary for nodes which are located far away from the forwarding path to employ periodic beaconing because these updates are not useful for forwarding the current traffic.

In this work, we propose a novel beaconing strategy for geographic routing protocols called **Adaptive Position Updates strategy (APU)** [53]. Our scheme eliminates the drawbacks of periodic beaconing by adapting to the system variations. APU incorporates two rules for triggering the beacon update process. The first rule, referred as **Mobility Prediction (MP)**, uses a simple mobility prediction scheme to estimate when the location information broadcast in the previous beacon becomes inaccurate and therefore schedules the next beacon. The second rule, referred as **On-Demand Learning (ODL)**, aims at improving the accuracy of the topology along the routing paths between the communicating nodes.

We model APU to quantify the beacon overhead and the local topology accuracy. The beacon overhead generated by the first rule (MP) is a function of nodes mobility, while the beacon overhead triggered by the second rule (ODL) depends on the number of packet transmissions. Since a packet is retransmitted at each hop along the routing path, the total transmission volume relies on the hop count of each pair of communicating nodes. In our first part of work, we have derived the average hop count from a source to a destination based on the hop count distribution. We also propose a simplified approximation that can reduce the computation complex significantly. In this work, we utilize these average hop count results to model the beacon overhead incurred in APU.

1.3 Contribution List

The main contributions of this thesis are listed as follows:

1. We accurately analyze the hop count distribution and mean value for greedy routing in both ideal radio and realistic radio environments (i.e. log-normal shadowing radio). We also propose approximations to ease the calculation of average hop count. Our hop count analysis reveals that

the widely used approach of taking the ratio of source-to-destination Euclidean distance to radio range may significantly overestimate the actual hop count if random signal fading is present in the wireless network. Our analysis also challenges the current belief that greedy geographic routing can (approximately) find the shortest path between a source and destination in a multi-hop wireless network. We show that this is true only if no random fading is present. Under the presence of random fading, greedy routing can take paths significantly longer than the shortest path (Chapter 3).

2. Based on the hop count analysis results, we present an analytical model for estimating the per-node traffic load in a multi-hop wireless sensor network considering both idealistic and realistic radio model. Our results confirm that irrespective of the radio models, the traffic load generally increases as a function of the node's proximity to the sink. In the immediate vicinity of the sink, however, the ideal radio model shows the existence of a volcano region near the sink, where the traffic load drops significantly. On the contrary, with the log-normal shadowing model, the traffic load actually increases at a much higher rate as one approaches the sink, resulting in the formation of a mountain peak (Chapter 4).

3. Based on the hop count results, we propose a novel TTL-based geocasting scheme. Our approach eliminates the location requirement imposed by the previous geocasting schemes. Our analytical results, which are validated by simulations, confirm that TTL-based scheme can achieves similar high performance as location-based schemes (Chapter 5).

4. We propose the **A**daptive Position Update (APU) strategy for geographic routing, which dynamically adjusts the frequency of position updates

based on the mobility dynamics of the nodes and the forwarding patterns in the network. Based on the hop count analysis results, we theoretically analyze the performance of the proposed APU. The analysis and simulation results shows that APU can significantly reduce the update cost and improve the routing performance in comparison with periodic beaconing and other recently proposed updating schemes. (Chapter 6)

1.4 Thesis Organization

The rest of the thesis is organized as follows. In **Chapter 2**, we review the related work focusing on introducing geographic routing and previous hop count analytical work. In **Chapter 3**, we analyze the relationship between the source-to-destination distance and the hop count under two radio models. In **Chapter 4**, we introduce the first application of the hop count results, i.e. the analysis of per-node traffic load for wireless sensor networks. In **Chapter 5**, we proceed to present the second application, i.e. geocasting without location information. In **Chapter 6**, we propose the mobility and traffic adaptive position update (APU) for geographic routing. Finally, **Chapter 7** concludes this thesis.

Chapter 2

Literature Review

In this chapter, we will review and discuss previous work which is related to our proposals. We will first introduce the geographic routing protocol, which includes the geographic forwarding strategy, position update (this maintains the neighbors' location information) and geocast technique (this is a technique which disseminates messages to a particular geographic area). In the second part, we will first review the previous analytical works that study the relationship between the distance of the two communicating node and the hop count. We will then discuss the applications that have been proposed to utilize this hop count knowledge.

2.1 Geographic Routing

With the growing popularity of positioning devices (e.g. GPS) and other localization schemes [44], geographic routing protocols are becoming an attractive choice for use in wireless ad hoc networks [29], [33], [54], [55]. The underlying principle used in these protocols involves selecting the next routing hop from amongst a node's neighbors, which is geographically closest to the destination. Since the forwarding decision is based entirely on local knowledge, it obviates the need to create and maintain routes for each destination. By virtue of these characteristics, position-based routing protocols are highly scalable and particularly robust to frequent changes in the network topology. Furthermore, since the forwarding decision is made *on the fly*, each node always selects the optimal next hop based on the most current topology. Several studies [29], [33], [34] have shown that these routing protocols offer significant performance improvements over topology-based routing protocols such as DSR [17] and AODV [18].

We will introduce geographic routing from three aspects that are mostly related to our proposed work. We will start with the geographic forwarding strategies that define how to advance packets to destinations (Section 2.1.1). We will then discuss the position update mechanisms that maintain the one-hop neighbor's locations (Section 2.1.2). Lastly, we will introduce the geocast protocols that aim to deliver packets to the nodes within a certain geographic region (Section 2.1.3).

2.1.1 Geographic Forwarding Strategies

In geographic routing, the forwarding decision is based on local topologies of the forwarding nodes. When a source has a packet to send, it first retrieves the location of the destination from the location service [49, 50, 56]. Then the source piggybacks the destination's location in the data packet. Each intermediate forwarding node selects the next hop based on the destination's location and the locations of its one-hop neighbors. The geographic forwarding strategies are generally classed into two categories: greedy routing and face routing. The former is used to advance packets progressively toward the destination, whilst the latter is used as a recovery routing when greedy routing fails.

2.1.1.1 Greedy routing

Greedy routing, firstly proposed by Finn in [57], has been widely accepted as one of main geographic routing protocols. In greedy routing, the forwarding node selects the next hop that is closet to the destination amongst its one-hop

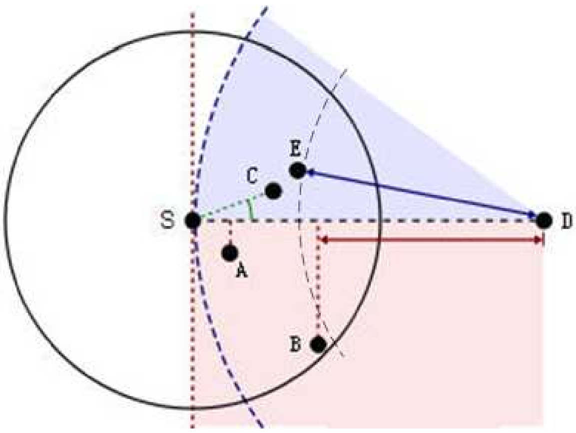


Figure 2.1: Example of greedy routing

neighbors. In doing so, greedy routing aims to minimize the remaining distance from the next hop to the destination and therefore reduce the number of hops from the source to the destination. We illustrate greedy routing using Fig. 2.1, where the node S and D denote the current forwarding node and the destination of a packet respectively. The forwarding node S calculates the distance from each of its one-hop neighbors to the destination based on the location information which it knows. In this example, node S finds that the neighbor E is the closet one to the destination and therefore selects node E as the next hop.

Most Forward within Radius (MFR [58]), a variant of greedy routing and is the earliest geographic routing proposed. In MFR, a forwarding node selects the next hop that can maximise the progress of a packet in the direction of the destination. For example, in the Fig. 2.1, the forwarding node S projects its neighbors on the straight line SD connecting both itself and the destination. The forwarding node S measures the distances from itself to those projected points, and selects the neighbor that makes the biggest distance as the next hop. In this example, node B should be selected as the next hop.

Another variant is compass routing, also referred to as DIR routing as proposed by Krannakis [59]. In compass routing, the forwarding node selects the neighbor whose edge has the smallest slop to the straight line connecting the source and the destination. In the example Fig. 2.1, node C should be selected as the next hop. If we draw a line connecting all the forwarding nodes selected by the compass routing, the trajectory should close to straight line connecting the source and the destination. Therefore we can see that compass routing tries to minimize the total traversal distance of a packet.

Stojmenovic in [60] conducted simulations to compare the performance of greedy routing, MFR and compass routing. They concluded that greedy routing can find the same routing paths as MFR in most of cases, whereas compass routing produces longer routing paths. Karp in [29] has shown that greedy routing can approximately find the shortest path from the source to the destination in a relative dense network. However, comparing to the traditional shortest path routing, which requires global topology knowledge, greedy routing only needs the one-hop neighbors' information. Therefore, greedy routing has been recognized as an efficient routing protocol to forward packets to destinations and also has been widely used as a primitive routing component in geographic routing protocols.

2.1.1.2 Face routing

Despite its advantages, greedy routing cannot successfully find the routing path in some cases. For example, in Fig. 2.2, the current forwarding node S finds there is no neighbor closer to the destination than itself: a situation referred as the "local maximum" or the "hole problem". In this case, greedy routing fails even though a routing path from the node S to the destination exists, as shown in Fig. 2.2. Clearly, it is therefore not wise to simply drop the packets or flood the packets in such situations.

Face routing is therefore proposed to guarantee that the packet can be delivered to the destination (GFG[61], GPSR [29], GOAFR [62, 63]). Face routing consists of two steps. Firstly, it abstracts the network topology as a graph and

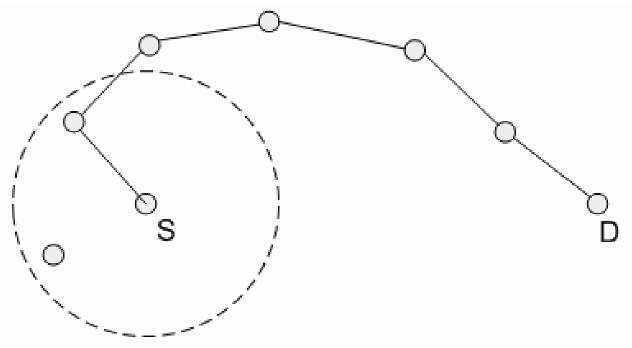
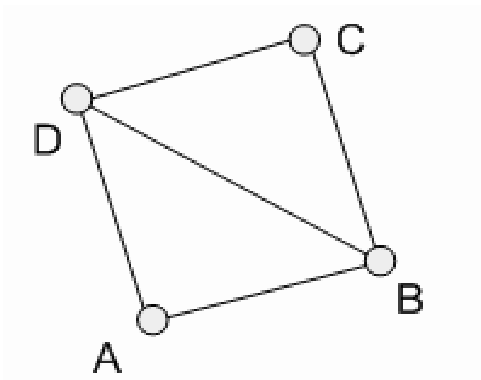
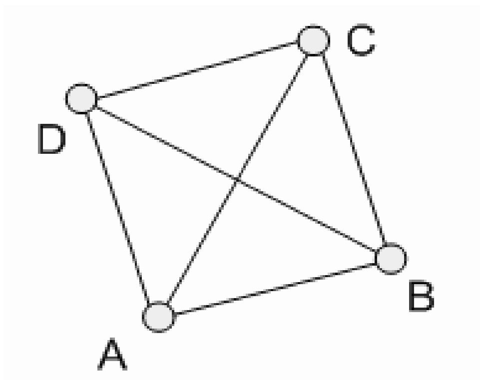


Figure 2.2: Example of failure of greedy routing



(a) A planar graph



(b) Not a planar graph

Figure 2.3: Examples of planar graph

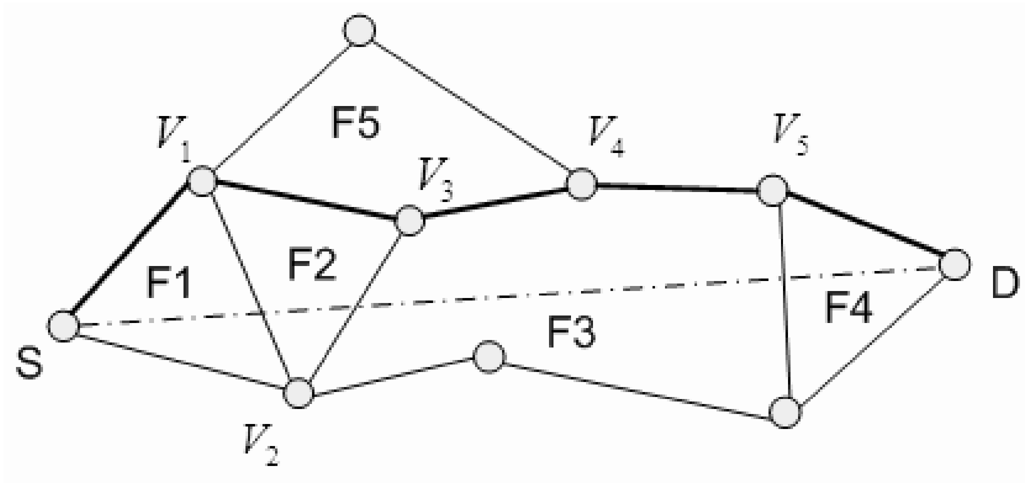


Figure 2.4: Example of face routing

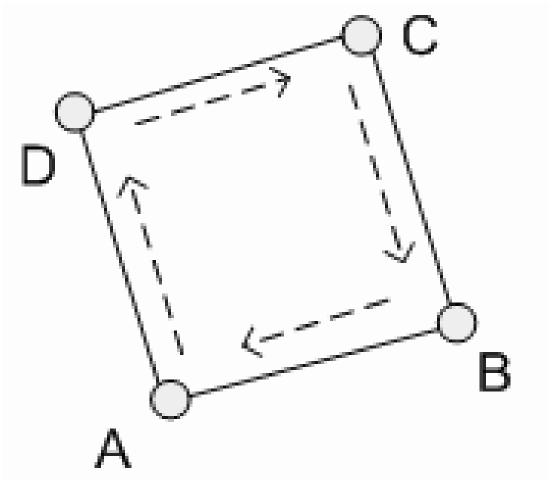


Figure 2.5: Example of right hand rule

converts the graph into a planar graph by removing some edges [29, 64, 65] (A planar graph is a graph with no crossing edges). For example, Fig. 2.3(a) is a planar graph, whereas Fig. 2.3(b) is not, since edge AC crosses with edge BD. A planar graph consists of faces, which are enclosed polygonal regions bounded by edges. For example, in Fig. 2.4, the planar graph consists of five faces. Note that the straight line connecting the source and the destination crosses the faces numbered from F1 to F4. If the packet visits those faces consecutively, the packet is guaranteed to be delivered to the destination. Therefore, the second step of face routing is to define the face traversal strategy to forward the packet along the faces of the planar subgraph.

The face traversal strategy again, includes two components. The first one defines how to travel (or tour, explore) inside a face, i.e. giving an order in which to travel the edges of a face. The right-hand rule is a popular method for this purpose. In right-hand rule, a packet traverses the edges of a face in a clockwise order. Fig. 2.5 gives an example, where the edge traversal order is edge AD, DC, CB, then BA. Clearly, a packet should not traverse a face infinitely. It should move from one face to the next face, when appropriate, in order to deliver the packet to the destination. Therefore, the second component, i.e. face changes, defines the sequence of faces being visited and when it should switch face from the current face to the next. Several algorithms have been proposed to effectively traverse faces, including GFG [61], GPSR [29], compass routing II [59], GOAFR [62] and $GOAFR^{+}$ [63].

2.1.2 Position Update

In geographic routing, the forwarding decision at each node is based on the locations of its one-hop neighbors and also the location of the destination. A forwarding node therefore needs to maintain these two types of locations. Many works, e.g. GLS [49], Quorum System [50], Homezone [56], have been proposed to discover and maintain the location of a destination. However, the maintenance of one-hop neighbor's location has often been neglected. Some geographic routing schemes, e.g. [66, 63, 62], simply assume that a forwarding node knows the location of its neighbors'. Some others, e.g. [29, 41, 51, 52], uses periodical beacon broadcasting to exchange its neighbor's locations. In the periodic beaconing scheme, each node broadcasts a beacon with a fixed beacon interval. If a node does not hear any beacon from a neighbor for a certain time interval, referred as a "neighbor time-out interval", the node considers this neighbor has moved out of the radio range and removes the outdated neighbor from its neighbor list. The neighbor time-out interval often is multiple of the beacon interval.

Son et al. [67] have demonstrated that the inaccuracy of the location information has a significant impact on the performance of geographic routing protocols. They applied a mobility prediction scheme based on nodes moving velocity and studied its impact on their performance. However, they only used the prediction scheme to compute current position of neighbors and still employed periodic beacon updates.

Heissenbuttel et al. [68] have showed that a periodic beaconing can cause inaccurate local topologies in highly mobile ad-hoc networks, which leads to performances degradation, e.g. a frequent packet loss and longer delay. The authors discuss that the outdated entries in the neighbor list is the major source which decreases performance. They proposed several simple optimizations that adapt the beacon interval to node mobility or traffic load, including distance-based beaconing, speed-based beaconing and reactive beaconing. We will discuss these three schemes below.

In distance-based beaconing, a node transmits a beacon when it has moved a given distance d. The node removes an outdated neighbor if the node does not hear any beacons from its neighbor while the node has moved more than k-times the distance d, or alternatively after a maximum time-out of five seconds. In other words, if a node moves at speed of v, its beacon interval is d/v and neighbor time-out interval is the minimum of $\{k \cdot d/v, 10\text{s}\}$. This approach therefore is adaptive to node mobility, e.g. a faster moving node sends beacons more frequently and vice versa. However, this approach has two problems. Firstly, a slow node may have many outdated neighbors in its neighbor list since the neighbor time-out interval at the slow node is longer. Secondly, when a fast moving node passes by a slow node, the fast node may not detect the slow node due the infrequent beaconing of the slow node, which reduces perceived network connectivity.

In the speed-based beaconing, the beacon interval is dependent on the node speed. A node determines its beacon interval from a predefined range $[a, b]$ with the exact value chosen being inversely proportional to its speed. More specifically, the beacon interval of a node moving at speed v is estimated as, $a + (b - a) \cdot (\frac{v_{max} - v}{v_{max} - v_{min}})^n$, where $[a, b]$ is a predefined interval range and v_{max} and v_{min} are the maximum and minimum speed of the node. The neighbor time-out interval of a node is a multiple k of its beacon interval. Nodes piggyback their neighbor time-out interval in the beacons. A receiving node compares the piggybacked time-out interval with its own time-out interval, and selects the smaller one as the time-out interval for this neighbor. In this way, a slow node can have a short time-out interval for its fast neighbor and therefore eliminate the first problem presented in distance-based beaconing. However, speed-based beaconing still suffers from the problem that a fast node may not detect the slow nodes.

In reactive beaconing, the beacon generation is triggered by data packet transmissions. When a node has a packet to transmit, the node first broadcasts a beacon request packet. The neighbors, overhearing the request packet, respond with beacons. Thus, the node can build an accurate local topology before data transmission. However, this process is initiated prior to each data transmission, which can lead to excessive beacon broadcasts, particularly when the traffic load in the network is high.

Geographic routing tends to forward packets to the neighbors near the border of the radio range in order to maximize the progress distance toward destinations. However, the neighbors near the border which can easily move out of radio range, or have weak signal that exhibits higher bit error rate [27, 28]. Heissenbuttel et al. [68] propose a receiving power threshold based neighbor maintenance scheme, which excludes those neighbors near the border of the radio range or which has a weak link, when constructing the neighbor list. In this scheme, a node samples the power when receiving a beacon and only processes the beacons received at a power level greater than a certain threshold. Thus the node only includes the neighbors with a "good link" or a "stable link" in its neighbor list. A shortcoming of this scheme is the difficulty to determine the receiving threshold. Also, this scheme requires the PHY layer to accurately estimate the receiving power level, which is not practical for some small radio devices.

2.1.3 Geocasting

Geocasting is a set of protocols that utilizes nodes' locations in order to disseminate packets to a certain geographic region. All nodes within this region, referred as the "geocast region", are required to receive the packets. A geocasting protocol typically consists of two steps. The first step is to forward packets from the source to the geocast region. The second is to flood packets within the geocast region.

2.1.3.1 Forwarding Packets to Geocast Region

A simple way to forward packets to a geocast region is global flooding, where each node in the network broadcasts a packet at least once. Clearly, global flooding is not attractive because the broadcasting in the region, other than the geocast region, is considered to be wasteful. An improvement to this process would be to restrict the flooding in a region that covers the source and the geocast region. This technique is used in DREAM [33], LAR [55] and [69].

In DREAM, the geocast technique is used to assist the delivery of packets from a source to a destination. A source node estimates the expected region (i.e. geocast region) that the destination currently may reside in. Then the

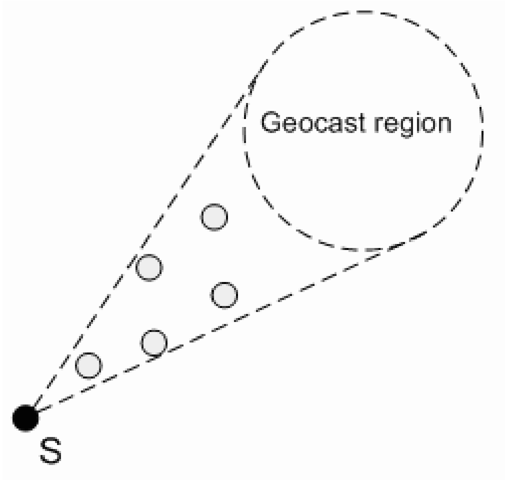


Figure 2.6: Geocasting in DREAM

source floods the packet in the direction of the geocast region. Fig. 2.6 illustrates an example, where S is the source and D is destination. The flooding region is bounded by the two tangent lines to the geocast region, which restricts the flooding in the direction of the geocast region.

In LAR [55], the geocast is used to restrict the flooding of control packets when establishing the routing path in reactive topology-based routing protocols, such as AODV and DSR. In [69], LAR has been extended to flood data packets to a geocast region. We introduce the extension version here. Three algorithms were proposed in [69]. The first algorithm, FRFZ, restricts the flooding in the smallest rectangle which includes the source and the geocast region, as illustrated in Fig. 2.7. In the second algorithm, ARFZ, each intermediate node re-estimates the rectangle that cover the intermediate node itself and the geocast region, which results in a smaller flooding area compared to the fixed rectangles, as shown in Fig. 2.8. In the third algorithm, PCN, only the intermediate nodes are closer to the geocast region broadcast packets, as shown in Fig. 2.9.

DREAM, LAR and the three algorithms proposed in [69] basically define a restricted flooding area that can forward the packet toward the geocast region. Inside these areas, a simple flooding approach, i.e blind flooding, can be used.

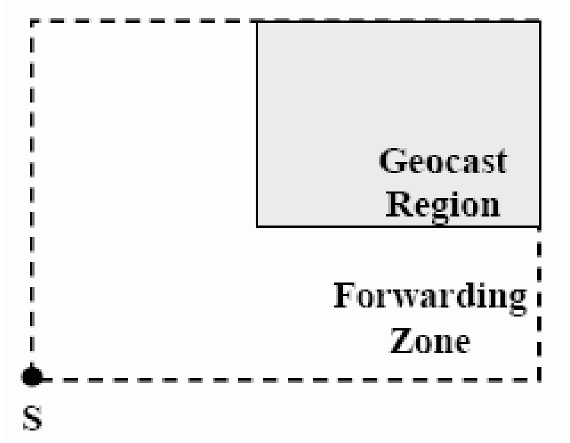


Figure 2.7: Geocasting: Fixed Rectangular Forwarding Zone (FRFZ)

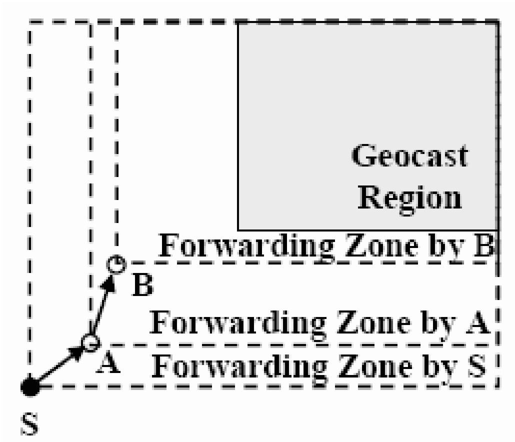


Figure 2.8: Geocasting: Adaptive Rectangular Forwarding Zone (ARFZ)

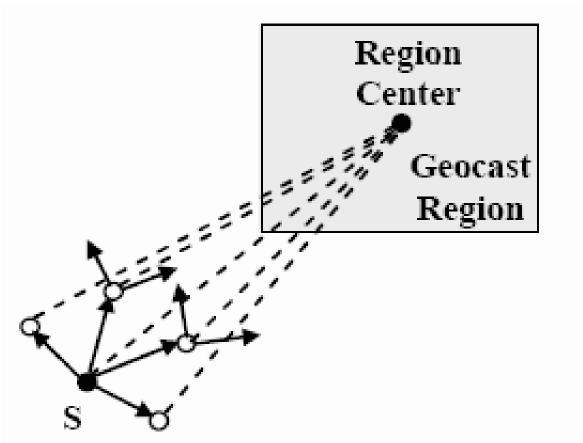


Figure 2.9: Geocasting: Progressively Closer Nodes (PCN)

In blind flooding, a node compares its own location with the flooding area info which is piggybacked in the packet. Only the node residing inside the flooding region should rebroadcast the packet. Blind flooding is not efficient as the radio coverage of nodes overlaps with each other significantly, especially in a dense network. For example, a node does not need to broadcast if all its neighbors have already received the packet from other nodes. Therefore, more smart flooding protocols can be used to replace the blind flooding inside the restricted flooding area, e.g., the probabilistic-based flooding [70], counter-based flooding [71], area-based flooding [71] or neighbor-knowledge-based flooding [72].

Note that the above geocast protocols use a multicast to forward packets to the geocast region. GeoTORA [73], instead, uses unicast routing protocol TORA [74] to deliver the packet to the geocast region, which aims to reduce the forwarding overhead of multicast broadcasting. Similar, Greedy-Forwarding-Geocast (GFG), proposed by Seada [69], uses greedy routing to forward packets. If the greedy routing fails, it switches to face routing to route around the local maximum (hole problem).

2.1.3.2 Geocast Region flooding

Once the packet enters the geocast region, it is ready to flood the packet within the geocast region. If using blind flooding, each node inside the region that receives the packet for the first time broadcasts the packet and the other nodes outside the region discard the packet. We can also use smart flooding to reduce the broadcast overhead as discussed earlier.

This flooding mechanism works very well if nodes within the geocast region are well connected. However, the flooding alone may not work if the connectivity of the geocast region is partitioned due to sparse nodes distribution or due to obstacles. For example, in Fig. 2.10, the nodes in the geocast region are partitioned

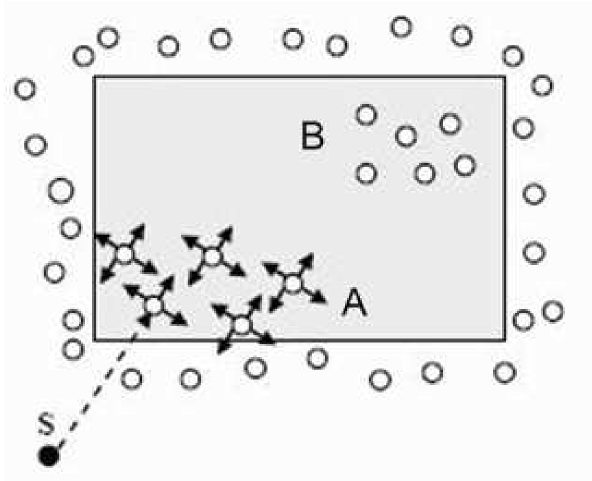


Figure 2.10: Partition within geocast region

into group M and group N. Assuming a packet enters the geocast region from a node in group M, the node then floods the packet within the geocast region. However, due to the partition, the packet could never reach the nodes in group N.

Seada in [69] proposes GFPG to address this problem. In such a situation, GFPG routes the packet through the nodes outside the geocast region that can connect the nodes in group M to the nodes in group N, as illustrated in Fig. 2.11. GFPG combines the flooding with face routing in order to guarantee the delivery of packets to all nodes within the geocast region. Discussed in more detail, this means that the nodes at the internal border of geocast region use face routing to route the packet through the nodes outside the geocast region, whereas the other nodes within the geocast region use simple broadcasting. The internal border nodes refer to the nodes that are within the geocast region but also whilst having neighbors outside the geocast region. For example, in Fig. 2.11, assuming node A is the first node inside the region that receives the geocast packet. It floods the packet within nodes in group M. Since node B has a neighbor outside the region, node B is identified as the internal border node. By applying face routing, node B forwards the packet to the nodes outside the region. If the nodes in group M and nodes in group N are connected, face routing guarantees the packet can be

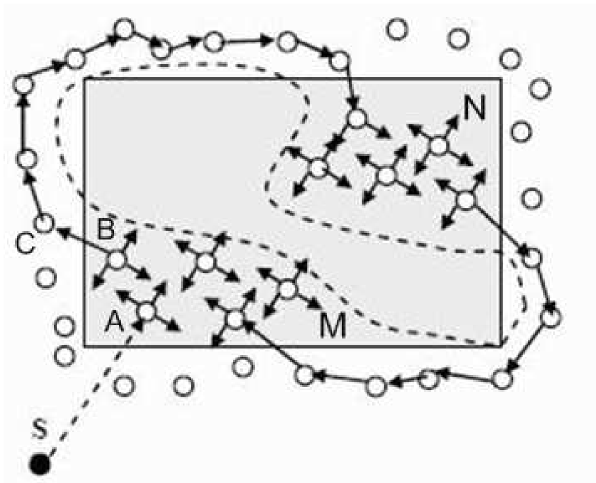


Figure 2.11: Geographic Forwarding Perimeter Geocast (GFPG)

delivered to nodes in group N. Thus all nodes within the geocast region are able to receive the packet.

Note that GFPG applies face routing at all internal border nodes, which introduces an unnecessary overhead if the network is dense and well connected (i.e. we actually do not need the nodes outside the geocast region to assist the routing). The author in [69] further proposes GFPG+ algorithm, which only applies the face routing at the border nodes if the nodes are sparsely distributed. They have shown by simulations that GFPG+ can achieve a high packet delivery rate in sparse networks and a low overhead in dense networks. Another similar scheme to GFPG was also proposed by Stojmenotic in [75], which was first to show that GFPG may not be able to guarantee packet delivery as claimed in some situations. The authors then proposed an algorithm that also combines flooding with face routing. The difference is that, in this algorithm, the external border nodes of the geocast region also perform face routing, which can also guarantee packet delivery.

2.2 Hop Count Analysis and Applications

In a multi-hop wireless ad-hoc network, a packet is retransmitted at each hop until the packet reaches its destination. Hop count, i.e., the number of hops from a source to a destination, is an important measure in the performance evaluation of multi-hop wireless networks. Hop count also has a determinative effect on the performance measures such as throughput, energy consumption, routing overhead and end-to-end delay.

The most widely used hop count estimation in current literature is by L/R, where L is the distance between a source and a destination and R is the radio range of the assumed ideal radio model. The estimation assumes that nodes can always find the next hop on the border of the radio range, and the next hop lies on the straight line connecting the source and the destination. These assumptions are valid only when the node density in the network tends to go to infinity, something that is not true for many applications. In a network with a realistic node density, L/R often underestimates the hop count of two communicating nodes. Several work have attempted to estimate hop count in more realistic scenarios. The hop count between a source and a destination depends on the routing protocol employed to find the routing path. We class the previous literature into two categories: hop count analysis for the shortest path routing and hop count analysis for greedy routing. We will introduce these two categories in the following two subsections. In the end of this section, we will also discuss the applications proposed in current literature that exploit the hop count knowledge.

2.2.1 Analysis of Shortest Path Routing

Miller et al. [76] has formulated the two-hop connection probability, i.e. the probability that communication pair is connected by the two hops shortest path. They assumed that each node has a circular radio range and the nodes are Gaussian distributed. With these assumptions, nodes are more concentrated around the center of the network, which leads to an uneven network density. The two-hop connection probability is the probability that the source and the destination are not directly connected and there is at least one relay node which directly connects with these two nodes. The authors use this geometric property to calculate the two-hop probability. For the k-hop ($k > 2$) connection probability, they provided an upper bound estimation under the assumption of an extremely dense network and the use of straight line routing. No simulations and validations have been presented in this work. A very similar work has been proposed by Bettstetter et al. [23]. The differences are that the work in [23] assumes uniformly distributed nodes and they are presented with simulation results in order to illustrate the hop count distribution. Ta and Mao et. al. [77, 78] studied the k-hop connection probability under the assumption of an ideal radio range and the Poisson distributed nodes. Their model recursively calculates the k-hop connection probability (k is any positive integer) given a source-to-destination distance. The analytical results are validated by the simulations.

Mukherjee [25] formulated the k-hop outage probability $q_k(L)$, i.e. the probability that a source and a destination separated by distance L, cannot communicate in $<= k$ hops. This model considered a more realistic radio model, i.e. the log-normal shadowing model, which introduces a random fading on top of distance-based attenuation. We should recall that in an ideal radio model, the direct link between two nodes is a binary problem, i.e., either there exists a direct link or there does not, based on the distance separating them. Under a log-normal shadowing radio model, the existence of a direct link between two nodes is a probabilistic problem. The probability depends on the distance between the two nodes and random signal fading. The radio coverage of each node in this model is irregular, which resembles realistic situations [27, 28]. Therefore, the log-normal shadowing radio model has been accepted as a realistic radio model. The analysis in Mukherjee [25] is the first one (and the only one in the literature) that has considered this realistic radio model in hop count analysis. However, Mukherjee's work only provides a non-closed form formulas for the low bound of the m-hop outage probability and the results were not validated by simulations. Further, due to the extremely high computational complexity of the analytical model, the authors only illustrated the analysis results of 1-hop outage probability and 2-hop outage probability. For k-hop ($k > 2$) outage probabilities, these were only illustrated using simulation results.

Zhao and Liang [22] used a different approach to estimate the average hop count incurred in the shortest path routing. They first analyzed the reversed problem, i.e, what the source-to-destination distance is given that the hop count is known. Let $f(D|h_i)$ be the distribution of source-destination distance given that the hop count from the source to the destination is h_i. Due to the complexity to analyze the $f(D|h_i)$, they conducted simulations and by observing the simulation results, they proposed to use an attenuated Gaussian distribution to approximate $f(D|h_i)$. Knowing the distribution $f(D|h_i)$ and a given distance d, the model enumerates all possible values of hop count, h_i, and selects the hop count that maximizes $f(D = d|h_i)$. This work has also discussed two simple applications that utilize the estimated hop count in order to calculate latency and energy consumption. We will discuss these in a later section.

2.2.2 Analysis of Greedy Routing

Kleinrock and Silvester [19] used a simple geometric calculation to approximate the average progress in one hop incurred in a MFR routing (a variant of greedy routing, see Section 2.1.1.1). The average progress measures how far a packet can progress towards the destination on average, in one hop. It is defined as the average distance from the current forwarding node to the projected point of the next hop on the line connecting the forwarding node and the destination. Knowing the average progress in one hop, referred to as *hop distance*, the average hop count given a source-to-destination distance is estimated as the ratio of the Euclidean distance to the hop distance. Similar, Haque et. al. [79] analyzed the expected hop count incurred in both compass routing and greedy routing protocol. They formulated the forwarding probability in one hop, based which the hop distance was obtained.

Swades et al [80, 21] analyzed the average hop count of greedy routing given a distance in a two-dimensional network with the assumption of an ideal radio model. They first studied the greedy forwarding behavior in one-hop, and formulated the probability distribution of the remaining distance from the next hop to the destination. The distribution of the remaining distance was then used to estimate the average progress towards the destination in each hop, referred to as λ. The average hop count given the source-to-destination distance of L can be simply estimated by L/λ. Swades et al further applied numeric simulation (Monte-Carlo) to approximate the probability distribution of hop count and the lower/upper bounds of the average hop count. However, the analytical model was not well justified by the simulations. For example, the validation for the average hop count is missing in [21].

Zorzi and Rao [81] proposed Geographic Random Forwarding (GeRaF) for the network where nodes are assumed to randomly turn on and off. In GeRaF, when a forwarding node broadcasts a data packet, all its neighbors compete against each other to relay the packet using a priority-based scheme. In this scheme, the neighbor closest to the destination has the highest priority to relay the packet. Therefore, GeRaF essentially is a greedy routing. The difference is that the node in GeRaF does not need to know its one-hop neighbor's location

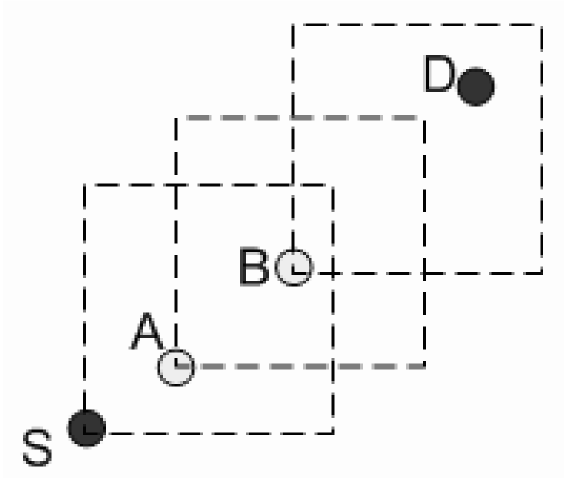


Figure 2.12: A routing scheme with square radio coverage

when making forwarding decision. The node just broadcasts the packet and let its neighbors decide if they want to forward the packet. Zorzi and Rao analyzed the hop count given a source-to-destination distance incurred in GeRaF. The mode assumed an ideal radio model and Poisson distributed nodes. They formulated a tight lower bound and upper bound of the average hop count, which have been validated by these simulations.

Lebedev and Steyaert [20] analyzed the hop count incurred in a specialized routing scheme with the assumption of square shape radio coverage. In this routing scheme, each intermediate node randomly selects the next hop from the neighbors that are within the quadrant of the radio range oriented to the destination. The routing path established in this way may have some redundant relay nodes. Therefore, the routing scheme further shorten the routing path by removing the redundant intermediate nodes. For example, In Fig. 2.12, the routing scheme initially randomly selects the routing path as S, A, B, D from the source S to the destination D. Since node B is within the radio range of node S, the source node S can directly communicate with node B. Thus, the relay node A is a redundant node and should be removed. The authors had aimed to estimate the hop count under this assumed non-realistic routing scheme.

Kuo et. al. [82] modeled the hop count for a source-destination pair in mobile ad-hoc networks. Since nodes are mobile, the distance between a source and a destination changes with the time. The work assumes that the node movement forms a growing circle centered at each node. The radius of the circle depends on the moving speed and the time interval of the movement. Based on this mobility model, the work can calculate the distance distribution between the source and the destination. However, their analysis mapping from distance to hop count assumes a very high node density, where hop distance (i.e. the progress distance per hop) equals to the radio range of node. Therefore they assumed an unrealistic straight-line routing scheme. In their successive work [83], they relaxed the assumption of high node density. The work assumed the use of greedy routing. They calculated the average progress per-hop and approximated the path connectivity probability, which is probability that a source can find a greedy routing path to the destination. Based on the per-hop progress and the path connectivity probability, they further approximated the hop count distribution. The work also demonstrated the use of the analytical results to evaluate the cost and latency of different flooding schemes in target destination discovery. However, these work still assumed the perfect circular radio coverage for each node.

The above works introduced so far focus on estimating the hop count of a communication pair, given the Euclidean distance between the source and the destination. Vural and Ekici [84, 85] studied the reversed problem. They aimed to analyze the distance distribution between a source and a destination given that the hop count is known. The authors have assumed an ideal radio model and uniformly distributed nodes. This work focused on a one-dimensional network, where the node selects the next hop based on greedy routing. They first analyzed the probability distribution of the one-hop-distance (i.e. the distance covered by one hop connection) and the multi-hop-distance (i.e. the distance covered by a multi-hop connection). Then they proposed approximations to estimate the expectation and standard deviation of single-hop-distance and multi-hop-distance. Based on the numeric results, they further showed that the multi-hop-distance distribution can be approximated by a Gaussian distribution (similar to [22]). These studies have work only discussed the possible solutions for two-dimensional networks.

In their recent work [86], Vural and Ekici extended the analysis to two-dimensional networks. They proposed a generalized greedy routing to maximum the Euclidean distance covered by multi-hop communications. At each hop, this routing scheme restricts the propagation direction outwards from the source node and greedily select the furthest neighbor as the next hop. In other words, the next hop of a forwarding node is selected from a sector area centered at the forwarding node. The work showed that the Gaussian distribution used to approximate the multi-hop distance in one-dimensional networks is no longer accurate in two-dimensional networks. Instead, they used simulations results to illustrate that a simple linear transformation of Gamma distribution can approximate the distance distribution in two-dimensional case. The work then focus on analytically computing the parameters of such Gamma distribution and the mean value of the maximum distance covered by multi-hop communications. The work assumed the circular radio coverage for each node at the physical layer.

Nath and Kumar [87] studied the proportionality between Euclidean distance of two nodes and the hop count between them. The work shows that, for an arbitrary node deployment, the proportionality between distance and hop count does not hold. However, with random nodes deployment, the source-to-destination distance is proportional to the hop count. The work formulates the asymptotic lower bound and upper bound of the distance between two nodes given that the hop count is known. This work also assumes a circular radio coverage for each node.

Dulman et al. [24] studied the relationship between the hop count and Euclidean distance. They started from one-dimensional network scenarios. This model uses a recursive approach to estimate the distribution of distance given the hop count from the source to the destination by the shortest path. Then, in the two-dimensional network scenarios, the model focuses on greedy routing and recursively calculates the hop count distribution, given the source-destination distance. These analytical models have been validated by the simulations. They also have given an example to apply the hop-distance knowledge to a DVHop localization scheme, which improves the localization error significantly. We will discuss this application in detail in the next section.

In our proposed hop count analysis, we will study greedy routing with the consideration of both the idealistic and realistic radio model (log-normal shadowing radio model). We will theoretically analyze the hop count distribution under these two radio models in a two-dimensional network where the nodes are uniformly distributed. The analysis results are validated by an extensive set of simulations including a real-world vehicular network which exhibits a realistic network topology. We will further propose approximations for both radio models to simplify the calculations. Our hop count analysis reveals that the L/R estimation may overestimate the actual hop count if random signal fading is present. Our analysis also challenges the current belief that greedy geographic routing can (approximately) find the shortest path between a source and destination. We will show that this is only true if no random fading is present. Under the presence of random fading, greedy routing can take paths significantly longer than the shortest path.

2.2.3 Applications of Hop Count Analysis

Despite that much work has been done to analyze the hop count through studies and literature, only some of it has created applications in order to utilize these hop count analysis results. Dulman et al. [24] has proposed an application that incorporates hop count knowledge in localization for wireless sensor networks. Zhao and Liang [22] has illustrated how to use hop count results to estimate end-to-end latency and energy consumption. We will discuss these applications in this section.

2.2.3.1 Localization

In wireless sensor networks, sensors often need to know their locations as the sensing data is only meaningful when incorporated with the location of data source. Equipping each sensor nodes with a positioning device is not practical in a wireless sensor network, where nodes are often dense deployed and each sensor ought to have a minimum cost. Also, in some indoor sensor network applications, the positioning devices, which rely on the line-of-sight satellite signal (e.g. GPS), do not work well due to the building obstacles. Hence, localization algorithms are proposed to address the problem of estimating the sensors' location.

Hop count based localization is one genre of localization protocols. It exploits the relationship of hop count to the Euclidean distance between two nodes. In DVHop localization [88], each sensor measures the hop counts form itself, to some reference nodes. The reference nodes, also called as anchors, are special nodes that know their own location. The sensor then estimates the Euclidean distances from itself to each anchor, based on the hop count. By knowing the distance to each anchor, the sensor can use triangulation method to estimate its own location. The algorithm is detailed as following steps.

1. Each anchor node floods a message to the network. Thus a sensor node

can learn the hop count from it to each of the anchors. Each anchor can also learn the hop count to any other anchors.

2. By knowing the hop count between two anchors and the anchors' location, DVHop can estimate the average progress distance per hop (i.e. hop distance). DVHop then distribute this information to the network.

3. Having the hop distance information and the hop count to the anchors, a sensor can estimate the Euclidean distance to each anchors. Each sensor then uses the triangulation algorithm to estimate its own location.

Note that DVHop algorithm learns the hop distance when the network is on operations. However, if the sensors are randomly deployed, we can take advantage of the statistics of the random network and compute the hop distance beforehand. By doing so, we can eliminate step 2 in above algorithm. The improved protocol DVHopSE [89, 24] adopts this ideal. DVHopSE estimates the hop distance based on the hop count analysis and utilizes this analytical result directly in the algorithm, which saves the step 2 and the corresponding overhead.

The original DVHop algorithm uses the least square method to compute the sensors' location via the triangulation algorithm, which means each anchor has an equal weight to determine a sensor's location. However, the accuracy of the estimated distance is often inversely proportional to the estimated distance itself. In practice, it is reasonable to let the anchors far from the sensor play a less determinant role in the triangulation algorithm. DVHopSE further proposes to use a weighted least square method to improve accuracy. The weight is calculated as $w_i = 1/\delta_i^2$, where δ_i is the variance of distance estimation based on the hop count i. The simulation results have shown that DVHopSE can reduce the localization error of the original DVHop scheme by around 25%.

2.2.3.2 End-to-End Delay Estimation

Zhao and Liang [22] applied the analysis results of hop count to estimate the latency from a source to a destination. The application assumes that the latency increases linearly with the hop count. The latency introduced in each hop is

$$T_{hop} = T_{tx} + T_{pr} + T_{rx} \tag{2.1}$$

where T_{tx} and T_{rx} is the time for a node to process one bit of incoming and outgoing message respectively, and T_{px} is the required time to transmit one bit of message.

Given the distance from the source to the destination, we can estimate the average hop count, $\overline{h}$, based on the analytical model. Thus, the total latency of an l-bits message from the source to the destination is

$$l \cdot \overline{h} \cdot T_{hop} \tag{2.2}$$

2.2.3.3 Energy Consumption Estimation

Zhao and Liang [22] further proposes an energy consumption model that utilizes the analyzed hop count results. The model derives an end-to-end energy consumption for sending l bits over the source-to-destination distance d. It assumes a perfect power control model as used in [35]. To transmit one bit over distance s, the sender consumes

$$\epsilon_{tx}(s) = \epsilon_{elec} + \epsilon_{amp}s^2 \tag{2.3}$$

and the receiver consumes

$$\epsilon_{rx} = \epsilon_{elec} \tag{2.4}$$

Where ϵ_{elec} is the energy dissipated to run the transmitter or receive circuitry and ϵ the energy used for transmit amplifier.

Let s be the hop distance, i.e. the distance between a sender and a receiver in each hop. The end-to-end energy consumption for sending l bits over distance d is therefore given by,

$$\begin{aligned} E_{total} &= \overline{h} \cdot E(\epsilon_{tx}(s) + \epsilon_{rx}) \\ &= \overline{h}(2\epsilon_{elec} + \epsilon_{amp}E(s^2)) \\ &= \overline{h}(2\epsilon_{elec} + \epsilon_{amp}(\overline{s}^2 + \sigma^2)) \end{aligned} \tag{2.5}$$

Where $\overline{h}$ is the average hop count given distance d, $\overline{s}$ is the average hop distance and σ is the variance of the hop distance. Those parameters can be calculated based on the statistic results, as illustrated in [22].

2.3 Summary

This chapter presents previous studies and literature most related to our proposals. We have discussed the previous hop count analytical studies and the existing applications that utilize the hop count analysis results. We also have discussed the geographic forwarding strategy, position update that aims to maintain the neighbors' location information, and geocast protocols that deliver a message to a certain geographic region.

Chapter 3

Hop Count Analysis in Greedy Routing

Hop count is a fundamental metric in multi-hop wireless ad-hoc network. It has a determinative effect on the performances of wireless network, such as throughput, end-to-end delay and energy consumption. Identifying hop count metric, including the distribution function and the mean value, is therefore vital for analyzing wireless network performance. This chapter proposes a theoretical model to accurately analyze the hop count distribution and its mean value. Given a communication pair, its hop count metric is dependent on the routing protocol selected and the network topology determined by the physical radio model. At routing layer, our model focuses on the widely used greedy routing. At physical layer, the model investigate the ideal radio model, and a more realistic radio model, e.g. log-normal shadowing model. We conduct a rich set of simulation to validate our analytical model. The comparison results show that the simulation results closely match with the analysis results. The analytical model is further validated through a trace driven simulation of a practical vehicular ad-hoc network that exhibits realistic topologies of public transport buses in a metropolitan city.

3.1 Overview of the System Model

For mathematical tractability, we make the following simplifying assumptions:

- The node distribution follows a homogenous Poisson point process with a density of ρ sensors per unit area, which can approximate uniform distribution for large area. This assumption has been widely used in analyzing multi-hop wireless ad-hoc networks [14, 15, 16].

- No Boundary: In a typical ad hoc deployment, nodes located near the network boundary have fewer neighbors that nodes located elsewhere. To avoid this distinction, we ignore the existence of the boundary. Consequently, the probability distribution function for the number of neighbors at each node is identical [93].

- All nodes have identical transceivers and the wireless links are assumed to be symmetric.

- Complete Knowledge of Local Topology: We assume that nodes always have an up-to-date view of their local topology, i.e. each node is aware of the locations of its immediate neighbors. The nodes can employ a neighbor discovery protocol for this purpose. Consequently, each intermediate node can always find the optimal next hop.

- The network is dense enough such that the greedy routing always succeeds in finding a next hop node that advances the data packet towards the sink. In other words, we assume that the forwarding strategy does not encounter a *local minima condition* and thus, neglect the effect of planar routing, which is employed in these circumstances.

The above assumptions, some of which are somewhat unrealistic, are necessary in making the analysis tractable. However, in our simulation study, we relax several of these assumptions (e.g: uniform distribution of the nodes) to create more realistic scenarios and compare the resulting outcomes to those from our analysis.

In the first part of our analysis we consider an ideal radio model, wherein the signal attenuation between any two nodes is a function of the Euclidean distance separating the nodes. Consequently, in this idealistic environment, the radio coverage of a sensor node is a perfect circular disc with the radius equal to its radio range. However, in reality, the signal attenuation is not solely dependent on the distance. For example, signal reflection or signal noise can also attenuate the signal. To make our analytical results more realistic, we extend our analysis and incorporate the log-normal shadowing radio model. This model adds a random signal loss component to the purely distance-dependent signal attenuation. As will be elaborated later, we have observed significant differences in the analytical results with the two models. Note that, by employing these two radio models, we implicitly assume that signal attenuation over different link are independent. For the sake of mathematical tractability, we do not consider signal correlation among different links. This link independent log-normal shadowing model has been widely used to approximate the real environment [14, 15, 16].

Assuming that the distance between the source and destination is known, our analysis seeks to develop a model for analyzing the hop count from the source to the destination incurred in greedy routing. We use a discrete Markov chain to model the hop-by-hop progress of a packet from the source to the destination. The state of the Markov chain is defined as the Euclidean distance (measured in some consistent metric unit, e.g. meters) between the current forwarding node that holds the packet and the destination. Ideally, this distance should be mod-

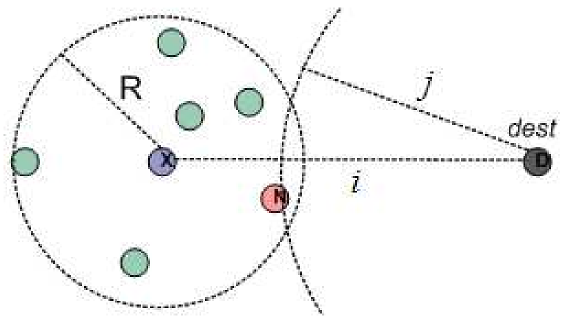


Figure 3.1: Example of state transition (from state i to state j)

eled as a continuous random variable. However, to simplify our model, we use a discrete state space to approximately represent the continuous distance values. We quantize the distances resulting in a state space of $(0, \varepsilon, 2\varepsilon, ..., n\varepsilon, ...)$, where the parameter ε is the interval of the state space (i.e. the quantization coefficient). When the interval ε is small enough, the discrete state space approximates the original continuous distance metric.

We elaborate on the state transition of the Markov chain using the example illustrated in Fig. 3.1, Assume that a packet is currently held by node X as it makes its way towards the destination, node D. Since node X is at a distance i from the destination, the current state for this packet is i. Assume that the next hop node chosen by node X using greedy forwarding is node N, which is at a distance of j from the destination. The packet forwarding operation thus results in a state transition from i to j for the packet. In general, the hop-by-hop progress made by a packet towards the destination can be represented by a series of states that the packet transitions through, eventually culminating in state 0 when the packet reaches the destination.

Our analysis is composed of the following steps. The first step involves determining the state transition probabilities for the Markov chain (section 3.2) using geometric calculation assuming the ideal circular disc radio model. Next we extend this to include the log-normal shadowing model (Section 3.3). Based on the transition probabilities, we recursively compute the hop count distribution and the mean value given a communication pair (Section 3.4). The main symbols used in the work are listed in Table 3.1.

Table 3.1: List of main symbols used in the analysis

Symbol	Definition
R	Average radio range of sensors
ρ	Node density, i.e. the number of nodes per unit area
ε	Quantization interval
i, j, d	Euclidean distance from a sensor to the destination
ξ	Signal randomness parameter in log-normal shadowing radio model
$P_\wedge(s)$	The probability that two nodes separated by distance s can communicate with each other
$P_{i,j}$	State transition probability. i.e. the probability that a sensor at distance i from the sink can forward its packets to the sensor at distance j

3.2 Evaluating the State Transition Probability for the Ideal Radio Model

For the ideal radio model, a node can only communicate with other nodes that are located within the circular coverage region of this node. Let R be the radio range of each nodes. For two nodes separated by a distance s, the probability that they have a direct link, denoted as $P_\wedge(s)$, is

$$P_\wedge(s) = \begin{cases} 1 & \text{if } i \leq R, \\ 0 & \text{if } i > R. \end{cases} \tag{3.1}$$

We employ an approach that uses geometric computations and probability theory to prove the following,

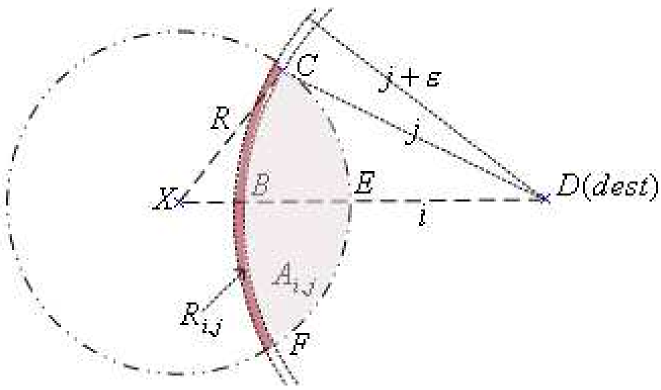


Figure 3.2: Illustration used to prove Theorem 1.1

Theorem 1. *In the context of an ideal radio model, the transition probability of a packet from state i to j when employing greedy routing is,*

$$P_{i,j} = \begin{cases} 1 & \textit{if } i \leq R \textit{ and } j = 0, \\ \exp(-\rho A_{i,j}) - \exp(-\rho A_{i,j+\varepsilon}) & \textit{if } i > R \textit{ and } i - R \leq j < i, \\ 0 & \textit{others}, \end{cases} \tag{3.2}$$

where $A_{i,j}$ *is,*

$$A_{i,j} = R^2 \arccos \tfrac{i^2+R^2-j^2}{2iR} + j^2 \arccos \tfrac{i^2+j^2-R^2}{2ij} - \tfrac{\sqrt{(R+i+j)(R+i-j)(R-i+j)(i+j-R)}}{2} \tag{3.3}$$

Proof:

Assume that a packet is currently at node X as it makes its way towards the destination. Let node X be at a distance i from the destination as illustrated in Fig. 3.2. Consequently the packet is currently in state i. The probability that the packet is forwarded to a sensor at distance j and thus resulting in a transition to state j is the probability that node X finds a neighbor at distance j as the next hop.

We start with a simple case, where $i \leq R$, i.e. the destination node is within radio coverage of the current node X. Hence, as the next hop is the destination, the state i must transition to state 0. Consequently, we have,

$$P_{i,j} = \begin{cases} 1 & \text{if } i \leq R \text{ and } j = 0, \\ 0 & \text{if } i \leq R \text{ and } j > 0. \end{cases} \tag{3.4}$$

Now let us consider the situation where $i > R$. Recall that, we have assumed that greedy routing can always succeed in finding a next hop node which is closer to the destination. Thus the next hop of node X must has a distance that is less than i from the destination. In other words, the probability that the next hop node lies outside distance region $[i - R, i)$, is zero. Therefore, we have,

$$P_{i,j} = 0, \qquad \text{if } i > R \text{ and } (j < i - R \text{ or } j \geq i) \tag{3.5}$$

Now, we discuss the more complicated and plausible case where, $i - R \leq j < i$. In greedy routing, if the next hop of node X is at distance j, it implies that at least one neighbor of node X is at distance j and none of its other neighbors are closer to the destination than j. Thus the transition probability is the probability that at least one neighbor of node X lies on the perimeter of the curve of radius j centered at the destination (see Fig. 3.2) with no neighbors located to the right of this curve. Since we assume a discrete state space with ε as the interval of the state space, we can approximate the curve as a ring of thickness ε, as illustrated in Fig. 2. Let $R_{i,j}$ represent the region of this thin ring that intersects with the radio range of node X (narrow dark region in Fig. 3.2). We also denote $A_{i,j}$ as the area of the light shaded region in Fig. 3.2, which is the intersecting region between the radio coverage of node X and a circle of radius j centered at the destination. $R_{i,j}$ and $A_{i,j}$ are also used to represent the area of each region referred.

Now, the transition probability $P_{i,j}$ is the probability that at least one node lies inside region $R_{i,j}$ and the no nodes are within $A_{i,j}$. Let P_1 be the probability that at least one node is within $R_{i,j}$, and P_2 be the probability that no nodes lie within $A_{i,j}$.

Recall that, we have assumed that the node distribution follows a homogenous Poisson point process with density ρ. As a property of this assumption, the number of nodes in any region of area A follows a Poisson distribution with mean of ρA. Thus the number of nodes in region $R_{i,j}$ follows a Poisson distribution with mean $\rho R_{i,j}$, and the number of nodes in region $A_{i,j}$ has a Poisson distribution with mean of $\rho A_{i,j}$. Note that, the area of $R_{i,j}$ can be computed as $A_{i,j+\varepsilon} - A_{i,j}$. Consequently, we have,

$$P_1 = 1 - \text{Prob}(\text{no node in } R_{i,j}) = 1 - \exp(-\rho R_{i,j}) = 1 - \exp(\rho A_{i,j} - \rho A_{i,j+\varepsilon}) \quad (3.6)$$

$$P_2 = \text{Prob}(\text{no node in } A_{i,j}) = \exp(-\rho A_{i,j}) \quad (3.7)$$

In the Poisson point process, the distribution of the number of nodes in any two disjoint region is independent. Thus P_1 and P_2 are independent and we have,

$$P_{i,j} = P_1 \cdot P_2 = \exp(-\rho A_{i,j}) - \exp(-\rho A_{i,j+\varepsilon}) \quad (3.8)$$

Now we come to compute the area of $A_{i,j}$. As shown in Fig. 3.2, the area $A_{i,j}$ can be computed as,

$$A_{i,j} = 2(A_{\widehat{CXE}} + A_{\widehat{CDB}} - A_{CXD}) \quad (3.9)$$

where $A_{\widehat{CXE}}$ is the area of the sector CXE; A_{CXD} is the area of triangle CXD and $A_{\widehat{CDB}}$ is the area of sector CDB. By applying the law of cosines and Heron's formula, we have,

$$\begin{cases} A_{\widehat{CXE}} = \frac{R^2}{2}\angle CXD = \frac{R^2}{2}\arccos\frac{i^2+R^2-j^2}{2iR} \\ A_{\widehat{CDB}} = \frac{R^2}{2}\angle CDX = \frac{j^2}{2}\arccos\frac{i^2+j^2-R^2}{2iR} \\ A_{CXD} = \frac{\sqrt{(R+i+j)(R+i-j)(R-i+j)(i+j-R)}}{4} \end{cases} \quad (3.10)$$

Combining Equations (3.9) and (3.10), we obtain Equation (3.3). Finally, combining Equations (3.4), (3.5), (3.8) and (3.3), the theorem is proved. ♠

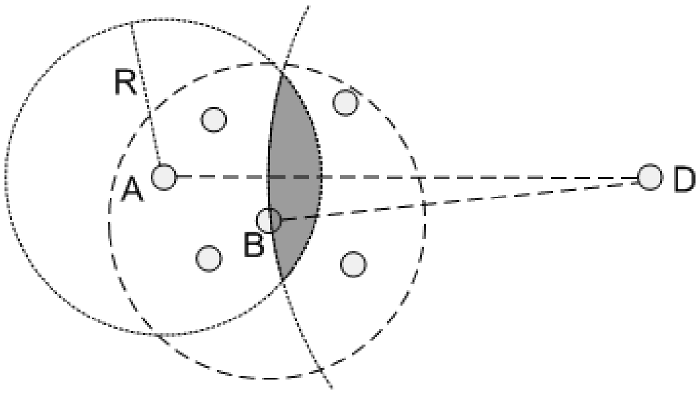


Figure 3.3: Illustration of forwarding dependency

Recall that, the transition probability $P_{i,j}$ is the probability that a node at distance i (from the destination) forwards its packets to a neighbor at distance j. At each hop, we use $P_{i,j}$ to calculate the forwarding probability from the current forwarding node to its neighbors. In Theorem. 1, we actually implicitly assume that the forwarding probability at each hop is independent of each other. However, in reality, the forwarding probability at a node may depend on the preceding hop. For example, in Fig. 3.3, node A forwards its packets to node B since B is the closet neighbor to the destination, i.e. there is no nodes existing in the shaded area. Assume that now its B's turn to make forwarding decision. Since there is no nodes within the shaded area, there is no way that B can forward its packets to the shaded area. However, in our analysis of forwarding probability $P_{i,j}$, we assume the nodes distribution seen by node B is independent with the nodes distribution seen by the preceding node A. Therefore, in the analysis model, we assume that B can still possibly find nodes in the shaded area and forwards its packets to them, which is not true in this example. Despite this discrepancy, our analysis can still hold as a close approximation and the simulations results (Section 3.5) well justify this approximation.

3.3 Evaluating the State Transition Probability for the Log-normal Shadowing Radio Model

Next, we study a more realistic radio model. In the log-normal shadowing radio model, the signal attenuation between two nodes is dependent not only on the distance separating the two nodes, but also a random signal loss. More formally, given a distance s that separates two nodes, the signal attenuation (in dB) from one node to another one is,

$$\beta(s) = \alpha \log_{10}(\frac{s}{\text{reference distance}}) + \beta_1 \tag{3.11}$$

where α is a path loss rate, and β_1 is a random variable that follows a normal distribution with zero mean and a standard deviation of σ,

$$f(\beta_1) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{\beta_1^2}{2\sigma^2}) \tag{3.12}$$

Now two nodes are one-hop neighbors, i.e. they have a direct link between them, only if the signal attenuation between them is less than or equal to a predefined attenuation threshold β_{th}. Thus, for two nodes separated by a distance s, the probability that they have a direct link, denoted as $P_\wedge(s)$, is given by,

$$P_\wedge(s) = Prob(\beta(s) < \beta_{th}) \tag{3.13}$$

The above equation has been solved by Bettstetter in [90] and the result can be represented by,

$$P_\wedge(s) = \frac{1}{2}\left[1 - erf(\frac{10}{\sqrt{2}\xi} log_{10}\frac{s}{R})\right], \quad \xi = \sigma/\alpha \tag{3.14}$$

where $R = 10^{\frac{\beta_{th}}{\alpha \cdot 10}}$, is referred to as the **average radio range**, which is the maximum distance that permits the existence of a link between two nodes in the

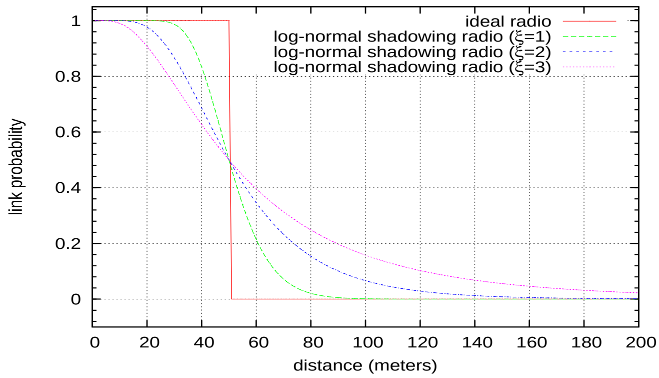


Figure 3.4: Link probability with different radio models ($R = 50$)

absence of signal randomness. The function $erf(.)$ is defined as follows,

$$erf(z) = \frac{2}{\sqrt{\pi}} \int_0^z \exp(-x^2)dx \tag{3.15}$$

As an illustrative example, Fig. 3.4 plots the link probability for the log-normal shadowing model for $R = 50m$ and different values of the random parameter ξ. Note that, the curve has a longer tail for increasing values of ξ, which implies that a node's radio may cover a larger area for larger ξ. Based on the aforementioned characteristics of the log-normal model, we can have the following theorem,

Theorem 2. *In the context of the log-normal shadowing radio model, the transition probability of a packet state i to j when employing greedy routing is,*

$$P_{i,j} = \begin{cases} P_{\wedge}(i) & \text{if } j = 0 \text{ and } i > 0, \\ 0 & \text{if } j > 0 \text{ and } j \geq i, \\ \left[1 - P_{\wedge}(i)\right] \cdot \exp(-\pi\rho j^2 P_{\wedge}(A_{i,j})) \cdot & \\ \qquad \left[1 - \exp(-\pi\rho\varepsilon(2j+\varepsilon)\frac{(j+\varepsilon)^2 P_{\wedge}(A_{i,j+\varepsilon}) - j^2 P_{\wedge}(A_{i,j})}{(2j+\varepsilon)\varepsilon})\right] & \text{others,} \end{cases} \tag{3.16}$$

Where $P_{\wedge}(i)$ is defined in equation (3.14), and

$$P_{\wedge}(A_{i,j}) = \int_{i-j}^{i+j} P_{\wedge}(s) f_{i,j}(s) ds \tag{3.17}$$

$$f_{i,j}(s) = \frac{1}{\pi j^2}(j^2\theta' + 2s\phi - i\sin(\phi) + \frac{s^2+j^2-i^2}{2}\phi') \tag{3.18}$$

$$\begin{aligned} \theta' &= \tfrac{2s}{\sqrt{4i^2j^2-(i^2+j^2-s^2)^2}} \qquad \phi = \arccos(\tfrac{s^2+i^2-j^2}{2is}) \\ \phi' &= \tfrac{is}{i\sqrt{4i^2s^2-(s^2+i^2-j^2)^2}}(\tfrac{i^2-j^2}{s^2} - 1) \end{aligned} \tag{3.19}$$

proof:

We start with the simple case when $j = 0$ and $i > 0$. Assume that a packet is currently in state i, while located at a certain node X. The transition probability of the packet from state i to zero is the probability that there is a direct link between node X and the destination. Thus we have $P_{i,j} = P_\wedge(i)$ when $j = 0$ and $i > 0$.

Now let us consider the situation where $j > 0$ and $j \geq i$. Since the next hop of node X must has a distance that is less than i from the destination, the probability that the next hop node lies outside distance region $[0, i)$, is zero. Therefore, we have $P_{i,j} = 0$, when $j > 0$ and $j \geq i$.

In other cases where the next state $j > 0$ and $j < i$, the transition probability $P_{i,j}$ is the multiplication of the following three independent probabilities,

- The probability that no direct link exists between node X and the destination (otherwise the packet can be forwarded to the destination directly), which is $1 - P_\wedge(i)$.
- the probability that the node X can find at least one neighbor at distance j, denoted as $1-P_1$, where P_1 is the probability that there is no neighbor at distance j.
- The probability that no neighbor is within the region that is closer to the destination than j, which is denoted as P_2.

Thus, we have,

$$P_{i,j} = \big[1 - P_\wedge(i)\big](1 - P_1)P_2, \quad \text{if } j > 0 \text{ and } j < i \tag{3.20}$$

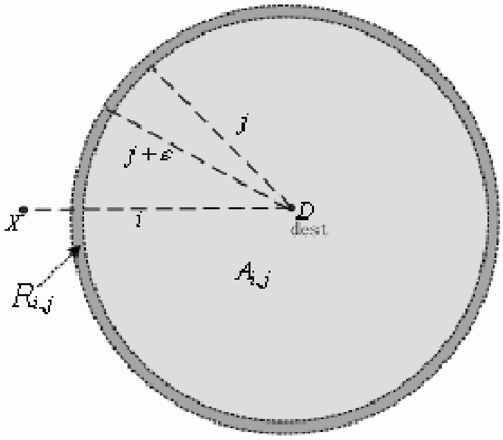


Figure 3.5: Illustration used to prove Theorem 1.2

Similar to the previous sub-section, we use a ring of thickness ε to represent the curve that is located at distance j from the destination, as illustrated in Fig. 3.5. Let $R_{i,j}$ denote the ring area that has distance j to the destination, and Let $A_{i,j}$ represent the shaded disc shaped region that is closer to the destination than j but does not include the location of the destination itself. Note that, the area included under $R_{i,j}$ and $A_{i,j}$ with the log-normal model is much larger as compared with the ideal radio model in section 3.2. The reason being that with the realistic log-normal model the one-hop neighbors of X can located anywhere in the network. On the contrary, in the case of the ideal radio model, the one-hop neighbors are restricted in the circular coverage area of node X. Thus, P_1 is the probability that no direct link exists between X and any node in region $R_{i,j}$, and P_2 is the probability that no direct link exists between X and any node in region $A_{i,j}$.

We first calculate P_1. Since the number of nodes within $R_{i,j}$ is a random variable, according to the law of total probability, we have,

$$\begin{aligned} P_1 = \sum_{k=0}^{\infty} \Big\{ & Prob(\text{k nodes in } R_{i,j}) \cdot \\ & Prob(\text{no direct link from X to any one of those k nodes}) \Big\} \end{aligned} \tag{3.21}$$

According to the Poisson point process, the number of nodes within $R_{i,j}$ has a Poisson distribution with mean $\rho R_{i,j}$. Also, given that there are k nodes in area $R_{i,j}$, these k nodes are independently distributed [94] [95]. Therefore the existence of a direct link between node X and every node in area $R_{i,j}$ is independent of each other. Let $P_{\wedge}(R_{i,j})$ be the probability that there exists a direct link between node X and a node within area $R_{i,j}$. Equation (3.21) can be rewritten as,

$$\begin{aligned} P_1 &= \textstyle\sum_{k=0}^{\infty} \frac{(\rho R_{i,j})^k}{k!} \exp(-\rho R_{i,j})(1 - P_{\wedge}(R_{i,j}))^k = \exp(-\rho R_{i,j} P_{\wedge}(R_{i,j})) \\ &= \exp(-\pi\rho\varepsilon(2j+\varepsilon)P_{\wedge}(R_{i,j})) \end{aligned} \tag{3.22}$$

Similar, for P_2, we can have,

$$P_2 = \exp(-\rho A_{i,j} P_{\wedge}(A_{i,j})) = \exp(-\pi\rho j^2 P_{\wedge}(A_{i,j})) \tag{3.23}$$

Combining Equations (3.20) (3.22) and (3.23), we have,

$$P_{i,j} = \big[1 - P_{\wedge}(i)\big] \cdot \exp\big[-\pi\rho\varepsilon(2j+\varepsilon)P_{\wedge}(R_{i,j})\big] \cdot \exp\big[-\pi\rho j^2 P_{\wedge}(A_{i,j})\big] \tag{3.24}$$

Next we compute $P_{\wedge}(R_{i,j})$ and $P_{\wedge}(A_{i,j})$. According to the definition of $A_{i,j}$, the combined region of $R_{i,j}$ and $A_{i,j}$ can be represented by $A_{i,j+\varepsilon}$. Therefore $P_{\wedge}(R_{i,j})$ can be represented by $A_{i,j}$ and $A_{i,j+\varepsilon}$. In the Poisson point process distribution, given that a node is present within $A_{i,j+\varepsilon}$, the node is uniformly distributed in the region and it is either inside region $R_{i,j}$ or $A_{i,j}$. By the law of total probability, we have,

$$\begin{aligned} P_{\wedge}(A_{i,j+\varepsilon}) &= P_{\wedge}(R_{i,j})Prob(\text{the node is within } R_{i,j}) \\ &= +P_{\wedge}(A_{i,j})Prob(\text{the node is within } A_{i,j}) \\ &= P_{\wedge}(R_{i,j})\frac{(2j+\varepsilon)\varepsilon}{(j+\varepsilon)^2} + P_{\wedge}(A_{i,j})\frac{j^2}{(j+\varepsilon)^2} \end{aligned} \tag{3.25}$$

Equivalently, we have,

$$P_{\wedge}(R_{i,j}) = \frac{(j+\varepsilon)^2 P_{\wedge}(A_{i,j+\varepsilon}) - j^2 P_{\wedge}(A_{i,j})}{(2j+\varepsilon)\varepsilon} \tag{3.26}$$

Combining Equations (3.24) and (3.26), we have,

$$\begin{aligned} &P_{i,j} = \big[1 - P_{\wedge}(i)\big] \cdot \exp(-\pi\rho j^2 P_{\wedge}(A_{i,j})) \cdot \\ &\big[1 - \exp(-\pi\rho\varepsilon(2j+\varepsilon)\tfrac{(j+\varepsilon)^2 P_{\wedge}(A_{i,j+\varepsilon}) - j^2 P_{\wedge}(A_{i,j})}{(2j+\varepsilon)\varepsilon})\big] \end{aligned} \tag{3.27}$$

Finally, we compute the last unknown variable $P_\wedge(A_{i,j})$, i.e., given there exists a node within region $A_{i,j}$, the probability that this node has a direct link with node X. Let $f_{i,j}(s)$ represent the probability that this node is located at distance s from the node X. Based on the law of total probability, we have,

$$P_\wedge(A_{i,j}) = \int_{i-j}^{i+j} P_\wedge(s) f_{i,j}(s) ds \tag{3.28}$$

Given a node is within region $A_{i,j}$, the distance from this node to node X varies from $(i - j)$ to $(i + j)$. Thus the integral in Equation (3.28) represents the conditional probability. Following rigorous geometric calculations, $f_{i,j}(s)$ is computed as indicated in Equation (3.18). The detailed derivation is omitted here due to the limited space.

Finally, combining Equations (3.27), (3.28) and (3.18), the theorem is proved.

♠

We now provide an example to illustrate the state transition probability, $P_{i,j}$. In this example, we assume the following set of parameters, $R = 50m, \varepsilon = 1m, \rho = 0.0019$, the current state of a packet is $i = 100$ and the next state varies from 100 to 0. Fig. 3.6 illustrates the distribution of the transition probability from state i to the next state j for both radio models under consideration. Note that, the log-normal model reduces to the ideal circular coverage model when the random parameter ξ is equal to zero. For the ideal radio model the peak of the distribution is around $j = 57$ and it reduces to zero for all states beyond 50. This is because of the circular coverage assumption (recall that $R = 50m$). With the more realistic log-normal model the distribution is more spread out over the entire range and the peak shifts towards the right, i.e., closer to the destination. This effect is more pronounced as the random parameter ξ increases. This is because higher the randomness in the signal, the greater is the chance that a node closer to the destination is chosen as the next hop.

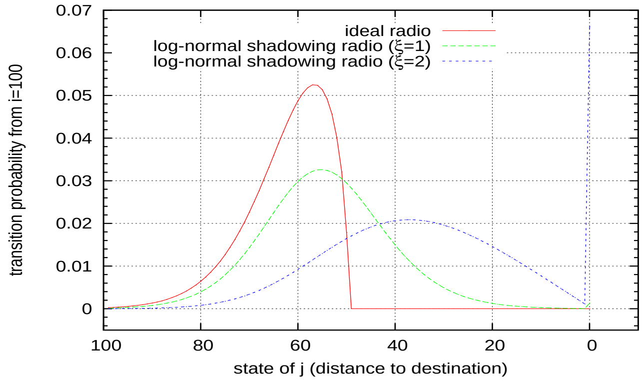


Figure 3.6: State transition probability from $i = 100$ ($R = 50$)

3.4 Hop Count Distribution and the Mean value

Based on the state transition probability $P_{i,j}$, we now proceed to derive the hop count distribution and the mean value of a communication pair. We also propose some approximations to simplify the mean hop count calculations in this section.

3.4.1 Hop Count Distribution

The analysis is independent of the radio model under consideration. One simply has to substitute the appropriate state transition probability equations as derived in the previous two sections for the radio model under consideration.

Recall that, the state variable in our Markov model represents the distance between the current node and the destination. Based on the transition probability computed in previous two sections and using the approach of recursive computation, we obtain the probability distribution function of the hop count as follows,

Theorem 3. *Given a source and destination separated by distance i, the proba-*

bility distribution of hop count (denoted as H) in greedy routing is given by,

$$P(H = h|D = i) = \begin{cases} P_{\wedge}(i) & \text{if } h = 1, \\ \sum_{j \in (i,0)} P(H = h-1|D = j)P_{i,j} & \text{if } h > 1, \end{cases} \tag{3.29}$$

where $P_{\wedge}(i)$ and $P_{i,j}$ denote the link probability and the state transition probability for the radio model under consideration. For ideal radio model, $P_{\wedge}(i)$ and $P_{i,j}$ are derived in Equation (3.1) and Theroem 1. For log-normal shadowing radio model, $P_{\wedge}(i)$ and $P_{i,j}$ are derived in Equation (3.14) and Theroem 2.

Proof:

When $h = 1$, the probability that the source is one hop away from the destination is the probability that they have a direct link. Thus,

$$P(H = h|D = i) = P_{\wedge}(i), \qquad \text{if } h = 1 \tag{3.30}$$

For the other cases, we can apply the recurrence computation. As illustrated in Fig. 3.7, the possible next hop states j originating at i are constrained between i and 0, and each subsequent step in the state space is separated by ε. If the hop count from the current state i to the destination is h, the hop count from the next state j to the destination must be $h-1$. By applying the law of total probability, we have,

$$P(H = h|D = i) = \sum_{j \in (i,0)} P(H = h-1|D = j)P_{i,j} \tag{3.31}$$

Combining Eqs. (3.30) and (3.31), the theorem is proved. ♠

Using Eq. (3.30), one can readily determine the probability of one hop. Subsequently, using Eq. (3.31) and the probability of one hop, the probability of two hops can be computed. Similarly, employing recursive computations, the probability of all h hops can be computed.

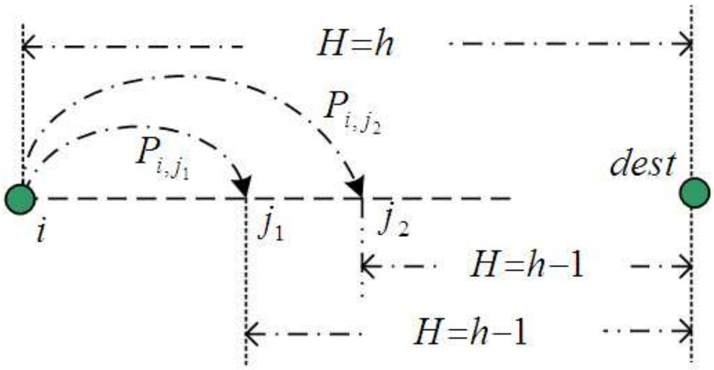


Figure 3.7: Computation of the hop count probability distribution

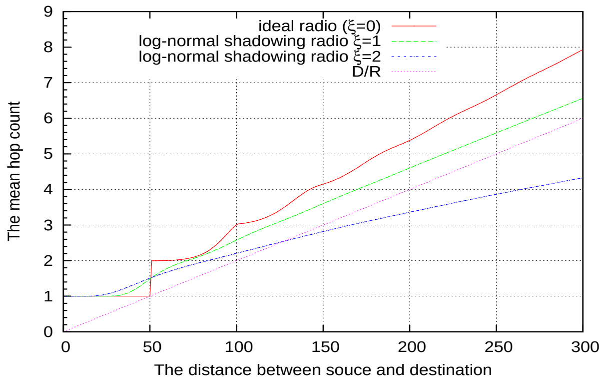


Figure 3.8: Mean hop count with varying distance and radio model

3.4.2 Mean Hop Count

Based on the hop count distribution result of Theorem 3, we can easily calculate the mean hop count for a pair of communication nodes that are separated by distance i. Let $\overline{H(i)}$ represent the mean hop count given i. $\overline{H(i)}$ can be computed as follows,

$$\overline{H(i)} = \sum_{h=1}^{\infty} h \cdot P(H = h | s = i) \tag{3.32}$$

Now we illustrate the mean hop count for Cons $R = 50m, \varepsilon = 1m, \rho = 0.0019$, the same parameters as used in Fig. 3.6. Fig. 3.8 plots the mean hop count as a function of the distance between the source and destination for different radio models. One can readily observe from Fig. 3.8 that the mean hop count is approximately a linear function of the distance. This observation implies that the ratio of the source-destination distance to the mean hop count is approximately constant. In addition, the slope of the linear line is decreasing when the ξ is increasing. Therefore, given a same source-to-destination distance, a packet in a network with bigger ξ takes less number of hops to reach the destination. Fig. 3.8 also compare the analysis results with the widely used estimation, i.e. the ratio of source-to-destination Euclidean distance to the radio range. It shows that the widely used estimation under-estimates the mean hop count in ideal radio model, while may over-estimate the mean hop count in realistic radio model when a large random fading presents.

Note that, the above computation of the mean hop count requires us to recursively compute the hop count distribution of the hop count in entire distance space. This computation has a time complexity of $O(i^3)$. It is evident that evaluating the mean hop count for a sizable network can be an considerably computationally intensive task. Hence, in the next subsection, we evaluate an $O(1)$ technique for estimating the mean hop count. We also demonstrate that our estimate is quite accurate, especially when $L >> R$.

3.4.3 Approximation of Mean Hop Count

As discussed in the previous section, the exact computation of the mean hop count can be highly complex for a communication pair with a large distance. To easy this calculation, we develop a simpler approach to estimate the mean hop count in such situations (i.e. large network). In order to achieve this, we introduce a new parameter, known as the average progress of one hop, which measures how far the packet can progress towards the destination in one hop. Given a packet at state i, its average progress of next hop, $Q(i)$, can be computed from state transition probability $P_{i,j}$. We have,

$$Q(i) = \sum_{j \in (i,0)} (i - j) P_{i,j} \tag{3.33}$$

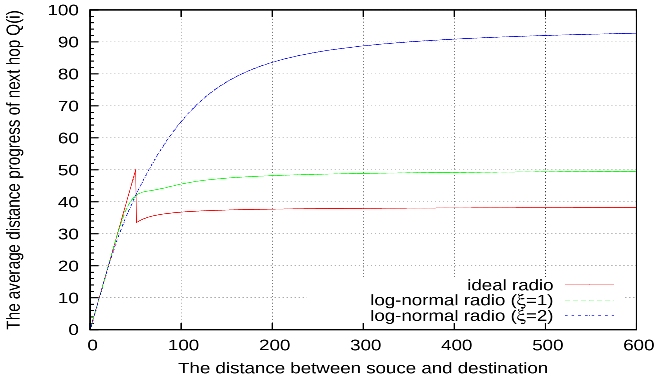


Figure 3.9: Average distance progress of next hop given current distance to the destination

The example of average progress is illustrated in Fig. 3.9. It shows that $Q(i)$ increases and converges to a certain value with the distance approaching to infinite. However, the increasing rate and convergence value is different for each radio models. Let λ be the converged value for the average progress in a particular radio model. Clearly, λ is an upper bound of average progress $Q(i)$ for any i in that radio model. Let $\widetilde{H(i)}$ denote the estimated mean hop count of a communication pair separated by distance i. If we use λ to represent the average progress for each hop from distance i to the destination, we can easily estimate the mean hop count as,

$$\widetilde{H(i)} = \frac{i}{\lambda} \tag{3.34}$$

Since the average progress $Q(i)$ is closer to the convergence value of λ with the increase of i, we expect that this estimation can accurately approximate the actual hop count for a large i. Therefore, if we can estimate λ, we can reduce the time complexity of hop count computation to O(1) in a large network. In the rest of this section, we theoretically analyze the convergence value of λ. We first define the value of λ for ideal radio model and then extend it for log-normal shadowing radio model. In the end of this section, we illustrate the performance of the simplified estimation by comparing the estimated results to the original analysis results of hop count.

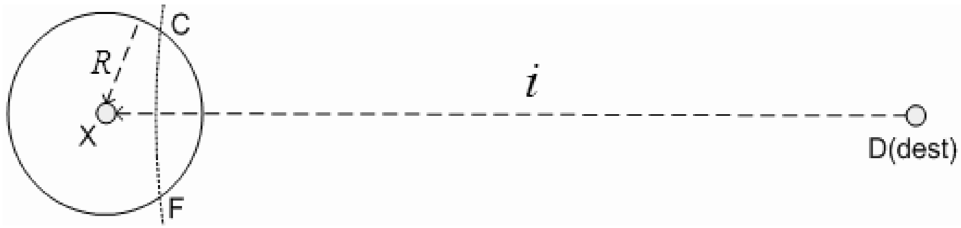


Figure 3.10: As $i >> R$, the arc $\overset{\frown}{CF}$ can be approximated as line CF

Theorem 4. *In ideal radio mode, the average progress of next hop at distance i converges as $i >> R$ and the value λ that it converges to is given by,*

$$\lambda = R\left[1 - \int_0^1 \exp(-\rho R^2(\arccos{(t)} - t\sqrt{1-t^2})dt)\right] \tag{3.35}$$

Proof:

From Theorem 1, we know that the transition probability $P_{i,j}$ is dependent on the area of $A_{i,j}$. Fig. 3.2 illustrates that $A_{i,j}$ is determined by the shape of curve $\overset{\frown}{CF}$. This area is computed using Eq. (3.9) and depends on both i and j. If the distance i between the source and destination is very large, as depicted in Fig. 3.10, the curve $\overset{\frown}{CF}$ can be approximated by a straight line. Consequently, this simplifies the computation of the area $A_{i,j}$, which now solely depends on the distance between node X and its next one-hop neighbor, $i - j$. Let x represent $i - j$. Now, we can calculate the area of $A_{i,j}$, as depicted in Fig. 3.11, as follows,

$$A_{i,j} = 2(A_{\widehat{CXE}} - A_{CXO}) = R^2 \arccos{\frac{x}{R}} - x\sqrt{R^2 - x^2} \tag{3.36}$$

In order that the progress made along the next hop from node X towards the destination (i.e. change of state from i to j) is less than x, all neighbors of X must reside in the region to the left of CF. In other words, there is no nodes in region $A_{i,j}$. Let T represent the progress made along this hop towards the destination. The probability that the progress of the next hop towards to destination is less

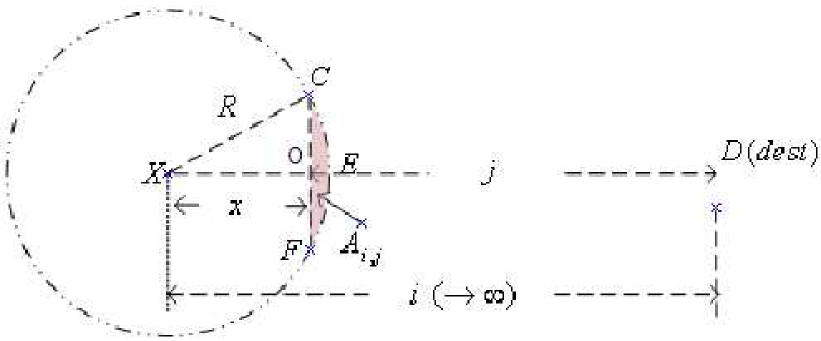


Figure 3.11: As i tends to infinity, $A_{i,j}$ is approximated by the shaded region

than x is,

$$\begin{aligned} F_T(x) &= P(T < x) \\ &= P(\text{no nodes in region } A_{i,j}) \\ &= \exp(-\rho A_{i,j}) \\ &= \exp(-\rho(R^2 \arccos \tfrac{x}{R} - x\sqrt{R^2 - x^2})) \end{aligned} \tag{3.37}$$

Consequently, the probability density function (pdf) of the progression T is given by,

$$f_T(x) = \frac{d}{dx} F_T(x) \tag{3.38}$$

Further, the average of the progression T is,

$$E(T) = \int_{-R}^{R} x f_T(x) dx \tag{3.39}$$

Recall that λ denotes the converged average progress. Thus, we have,

$$\begin{aligned} \lambda = \quad & E(T) = \int_0^R x f_T(x) dx \\ = \quad & \int_0^R x dF_T(x) \\ = \quad & \big[x F_T(x)\big]_{-R}^{R} - \int_0^R F_T(x) dx \\ = \quad & R - \int_0^R \exp(-\rho(R^2 \arccos \frac{x}{R} - x\sqrt{R^2 - x^2}))dx \\ \overset{x=Rt}{\Longrightarrow} \quad & R\left[1 - \int_0^1 \exp(-\rho R^2(\arccos(t) - t\sqrt{1 - t^2}))dt\right] \end{aligned} \tag{3.40}$$

Hence theorem 4 is proved. ♠

Now we proceed to analyze the λ for log-normal shadowing radio model where the signal randomness presents. Using similar approach, we have,

Theorem 5. *Under the consideration of log-normal shadowing radio model, the average progression of next hop at distance i converges as $i >> R$ and the value λ that it converges to is given by,*

$$\lambda = R'\left(1 - \int_0^1 \exp(-\rho R^2(\arccos(t) - t\sqrt{1-t^2})) \cdot g(t)dt\right) \tag{3.41}$$

where R' satisfies $P_\wedge(R') = \alpha$ (α is a very small decimal, e.g. 0.01), and $g(t)$ is,

$$\begin{aligned} g(t) &= \frac{10}{\sqrt{2\pi}\ln(10) \cdot \xi(\arccos t - t\sqrt{1-t^2})} \cdot \\ &\int_t^1 \frac{u^2 \arccos\frac{t}{u} - t\sqrt{u^2-t^2}}{u} \exp(-(\frac{10}{\sqrt{2}\xi} log_{10}\frac{R'u}{R})^2)du \end{aligned} \tag{3.42}$$

Proof:

In log-normal shadowing radio model, the signal attenuation between two nodes is not only dependent on the distance separating the two nodes but also a random shadowing value. As a result, the radio range of a node is not a perfect circle. However, we can still estimate a large circle around a node, which is large enough to cover the node's all immediate one-hop neighbors with a high probability. Let R' be the radius of such large circle of a node. We first define the value of R' and then we apply the similar approach used in proofing theorem 4 to estimate λ for log-normal shadowing radio model.

Recall that the probability that there exists a direct link between two nodes separating by distance s is expressed as

$$P_\wedge(s) = \frac{1}{2}\left[1 - erf(\frac{10}{\sqrt{2}\xi} log_{10}\frac{s}{R})\right], \quad \xi = \sigma/\alpha \tag{3.43}$$

The link probability $P_\wedge(s)$ is a decreasing function as the distance s increases, as illustrated in Fig. 3.4. Given a particular distance R', if $P_\wedge(R')$ is very small (e.g. $\alpha = 0.01$), it means that there is rarely a direct link between two nodes if their distance is greater than R'. In other words, R' can be approximately considered as the maximum radio range of each node. According to the definition, R' can be calculated as the distance that satisfies $P_{\wedge}(s) = \alpha$, where α is a very small value.

Now we can reuse the Fig. 3.10 and Fig. 3.11 to continue the proof if we change the symbol R to R' (i.e. from the average radio range to the maximum radio range). Clearly, when $i \gg R'$, as depicted in Fig. 3.10, the curve $\overset{\frown}{CF}$ can be approximated by a straight line. Consequently, the area of $A_{i,j}$ depends on the relative distance of i to j (i.e. x) and becomes irrelevant to i. We have

$$A_{i,j} = 2(A_{\widehat{CXE}} - A_{CXO}) = R'^2 \arccos \frac{x}{R'} - x\sqrt{R'^2 - x^2} \tag{3.44}$$

In order that the progress made along the next hop from node X towards the destination (i.e. change of state from i to j) is less than x, all neighbors of X must reside in the region to the left of CF. In other words, there is no nodes in region $A_{i,j}$, or there are some nodes in region $A_{i,j}$ but all these nodes do not have direct links to node X. Let T represent the progress made along this hop towards the destination. The probability that the progression of the next hop towards to destination is less than x is,

$$\begin{aligned} F_T(x) &= P(T < x) \\ &= P(\text{no direct link from } X \text{ to all nodes in region } A_{i,j}) \\ &= \sum_{k=0}^{\infty} \Big\{ Prob(\text{k nodes in } A_{i,j}) \cdot \\ &\qquad Prob(\text{no direct link from } X \text{ to any one of those } k \text{ nodes}) \Big\} \end{aligned} \tag{3.45}$$

Since nodes distribution follows a Poisson point process, the number of nodes within $A_{i,j}$ have a Poisson distribution with mean $\rho A_{i,j}$. Let $g(x)$ be the probability that there is a direct link from X to a node given the node is within region $A_{i,j}$ (note that $x = i - j$). Therefore,

$$Prob(\text{k nodes in } A_{i,j}) = \frac{(\rho A_{i,j})^k}{k!} \exp(-\rho A_{i,j}) \tag{3.46}$$

$$Prob(\text{no direct link from } X \text{ to any one of those } k \text{ nodes}) = (1 - g(x))^k \quad (3.47)$$

Thus $F_T(x)$ can be rewritten as ,

$$\begin{aligned}
F_T(x) &= \sum_{k=0}^{\infty} \left\{ \frac{(\rho A_{i,j})^k}{k!} \exp(-\rho A_{i,j}) \cdot (1 - g(x))^k \right\} \\
&= \exp(-\rho A_{i,j} g(x)) \sum_{k=0}^{\infty} \frac{[\rho A_{i,j}(1 - g(x))]^k}{k!} \exp[-\rho A_{i,j}(1 - g(x))] \\
&= \exp(-\rho A_{i,j} g(x)) \\
&= \exp(-\rho(R'^2 \arccos \tfrac{x}{R'} - x\sqrt{R'^2 - x^2}) \cdot g(x))
\end{aligned} \quad (3.48)$$

The probability density function (pdf) of the progression T is given by,

$$f_T(x) = \frac{d}{dx} F_T(x) \quad (3.49)$$

Therefore, the expectation of the progression T, i.e. λ, is,

$$\begin{aligned}
\lambda = \quad & E(T) = \int_0^{R'} x f_T(x) dx \\
= \quad & \int_0^{R'} x dF_T(x) \\
= \quad & \Big[x F_T(x) \Big]_0^{R'} - \int_0^{R'} F_T(x) dx \\
= \quad & R' - \int_0^{R'} \exp(-\rho(R'^2 \arccos \frac{x}{R'} - x\sqrt{R'^2 - x^2}) \cdot g(x)) dx \\
\overset{x=R't}{\Longrightarrow} \quad & R' \left(1 - \int_0^1 \exp(-\rho R^2 (\arccos(t) - t\sqrt{1 - t^2})) \cdot g(t) dt \right)
\end{aligned} \quad (3.50)$$

Now, we proceed to solve the $g(x)$, i.e, the probability that a node has a direct link to X given that the node is within the region $A_{i,j}$. Assume that node M is inside $A_{i,j}$, as shown in the Fig. 3.12. According to Poisson point process distribution, node M is uniformly distributed within $A_{i,j}$. Let S denote the random variable of distance between M and X. Given a particular value of s, the probability that variable S less than the value s is the probability that the node M falls within the shaded region depicted in Fig. 3.12. The figure shows that the shaded region has similar shape as $A_{i,j}$ but with a reduced size. Let $A_{i,i-s}$ represent the shaded region. The cumulative distribution function (cdf) of

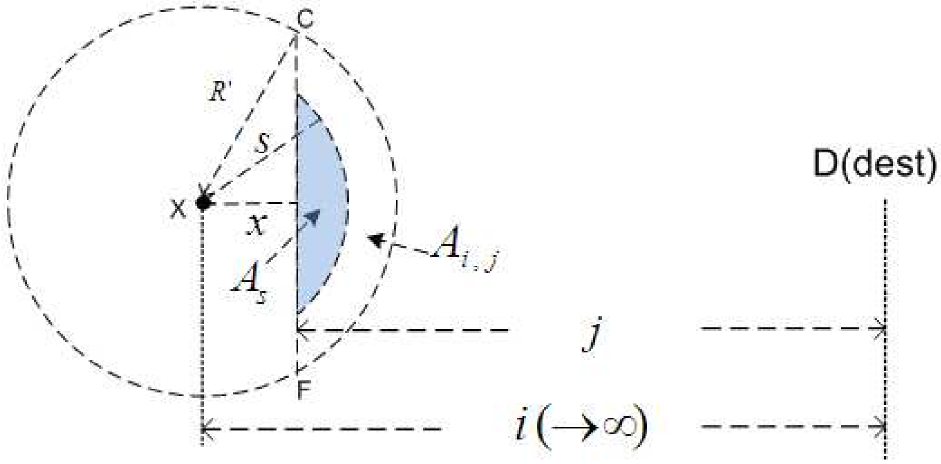


Figure 3.12: Illustration used to calculate the cdf of random variable S

random variable S can be expressed as,

$$\begin{aligned} F_S(s) &= Prob(S < s) = \tfrac{\text{area of } A_{i,i-s}}{\text{area of } A_{i,j}} \\ &= \frac{s^2 \arccos \frac{x}{s} - x\sqrt{s^2 - x^2}}{R'^2 \arccos \frac{x}{R'} - x\sqrt{R'^2 - x^2}} \end{aligned} \tag{3.51}$$

Consequently, the probability density function (pdf) of S is,

$$f_S(s) = \frac{d}{ds} F_S(s) \tag{3.52}$$

The probability that there exists a direct link between M and X is

$$\begin{aligned} g(x) &= \int_x^{R'} f_S(s) P_\wedge(s)(s) ds \\ &= \int_x^{R'} P_\wedge(s) dF_S(s) \\ &= \Big[P_\wedge(s) F_S(s) \Big]_0^{R'} - \int_x^{R'} F_S(s) dP_\wedge(s) \\ &= P_\wedge(R') + \int_x^{R'} F_S(s) \frac{10}{\sqrt{2\pi}\ln(10)\cdot \xi s} \exp(-(\frac{10}{\sqrt{2}\xi} log_{10} \frac{s}{R})^2) ds \\ &= \frac{10}{\sqrt{2\pi}\ln(10)\cdot\xi(R'^2 \arccos\frac{x}{R'} - x\sqrt{R'^2 - x^2})} \cdot \\ &\quad \int_x^{R'} \frac{s^2 \arccos\frac{x}{s} - x\sqrt{s^2 - x^2}}{s} \exp(-(\frac{10}{\sqrt{2}\xi} log_{10} \frac{s}{R})^2) ds \end{aligned} \tag{3.53}$$

Replace x with $R't$ and s with $R'u$, we have,

$$\begin{aligned} g(t) &= \frac{10}{\sqrt{2\pi}\ln(10)\cdot\xi(\arccos t - t\sqrt{1 - t^2})} \cdot \\ &\quad \int_t^1 \frac{u^2 \arccos\frac{t}{u} - t\sqrt{u^2 - t^2}}{u} \exp(-(\frac{10}{\sqrt{2}\xi} log_{10} \frac{R'u}{R})^2) du \end{aligned} \tag{3.54}$$

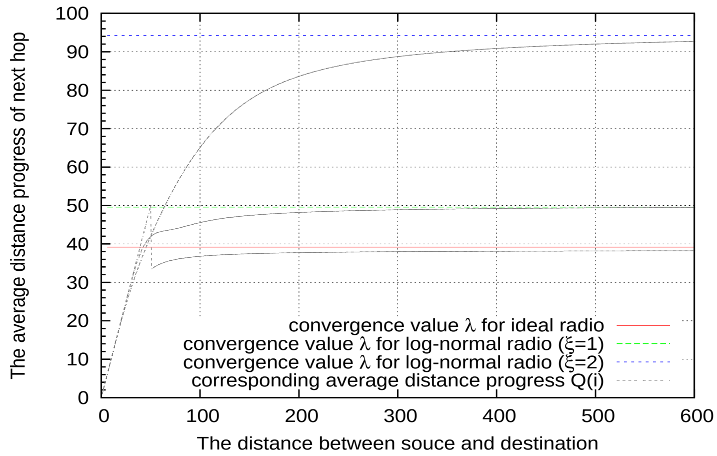


Figure 3.13: Convergence value of average distance progress

Finally combining Equation (3.50) and (3.54), the theorem is proved. ♠

We use Fig. 3.13 to illustrate the results of theorem 4 and 5, assuming $\rho = 0.0019, R = 50m$. The convergence value of λ for ideal radio model can be calculated straightforward using theorem 4. In the case of log-normal shadowing radio model (i.e. in theorem 5), we have a parameter α, which is a very small decimal and determines the approximation of nodes' maximum radio range, i.e. R'. In this example, we assume $\alpha = 1.0 \times 10^{-6}$. Correspondingly, based on the relation of $P_{\wedge}(R') = \alpha$, the R' is $150m$ and $300m$ for $\xi = 1$ and $\xi = 2$ respectively. Note that the maximum radio range R' is three times larger than the average radio range R (50m) when $\xi = 1$ and it expands to six times larger than R when ξ increases to 2. Knowing the value of R', the convergence value λ can be calculated according to theorem 5.

Fig. 3.13 compares the average progress of next hop to its analytical convergence value of λ under different radio model parameters. The figure clearly shows that the average progress converges to the analytical value and therefore justify Theorem. 4 and 5.

Knowing the value of λ, we can readily estimate the mean hop count of a communication pair. According to Equation. 3.34, we have the following corollary.

Corollary 1. *Given a communication pair separated by distance i, the lower bound of its mean hop count of is*

$$\widetilde{H(i)} = \frac{i}{\lambda} \tag{3.55}$$

where λ is the convergence value of average progress of one hop, and it is defined in Theorem 4 and 5 for ideal radio model and log-normal shadowing radio model respectively.

This estimation only needs time complexity of $O(1)$, which reduces the original complexity of analysis significantly. The comparisons of the estimation results to the original analysis results are illustrated in Fig. 3.14. The figure shows that the Corollary (1) can accurately approximate the mean hop count for ideal radio model and the log-normal radio model with small ξ. In the case of large value of ξ, the estimation can still serve as a lower bound of the mean hop count. Further, Fig. 3.14 shows that the estimations becomes more accurate with the increase of the distance for all radio models. For example, in the case of $\xi = 2$, the accuracy of the estimation is 76% when $i = 400m$, while the accuracy reaches to 83% when the distance i increases to 600m. Therefore, the proposed estimation in Corollary (1) can approximate the mean hop count for a large value of i, especially when $i \gg R$.

3.5 Simulation Results

In this section, we present a comprehensive set of simulations to validate our theoretical analysis. we developed a custom C++ simulator, which allows us to evaluate the results for the state transition probabilities, the hop count distribution and its mean values. In the first part of our simulations, where we validate our analysis, we use the network scenarios that conform to the assumptions made in the section 3.1. In the second part, we relax some assumptions, e.g. the as-

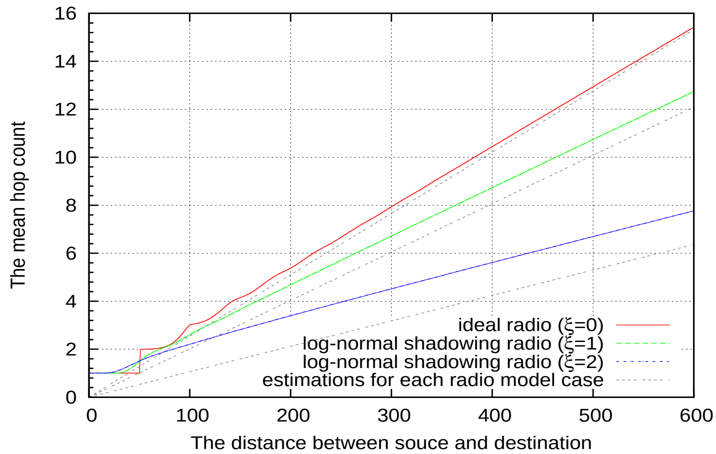


Figure 3.14: The lower bound estimation of mean hopcount

sumptions of homogenous Poisson point distribution and the network without boundary. We use the popular Random Way Point and the realistic movement traces of a vehicular network to generate two realistic network topologies. The objective of this exercise is to compare our analytical results with those from more realistic scenarios and more importantly to ascertain if the analysis can serve as bounds in these situations.

3.5.1 Scenarios Conform to the Assumptions

Recall that, our analysis assumes that the network has no boundaries. To realize this we simulate a large square network, and select a smaller square network at the center of this large network as the target network for our simulations. A similar approach is also used in [90]. We consider a square region of size $400m{\cdot}400m$, and assume that nodes are deployed with a node density of $\rho = 0.0019$ (resulting in a total of 304 nodes averagely). The average radio range of each node, R, is assumed to be $50m$. Thus the average number of one-hop neighbors m, derived by $m = \rho\pi R^2$, is equal to 15. We simulate three values of the signal randomness parameter ξ, i.e. 0, 1, and 2, where 0 represents the ideal radio model and other values represent the log-normal shadowing radio model. For each case of ξ, we run over 5000 simulations and the results presented are averaged over all runs.

For an individual run of the simulation, we randomly deploy nodes according to homogenous Poisson point process with a density of $\rho = 0.0019$. Once the nodes are placed, we use the appropriate radio model with the particular value of ξ to generate link connectivity over all pairs of nodes. Then for each pairs of nodes in the target network, we employ greedy routing to find the routing path from the one node (the source) to another (the destination). Once the routing paths are established, we can identify the next hop node for each individual node. This enables us to determine the next hop state j for each current state i. Grouping the transitions from all nodes located at distance i from the destinations gives us the distribution of the transition probabilities from state i. For each pairs, we also record its source-to-destination distance and the hop count. Then we cluster the pairs that have same source-to-destination distance together and compute the mean hop count and its distribution for each distance case. We also compute the confidence interval for these simulations results. Give sample size of n, sample average of μ and sample standard deviation of σ, we compute the confidence interval by $[\mu - 1.96 * \sigma/\sqrt{n}, \mu + 1.96 * \sigma/\sqrt{n}]$. The simulation details are explained in Appendix I(A)[1] .

Fig. 3.15 compares the simulation results of state transition probability with the corresponding analysis results derived in theorem 1 and 2. It shows that the simulations results are perfect in line with the analysis results, and therefore justifies that our model can accurately calculate the state transition probability.

Note that, we only plot the average values in Fig. 3.15. Since we get a large number of samples from the simulations, the sample average values presented here

[1] Our simulations are based on stead-state simulations, which do not have transient state, start-up or warm-up period. Some simulations in this thesis do last for a period of time. However, this aims to get a large number of samples so we can have statistically meaningful results. In all simulations, we sample results from the very beginning of each simulation.

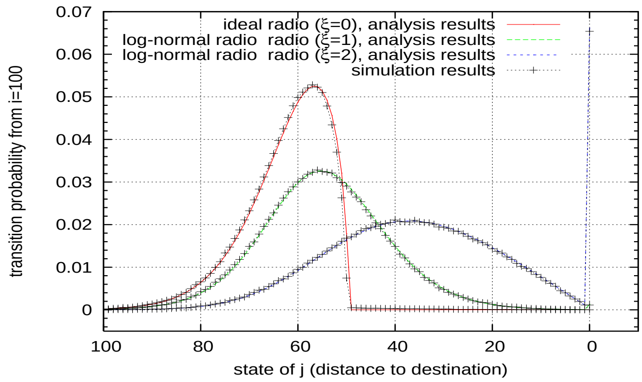


Figure 3.15: Validation of State transition probability($R = 50$, $i = 100$)

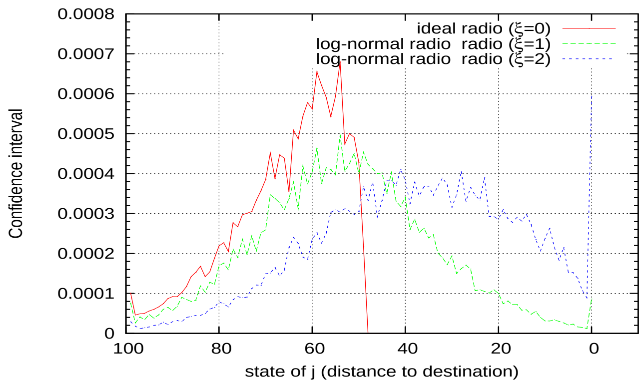


Figure 3.16: The confidence interval of simulated state transition probability($R = 50$, $i = 100$)

are close to their true values. As a results, the corresponding confidence intervals are very close to these average values. Fig. 3.16 illustrates the value of confidence interval, which is expressed as $1.96 * \sigma/\sqrt{n}$ (i.e. half size of the interval). The figure shows that confidence intervals are so narrow, which only fluctuate less than ± 0.0007 from the average values. Note that, the values of y axis in Fig. 3.16 are very small compared to those values in Fig. 3.15. If we plot the confidence intervals along with the mean values, the confidence intervals will overlap with the corresponding average values. Therefore, in the rest of the thesis, we either plot the confidence interval in a separate figure or we use text to clarify the values of the confidence intervals.

Fig. 3.17 shows the hop count distribution comparison between the simu-

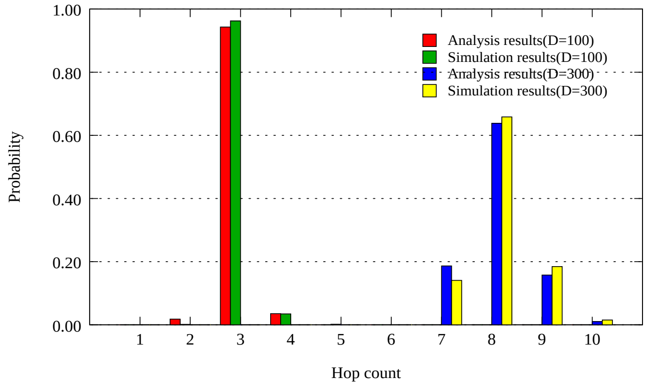


Figure 3.17: Hop count distribution comparison in ideal radio model

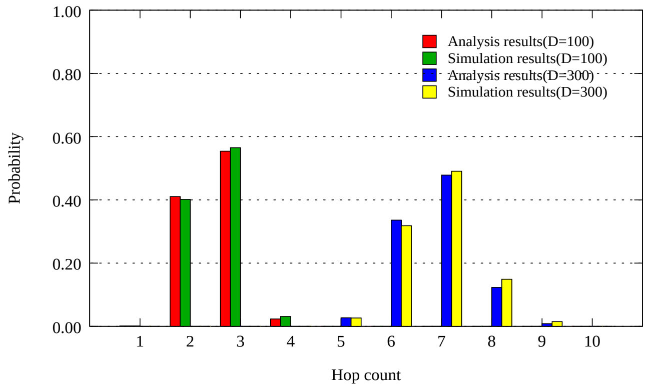


Figure 3.18: Hop count distribution comparison in log-normal radio model ($\xi = 1$)

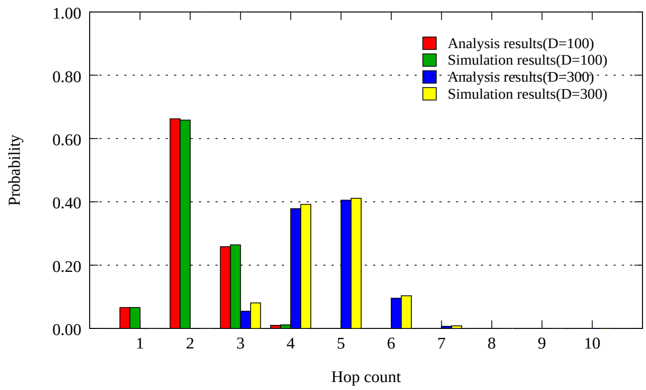


Figure 3.19: Hop count distribution comparison in log-normal radio model ($\xi = 2$)

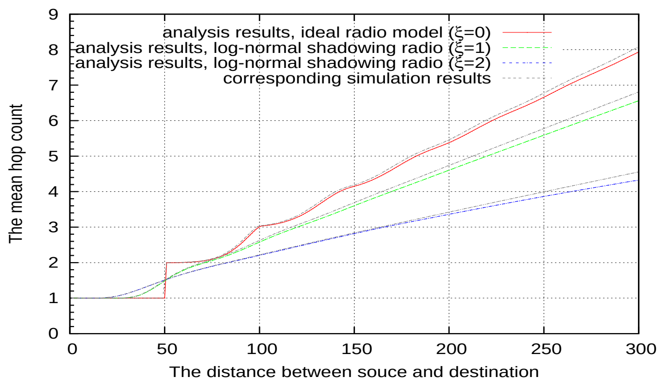


Figure 3.20: Mean hop count comparison of simulation results to analysis results

lation results (average values) and the analysis results under ideal radio model scenario. It illustrates the hop count distribution of two communication pairs. One is with the source-to-destination distance of 100m and another is 300m. Similar, Fig. 3.18 and Fig. 3.19 illustrate the hop count distribution for log-normal radio model with randomness parameter ξ of 1 and 2 respectively. The confidence intervals of these results are very narrow and are just within ± 0.003 from the average values in most of cases. The close matching between the simulation and analysis results in these three figures validates the the analytical model of hop count distribution derived in theorem 3.

Comparing the hop count distribution results, i.e. Fig. 3.17, Fig. 3.18 and Fig. 3.19, we can see that hop count is more evenly distributed when the signal randomness is increasing. For example, for ideal radio model (i.e. with zero signal randomness), the hop count of 8 clearly dominates the distribution of $D = 300m$ with the probability of 0.6, as shown in Fig. 3.17. However, when the signal randomness increases to 2, the dominating probability of $D = 300m$ drops to 0.4, and it happens at both hop count 4 and 5, shown in Fig. 3.19.

Fig. 3.20 depicts the mean hop count with varying source-to-destination distance for the different radio model. The corresponding confidence intervals are plotted in Fig. 3.21. Fig. 3.21 shows that the confidence intervals only fall

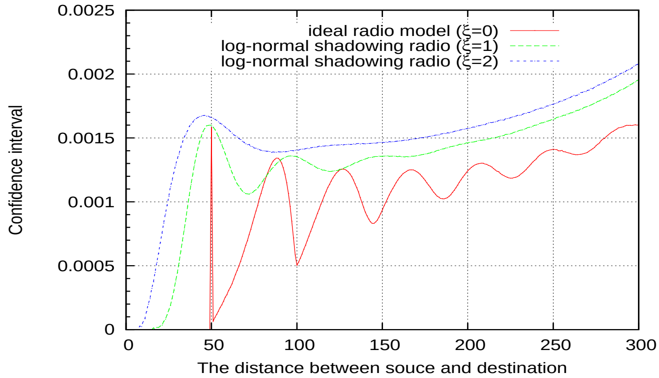


Figure 3.21: Confidence interval of simulated mean hop count

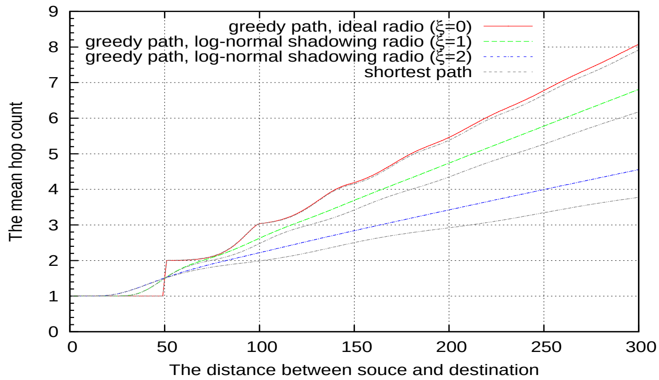


Figure 3.22: Mean hop count comparisons between shortest path and greedy path

within ± 0.002 from the average values, which verifies the accuracy of the simulated average values[2] . All these figures show that the simulation results are in line with our theoretical results. Thus the results validate our analysis exercise.

To understand how the routing path in greedy routing resembles the shortest routing path in term of hop count, we also simulate the hop count incurred in the shortest path routing. The comparison is illustrated in Fig. 3.22. It shows that the greedy routing matches the shortest path very well in term of mean hop count when ideal radio model is used. However, in log-normal radio, the greedy routing exhibits a noticeable difference compared to the shortest path. The difference is becoming more prominent with the increasing of signal randomness ξ. These

[2] Note that, the values of y axis in Fig. 3.21 are very small compared to those values in Fig. 3.20.

results reveal that the statement of "greedy routing can approximate find the shortest path in a relative dense network" claimed in some previous work [7, 29] only applies for ideal radio model.

3.5.2 Realistic Scenarios Not Conform to Assumptions

In the previous sub-section, all simulations parameters conformed to the assumptions used in our analysis. However, not all these assumptions will hold true for realistic wireless ad-hoc networks. In particular, a real-world network would not usually consist of homogenous Poisson point distributed nodes or uniformly distributed nodes. In this sub-section, we wish to investigate if our theoretical results are relevant in practical scenarios. For this we first use the popular random way point mobility model [96] to model a none-uniformly distributed nodes. In the second instance, we investigate a real-world vehicular ad hoc network, a popular application domain for MANETs. Our aim is to determine if the mean hop count as derived in our analysis are pertinent for these real-world networks.

For the first case we choose a scenario that uses the random way point mobility to model the nodes distribution. In random way point model, each node randomly selects a moving destination and a moving speed from the predefined speed range. The node then moves to the destination at the selected speed. After the node reaches to the destination, it pauses for a certain duration, also randomly determined, and then selects the next destination and repeat the process.

The simulated network is a square region with size of $400m \cdot 400m$. The total number of nodes is 304, which leads to the average node density of $\rho = 0.0019$. Each node has the radio range of 50 meters. The speed of each node varies uniformly from 0 to 20 m/s (i.e. from 0km/hour to 72km/hour) and the pause time is assumed to be zero.

A simulation run lasts for 5000 seconds during which we take a snapshot of

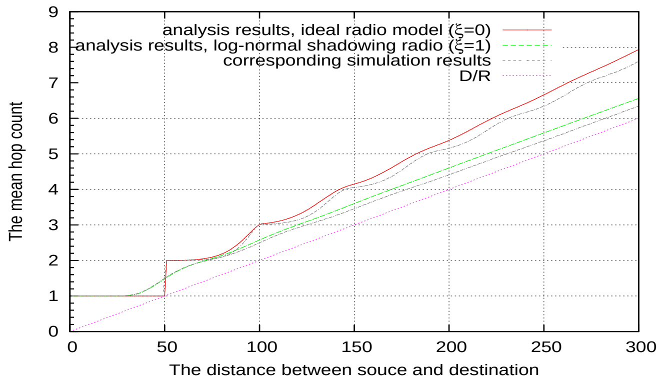


Figure 3.23: Mean hop count comparisons for random way point model

the network every five seconds. For each snapshot, we use the same approaches employed in previous section to simulate the hop count results. First, we use the appropriate radio model with the particular value of ξ to generate link connectivity over all pairs of nodes. Then for each pairs, we employ greedy routing to find the routing path from the one node (the source) to another (the destination). For each pairs, we record its source-to-destination distance and the hop count. Finally we cluster the pairs that have same source-to-destination distance together over all snapshots and compute the mean hop count for each distance case.

Fig. 3.23 illustrates that our analytical results slightly over-estimate the mean hop count as compared to the simulations. The reason for this is that in the random way point model, nodes tend to move towards the central area of the network, with the consequence that the central area is much denser as compared to the regions near the border [96]. Hence, on average, the progress made per hop towards the destination is larger as compared to a purely uniform distribution (as in our analysis), resulting in a shorter hop count. However, Fig. 3.23 clearly demonstrates that our results are far more accurate than the frequently used measure of the mean hop count, D/R.

The mobility model used in the second instance of our simulations is based on the actual movement of buses in the King County Metro bus system in Seattle,

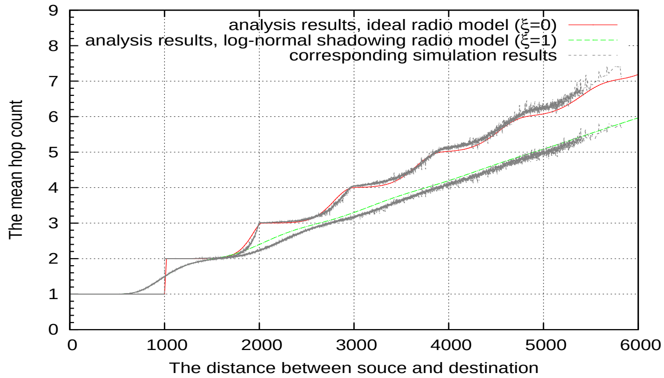


Figure 3.24: Mean hop count comparisons for realistic vehicular networks

Washington [97]. We extract an area of size $4000m * 7000m$ corresponding to the the downtown area of Seattle. The duration of this trace spans 30 minutes. We assume that the radio range of each node is 1000m, which is consistent with that for DSRC [98] and the results from [99]. During this simulation run, we found that the average number of nodes in the selected region was 288, and thus the nodes density is $1.03 * 10^(- 5)$. We apply these identical parameters to the analytical model and generate the mean hop count results. Fig. 3.24 compares the analysis results with those from the simulations for this realistic vehicular network. It is evident that the analysis results are well matched with those from the simulations.

3.6 Summary

Motivated by the fundamental role of hop count in the performance analysis in multi-hop wireless ad-hoc networks, we proposes an accurate analytical model to estimate the hop count metric for greedy geographic routing. We formulate the hop count distribution and the mean value given a communication pair under the consideration of both ideal radio model and the realistic log-normal shadowing model. The analysis results shows that the radio model has a great impact on the hop count metric. We further propose the approximations to simplify the mean hop count analysis, which reduce the time complexity significantly. We conduct a rich set of simulations that validate the analytical model. The analysis results are further confirmed through a trace driven simulation of a practical vehicular ad-hoc network that exhibits realistic topologies of public transport buses in a metropolitan city.

The derived hop count knowledge can be incorporated in performance analysis in multi-hop wireless ad-hoc network, or assisting routing protocol design. For example, several work [100] [13] [101] have concluded that the throughput of a given communication pair is inversely proportional to the hop count from the source to the destination. Therefore, the hop count knowledge can be used as a fundamental element to estimate the throughput of wireless ad-hoc networks. Further, the hop count of a traffic flow represents the number of retransmissions of a packet experiences from the source to the destination, which can be used to determine the traffic load volume (including relay transmissions) imposed on routing layer [102]. Last but not least, we can apply the probability distribution of hop count to assist protocol design, e.g. the distance-based local geocasting protocol as discussed in [103].

Chapter 4

Analysis of Per-node Traffic Load in Multi-hop Wireless Sensor Networks

Wireless Sensor Networks (WSN) are being increasingly used in a variety of applications ranging from health care, environmental monitoring to industrial automation [4]. Ensuring energy efficient operation is critical, especially given that a typical WSN is deployed in remote and unaccessible areas and sensor nodes are equipped with a limited battery source. A typical WSN consists of a large number of nodes deployed over a large area. Hence, packets generated at nodes that are outside the communication range of the sink have to be relayed by other nodes.

It has been well accepted that the energy expended in transmission and reception of packets forms a significant component of the total energy budget of a sensor node [4, 35]. Consequently, an analytical model that can accurately estimate the traffic load incurred at each sensor node is instrumental in predicting the energy consumption of the nodes and thus the operational lifetime of the entire WSN. In addition, the traffic load characterization can provide important insights for designing and configuring energy efficient network protocols. As an example, the information about the traffic load of nodes in a collision domain can be used to tune MAC parameters such as slot assignments in TDMA-based schemes and also for setting the wake-up and sleep durations in duty-cycling protocols where nodes periodically go to sleep to conserve energy. In addition, knowledge of the energy expenditure of nodes can be useful in planning deployment and maintenance of the WSN. The network designer can deploy redundant nodes in regions where nodes are expected to expend their energy at a higher rate. Maintenance cycles can also be planned for replacing or charging depleted nodes, thus preventing the formation of coverage holes in the network.

The traffic load of a given sensor node depends on several factors. The first and foremost is the relative distance of the node to the sink. In general, the closer the sensor is to the sink, the greater is the traffic load. This is because the nodes closer to the sink have to relay data packets transmitted by other far-off nodes. The traffic load also depends on the routing protocol employed in the network as it determines the selection of the next hop node for relaying the data towards the sink. Lastly, the characteristics of the environment, which affects the radio communication behavior of the sensor nodes also has an impact on the traffic load.

Analytical models which can accurately characterize the traffic load of nodes in a WSN are rare. The available models, e.g., those presented in [36, 37, 38], are very simplistic and do not incorporate the specifics of the routing protocol used by the sensor nodes. As such, these models can only be used to provide a rough estimate of the mean traffic load over a relative large geographic area. In [39], Esa Hyytia et. al. have analyzed the traffic load of nodes in a very dense wireless multi-hop network with randomly selected communication pairs. However, their measure of traffic load per unit radio coverage can only provide for a coarse-grained characterization of the traffic load. In addition, they have assumed that the shortest path from any node to the sink can be approximated by a straight line segment, which is true only for highly dense deployments. Furthermore, this work and in fact all other aforementioned efforts have adopted the idealistic circular coverage radio model, which is known to be a poor abstraction of the real communication environment.

In this chapter, we take a first step towards developing a detailed and precise analytical model for estimating the per-node traffic load in a WSN. Given that the target domain of WSN is quite broad in terms of the type of applications, networking protocols (routing, MAC, etc) employed and characterization of the communication environment, developing a generalised model is extremely challenging. Thus, we have chosen to focus on a important sub-set within this vast domain. In terms of the application load, we focus on periodic monitoring applications, wherein the sensor nodes sample the environment periodically and forward the collected data (e.g. temperature, humidity) to the sink. A significant portion of WSN deployed today fall into this category (e.g.: Redwood Forest [104], Great Duck Island [105]). With regards to the routing strategy, which is important for selecting the next-hop node, we have considered the popular **greedy routing** forwarding scheme [29, 106, 107, 108, 31]. For developing an analytical model of the traffic load it is necessary to use an appropriate model that abstracts the wireless communication characteristics of a realistic environment. In our analysis we have used both the idealistic radio model and the realistic log-normal shadowing model, thus enabling us to compare the impact of the two on the results. To the best of our knowledge, this work is the first attempt at developing a comprehensive model for characterizing the per-node traffic load in a WSN.

In a multi-hop wireless network, the traffic load at a node is a collective results of the packets originated from this node and the packets relayed by this node. The former one is given by the application requirement, while the latter one depends on the packet forwarding behavior of the routing scheme. In our previous hop count analysis (Chapter 3), we have identified this forwarding behavior of greedy geographic routing. Specifically, we apply a Markov chain model to analyze the probability that a node forwards its packets to another node. Having this knowledge, a node can estimate the chance that each of its neighbor uses this node to relay their packets, therefore the node is able to estimate the total number of packets it will relay.

The analytical model is validated by a rich set of simulations. Our results quantitatively confirm that the traffic load of a node increases with the proximity to the sink. More importantly, we observe a peculiar difference in the traffic load in the immediate vicinity of the sink for the two radio models. For the idea radio model, the traffic load declines significantly after a certain knee point as one moves closer to the sink, resulting in the formation of a volcano shape. On the contrary, with the more realistic log-normal shadowing radio model, we observe the opposite effect, i.e., the traffic load of nodes very close to the sink increases quite significantly as a function of the proximity to the sink, which results in a mountain peak pattern. For simplifying our analysis we have had to make several assumptions. We have also investigated the impact of relaxing some of these assumptions and observed that our analytical results are still valid in these circumstances.

The rest of the chapter is organized as follows. In Section 4.1, we provide an overview of our model. In Section 4.2, we present the analysis of the traffic load for both the ideal and log-normal radio models. Section 4.3 validates the analytical model by comparing the results with those from simulations. Finally, Section 4.4 summaries this chapter.

4.1 Overview of the System Model

For mathematical tractability, we make the following simplifying assumptions:

- The sensor nodes are randomly deployed in an infinite plane area. The node distribution follows a homogenous Poisson point process with a density of ρ sensors per unit area, which can approximate uniform distribution for large area. This assumption has been widely used in analyzing multi-hop wireless ad-hoc networks [90, 91, 92].

- The sensor nodes are deployed in a circular region with the sink (base station) situated at the centre. While this assumption simplifies the closed-form expressions in our analysis, the results can be readily applied to other deployment scenarios (e.g., a sink at the edge of the network) with minor modifications. It should be noted that similar assumptions are made in prior work [36, 37, 38].

- All sensors have identical transceivers and the wireless links are assumed to be symmetric.

- Each sensor node periodically generates a data packet containing the relevant sensed data and routes the packet to the sink. This *clock driven data generation model* is typical for many monitoring applications (e.g., monitoring of temperature, soil moisture, etc.) [109]. Our model can be readily extended for an *event-driven data generation model*, wherein the sensor nodes generate packets in response to certain events of interest.

- All sensors generate packets with the same periodicity. This is indeed the case with several realistic sensor network deployments. For example, the WSN deployments in Redwood Forest [104] and Great Duck Island [105] conform with this assumption.

- The network is dense enough such that greedy routing always succeeds in finding a next hop node that advances the data packet towards the sink. In other words, we assume that the forwarding strategy does not encounter a *local minima condition* and thus, neglect the effect of planar routing, which is employed in these circumstances.

- Sensor nodes forward packets towards the sink without any data aggregation.

- Packet loss rate is q and it is uniform for all data transmissions. Therefore, a sender needs to transmit averagely $\frac{1}{1-q}$ times to successfully deliver one packet to the receiver.

Excluding packets that are generated by sensor nodes which can directly communicate with the sink, packets generated by all other node are relayed by intermediate nodes as determined by the routing strategy employed. The behavior of a packets' progress towards the destination is therefore important in determining the traffic load at each sensor. We use a discrete Markov chain to model the hop-by-hop progress of a packet from the source to the destination. The state of the Markov chain is defined as the Euclidean distance (measured in some consistent metric unit, e.g. meters) between the current forwarding node that holds the packet and the destination. Ideally, this distance should be modeled as a continuous random variable. However, to simplify our model, we use a discrete state space to approximately represent the continuous distance values. We quantize the distances resulting in a state space of $(0, \varepsilon, 2\varepsilon, ..., n\varepsilon, ...)$, where the parameter ε is the interval of the state space (i.e. the quantization coefficient). When the interval ε is small enough, the discrete state space approximates the original continuous distance metric.

In previous chapter, we have illustrated the state transition of this Markov china model, as see in Fig 3.1. We also formulate its main property, i.e. the state transition probability, under both the idealistic radio model and the realistic log-normal shadowing model. We directly apply those results of the state transition probability into this work to analyze the per-node traffic load.

Intuitively, the sensors closer to the sink will have more packets to transmit, because they have to relay the packets originated from other sensors that are distant. Therefore the traffic load of a sensor is dependent on its distance to the sink. We refer the packet generating interval as "time unit". The goal of our analysis is to determine the traffic load at a sensor, which is defined as the total packets transmitted by the sensor in one time unit. Let $f(d)$ represent the average traffic load incurred by sensor nodes located at a Euclidean distance of d units from the sink. Note that, $f(d)$ includes the packet generated by the sensor and those forwarded on behalf of other sensors. Consider a sensor node located at distance j from the sink. Since greedy forwarding is employed, this node can only receive packets from its one-hop neighbors that are further from the base station than j. In other words, any other node at distance i, where $i > j$, could possibly forward its packets to this particular node. If we denote the state transition probability from state i to j as $P_{i,j}$, then the traffic load of a sensor at distance j is dependent on the traffic load of sensors at distance i and the state transition probability $P_{i,j}$.

Therefore, using the state transition probabilities derived in previous chapter, we can recursively calculate the traffic load of the individual sensors. The main symbols used in the rest of this chapter are listed in Table 4.1.

4.2 Analysis of the Per-node Traffic Load

The analysis is independent of the radio model under consideration. One simply has to substitute the appropriate state transition probability equations as derived in the previous chapter for the radio model under consideration. Recall that, the traffic load of a sensor refers to the average number of data packets transmitted by the sensor during one time unit (see definition in Section 4.1). By recursive calculation, we have,

Theorem 6. *Consider a circular shaped network of radius l with the sink located at the centre. The traffic load of a sensor located at a distance d from the sink,*

Table 4.1: List of main symbols used in the analysis of per-node traffic load

Symbol	Definition
l	Radius of circular network
R	Average radio range of sensors
ρ	Node density, i.e. the number of nodes per unit area
ε	Quantization interval
i, j, d	Euclidean distance from a sensor to the centrally located sink
ξ	Signal randomness parameter in log-normal shadowing radio model
q	packet loss rate
$P_{\wedge}(s)$	The probability that two nodes separated by distance s can communicate with each other
$P_{i,j}$	State transition probability. i.e. the probability that a sensor at distance i from the sink can forward its packets to the sensor at distance j
$f(d)$	Traffic load, i.e. the average number of data packets transmitted by a sensor at distance d during one time unit

$f(d)$, is given by,

$$f(d) = \frac{S_t(d)}{\pi(2d+\varepsilon)\varepsilon\rho} \cdot \frac{1}{1-q} \tag{4.1}$$

where $S_t(d)$ is given by,

$$S_t(d) = \begin{cases} \pi(2d+\varepsilon)\varepsilon\rho & \text{if } d = l, \\ \pi(2d+\varepsilon)\varepsilon\rho + \sum_{i\in(d,l]} P_{i,d}S_t(i) & \\ & \text{if } 0 < d < l, \end{cases} \tag{4.2}$$

and $P_{i,d}$ denotes the state transition probability for the radio model under consideration. $P_{i,d}$ for the ideal and log normal radio model have been derived in theorem 1 and theorem 2 respectively.

Proof. Let $n(d)$ be the average number of sensors located at distance d, and $S_t(d)$ be the total number of different packets collectively transmitted by these $n(d)$ nodes. Note that, to successfully deliver one data packet, we actually need $\frac{1}{1-q}$

transmissions due to packet lost rate of q. We have:

$$f(d) = \frac{S_t(d)}{n(d)} \cdot \frac{1}{1-q} \tag{4.3}$$

Recall that, we approximate distance as a discrete state space with an interval of ε. As a result, the number of sensors located at distance d are actually the number of sensors within the thin ring of thickness ε located between the two concentric circles of radius d and $d+\varepsilon$. Since the area of the thin ring is $\pi(2d+\varepsilon)\varepsilon$, the average number of sensors at distance d is given by,

$$n(d) = \pi(2d+\varepsilon)\varepsilon\rho \tag{4.4}$$

The sensors at distance d transmit all the correctly received packets plus the packets originated by these $n(d)$ sensors. Let $S_r(d)$ be the average number of different packets received by all sensors at distance d in one time unit. We have,

$$S_t(d) = n(d) + S_r(d) \tag{4.5}$$

We now proceed to calculate $S_r(d)$. When $d = l$, the sensors at distance l are the furthest sensors from the sink and therefore they are not involved in forwarding packets for other sensors. Therefore $S_r(d) = 0$ and

$$S_t(d) = n(d) \quad \text{if } d = l \tag{4.6}$$

For $d < l$, as discussed in Section 4.1, any sensor at distance farther than d can forward packets to distance d. In other words, the sensors at distance i, where $i \in (d, l]$ can forward packets to sensors at distance d. For a particular distance i, the average number of different packets transmitted from the nodes at i to those at d is $S_t(i)$ multiplied with the corresponding state transition probability $P_{i,d}$. By summing up all cases of i, we have,

$$S_r(d) = \sum_{i \in (d,l]} P_{i,d} S_t(i) \quad \text{if } 0 < d < l \tag{4.7}$$

Combining Equations (4.5) and (4.7), we have,

$$S_t(d) = n(d) + \sum_{i \in (d,l]} P_{i,d} S_t(i) \quad \text{if } 0 < d < l \tag{4.8}$$

Finally, combining Equations (4.3, 4.4), (4.6) and (4.8), the theorem is proved. □

Note that, $S_t(d)$ is a recursive function. Since, we know $S_t(l)$, the initial value of $S_t(d)$, we can calculate $S_t(l - \varepsilon)$ according to Equation (4.2). Similarly, $S_t(l-2\varepsilon)$ and so on can be derived. Finally, for any given d, $S_t(d)$ can be computed, following which, we can calculate $f(d)$ using Equation (4.1). Note that, theorem 3 derives the traffic load of all sensors excluding the sink. Recall that, the sink exclusively acts as a receiver. Hence, the traffic load of the sink is zero.

4.3 Simulation Results

In this section, we present results from an extensive set of simulations. The first goal of our evaluations is to validate the results of our analysis. For this, we developed a custom C++ simulator, which conforms to the assumptions listed in Section 4.1. We investigate if the state transition probabilities and traffic load derived in Section 4.2 conforms with the simulation results. Our second objective, is to investigate if our analytical results are valid in more realistic scenarios where the simplifying assumptions made in our analysis are relaxed. To study this, we used the NS2 simulator and incorporated a popular geographic routing protocol, GPSR and a CSMA/CA MAC protocol to make the simulations more realistic.

Recall that, our analysis assumes that nodes are deployed in an infinite plane and that we focus on a circular sub-region of area πl^2 (l being the radius). To realize this we simulate a large square network to approximate the infinite network, and select a smaller circular network at the center of this infinite network as the target network for our simulations. A similar approach is also used in [90]. We consider a circular region of radius $l = 200m$, and assume that sensor nodes are deployed with a node density of $\rho = 0.0019$ (resulting in a total of 240 nodes). The average radio range of each node, R, is assumed to be $50m$. We simulate three values of the signal randomness parameter ξ, namely, 0,1, and 2, where 0 represents the ideal radio model and other values represent the log-normal shadowing radio model. For each case of ξ, we run 5000 simulations and the results presented are averaged over all runs.

For an individual run of the simulation, we randomly deploy nodes according to homogenous Poisson point process with a density of $\rho = 0.0019$. Once the nodes are placed, we use the appropriate radio model with the particular value of ξ to generate link connectivity over all pairs of nodes. Then for each node in the selected circular network, we employ greedy routing to find the routing path to the centrally located sink.

Recall that, each sensor node generates one data packet during one time unit. In other words, each routing path in the network carries one data packet in one time unit. Therefore, for each node, the number of routing paths that traverse through the node is the traffic load at this node. Finally, we group all the nodes that are located at the same distance i from the sink and calculate the average per-node traffic load for that particular state, i. The simulation details are explained in Appendix I(B).

Fig. 4.1 plots the average per node traffic load as a function of distance to the sink for a network of radius $l = 200m$ and $R = 50m$ (assuming packet loss rate $q = 0$). Fig. 4.2 shows that the confidence intervals fall within ± 1 from the average values in most of cases, which verifies the accuracy of the simulated average values. The close match between the simulation and analytical results confirms the validity of the analytical model. As intuitively expected, the closer the node is to the sink, the greater is its traffic load. However, the interesting

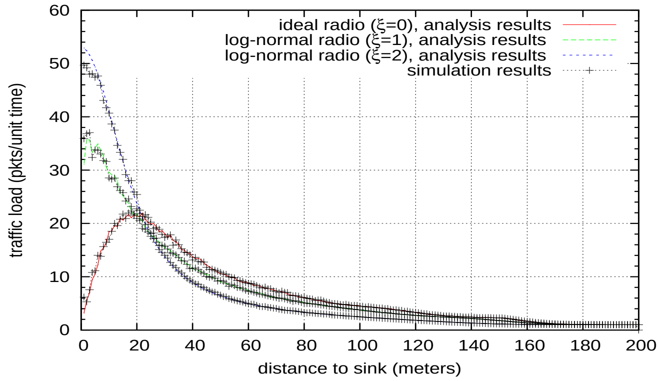


Figure 4.1: Traffic load as a function of distance to sink ($l = 200$, $R = 50$, $q = 0$)

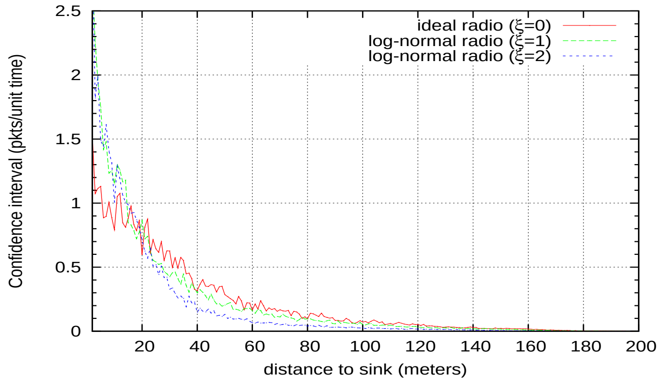
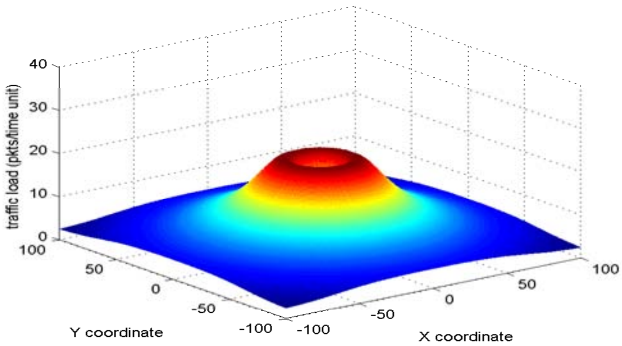
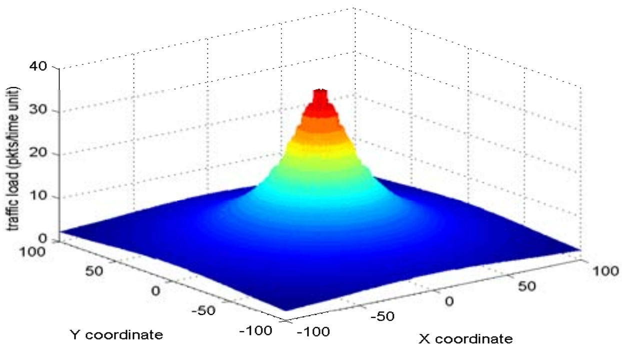


Figure 4.2: Confidence interval of per-node traffic load ($l = 200$, $R = 50$, $q = 0$)

result revealed by the graph is that the traffic load pattern of nodes close to the sink varies significantly with the radio model. A 3-D plot of the traffic load, as depicted in Fig. 4.3, illustrates this effect more clearly. In these graphs, the z-axis represents the traffic load and the x and y axes represent the two dimensional cartesian coordinate system with the sink located at the origin (0,0). For the ideal radio model, the traffic load declines significantly if nodes are too close to the sink, giving rise to a **volcano** shape as illustrated in Fig. 4.3(a). We observe that the traffic volcano is clearly contained within a circular area (volcano zone) of radius R, the radio range of the sensor nodes, around the sink. The crater signifies a safe area within the volcano zone where the sensor nodes do not experience the full force of the traffic volcano.



(a) ideal radio



(b) log-normal radio ($\xi = 1$)

Figure 4.3: Traffic load as a function of position relative to the sink ($l = 200$, $R = 50$, $q = 0$)

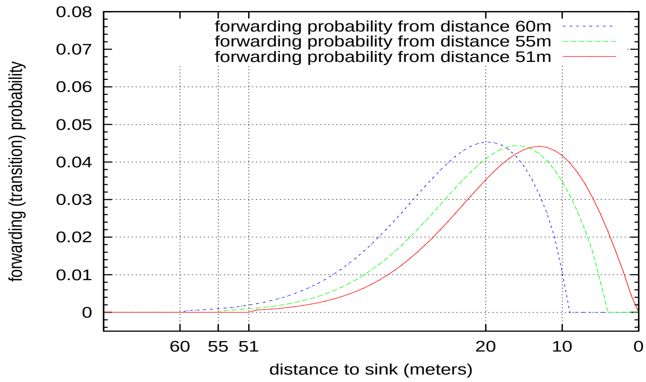


Figure 4.4: Forwarding probability (state transition probability) of nodes as a function of the distance to the sink

The volcano shape of the traffic load in Fig. 4.3(a) is a direct consequence of the state transition probability distribution (see Section 3.3). Since, the crater in Fig. 4.3(a) is observed in the radio range of the sink, we focus on nodes in that region. The traffic load of a node is largely dependent on the forwarding probabilities (i.e. state transition probabilities) of its one-hop neighbors, which could potentially select this node as the next-hop relay. Fig. 4.4 above illustrates the forwarding probabilities of three nodes, which are located at a distance of 60m, 55m and 51m, respectively from the sink. As seen from this graph, the forwarding probability initially increases up to a knee point as the distance to the sink reduces. However, after the knee point, the probability reduces in the immediate vicinity of the sink. The general shape of the forwarding probability curves implies that the nodes, which are very close to the sink are rarely chosen as forwarders (Note that the nodes, located at a distance less than 50m, do not need any forwarders). Hence, the traffic load decreases in the immediate neighborhood of the sink leading to the formation of the crater in Fig. 4.3(a).

As evident from Fig. 4.3(b), the situation is quite different with the log-normal shadowing radio. In contrast with the ideal radio model, the traffic load of the nodes increases continuously as we move closer to the sink creating a *mountain peak* effect. This pattern is caused by the radio signal randomness that is prevalent

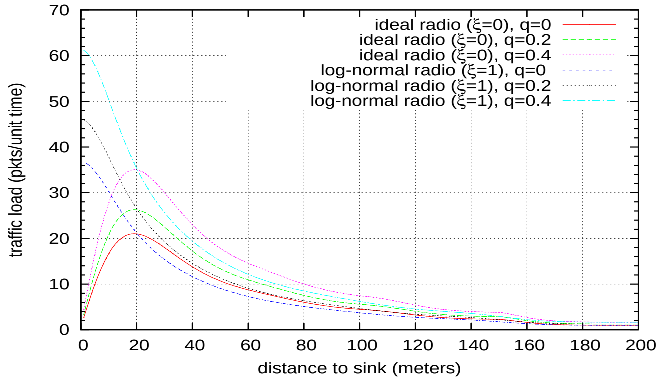


Figure 4.5: The effect of packet loss on traffic load ($l = 200$, $R = 50$)

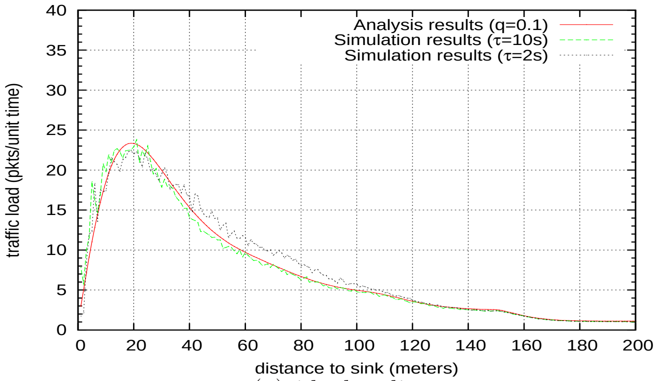
in the log-normal shadowing radio. Recall that, due to this two nodes that are separated by a distance greater than the average radio range R may still be able to communicate with each other. As a result, the nodes closer to the sink tend to receive a large number of packets from other distant nodes. At the same time the signal randomness does not guarantee that two nodes, which may be very close to each other will always be able to communicate. Consequently, even the nodes that are close to the sink may still require other nodes to relay their packets. This effect further compounds the traffic load in the vicinity of the sink, thus resulting in the mountain peak effect.

The effect of packet loss on the traffic load is depicted in Fig. 4.5, where the packet loss rate varies from 0 to 0.4. Fig. 4.5 shows that traffic load with the presence of packet loss has a similar characteristics as those of no-loss situation. However, the presence of packet loss increases the height of the volcano and the mountain peak, therefore further deteriorates the un-even distribution of the traffic load over network area.

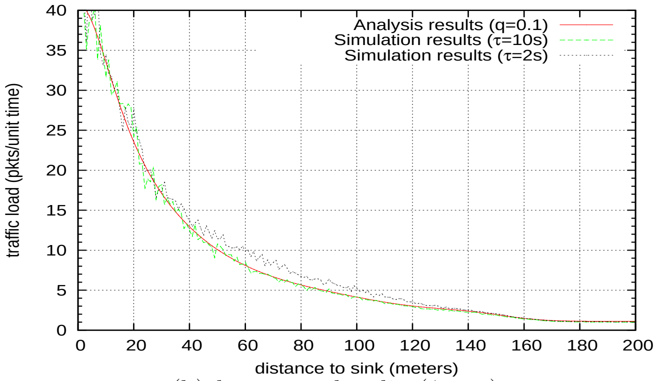
In addition to the above topology-driven simulation, we have also carried out simulations using NS-2. Unlike the previous simulations, the network now actively carries traffic and we have also relaxed several of the assumptions used in our analysis. This enables us to evaluate if the analytical results are applicable in more realistic scenarios. We now use a bounded circular network of radius, $l = 200m$ with a centrally located sink and incorporate GPSR, a popular greedy forwarding routing strategy. The node density is again 0.0019 and the average radio range is 50m. Further, we do away with the assumption of uniform packet loss rate and simulate a CSMA/CA MAC protocol. We wanted to investigate if these factors significantly affect the observed traffic load. For this we considered two different traffic loads - (i) low traffic load with sensors transmitting a new packet to the sink every 10 seconds and (ii) high traffic load with the packet generation periodicity reduced to 2 seconds. The results presented are averaged over 2000 runs, each of which lasts 60 seconds. We compare the simulation results to the analytical results where the packet loss rate is set to 0.1. As seen from Fig. 4.6(a), for low load, the observed traffic load is very close to the analytical results. At higher load, there is a slight deviation from the analytical results. This difference can be attributed to the retransmissions caused due to the frequent packets collisions. However, in both cases, the analytical results still provide an tight approximation. Similar results are also observed for log-normal shadowing radio model as shown in Fig. 4.6(b).

4.4 Summary

We have proposed an accurate mathematical model that analyzes the per-node communication traffic load, the dominant source of energy consumption, in a multi-hop wireless sensor network. Our results confirm that the traffic load of a node increases with the proximity to the sink. In addition, we discover that the radio characterization model has a significant impact on the traffic load pattern of sensors in the immediate vicinity of the sink. The ideal model leads to a volcano effect whereas the log-normal model causes a mountain peak shape. Our analytical model is validated by extensive simulations. Furthermore, the



(a) ideal radio



(b) log-normal radio ($\xi = 1$)

Figure 4.6: Traffic load simulation results using NS-2 ($l = 200$, $R = 50$)

simulations demonstrate that our results are also valid in realistic scenarios where the assumptions made for the analysis have been relaxed.

Models that accurately predict the traffic load of each sensor node make valuable contributions towards designing energy efficient sensor networks. For example, the per-node traffic load can be used to estimate the energy consumption of each sensor node, given that the energy expended in transmitting packets makes up a significant portion of a nodes' energy budget. The latter can then be used to predict the operational lifetime of the network, which is an important criteria for designing sensor networks. Further, the per-node energy consumption derived from our analysis can be useful in identifying **hotspots** in the network, i.e., regions where sensor nodes are expected to drain their energy at a faster rate (due to higher traffic load). Additional sensor nodes can be deployed at these hotspots to extend the operational lifetime of the network. In fact, the number of redundant nodes to be deployed at each of the hotspots can also be estimated using the knowledge of the per-node energy consumption. The per-node traffic load derived in our analysis can also be incorporated in MAC protocol design. In particular, the duty-cycle of each sensor node can be adjusted in accordance with the estimated traffic load, thereby improving the energy efficiency. We intend to investigate these applications in our future work.

Chapter 5

Geocasting without location information

In many wireless ad-hoc networking applications, a wireless node often needs to disseminate information to all other nodes within a target geographical distance from it. For example, first responders working in a large-scale disaster site often require broadcasting of warning, help, or discovery messages to other crew members within the geographic jurisdiction of the commanding team. In Intelligent Transportation System (ITS), road safety applications require vehicles to flood warning messages to other vehicles within an immediate neighborhood to prevent accidents or to warn of traffic hazards [40]. In a wireless sensor network, the sink node (base station) often need to send various types of query and command messages to all sensor nodes within a predetermined geographic region around it.

To support the aforementioned applications, researchers have proposed a new kind of packet forwarding primitive, called geocast [41]. In geocasting, the source broadcasts a packet specifying the target geographical region in the packet header. Any node receiving the packet rebroadcasts it only if its own location coordinates were within the region, ignores it otherwise. This way, the packet is quickly propagated to all nodes within the target geographical boundary. Packet propagation ceases as soon as the packet travels beyond the boundary.

While geocasting solves the problem of limiting information propagation within geographic boundaries, it assumes that every node will have access to precise location information at all times. This assumption is valid in general and is well backed up by the falling cost of location sensing hardware, e.g., GPS circuits, and advancements in GPS-less localization [42]. There are, however, many practical situations when a particular (set of) wireless node(s) may not have access to reliable location information. For example, for many tiny wireless sensing devices, e.g., MicaZ and TelosB [43], continuously running the GPS interface would deplete the battery prematurely. For vehicles inside a tunnel or in urban roads, access to one or more satellites may be temporarily blocked (note that GPS requires line-of-sight to four satellites). Similarly, a disaster recovery worker trying to discover human bodies under the rubbles may not be able to pick up GPS signals. Because of these, and many other unavoidable circumstances, it is practical to provision for alternate methods of forwarding so that geocasting can proceed in the event that reliable location information is absent.

We propose to use the well-known time-to-live (TTL) forwarding, which limits the propagation of a packet within a specified number of hops (TTL)[1] from the source. This is a popular forwarding technique used in both wireless ad-hoc networking, e.g., route discovery [18], and in wired networking, e.g., IPv4. TTL forwarding is based on a very simple concept. The source node broadcasts a packet with a specified TTL. Upon receiving the packet, each intermediate node decrements the TTL by one. If the TTL is still nonzero, and the node has not received the packet before from another node, the node rebroadcasts the packet. It drops the packet otherwise. A key feature of TTL forwarding, which is of particular interest to us, is that it does not require a node to know its location coordinates.

The main challenge in achieving geocasting with TTL forwarding is how to select the right TTL. This is especially challenging in the context of mobile ad-hoc

[1] TTL is a misnomer in the sense that it actually specifies the number of hops, not the time, a packet will live.

networking, where the node positions may not follow any regular pattern. A further complexity arises from the randomness in the radio propagation, which can create arbitrary connectivity (hops) between nodes irrespective of the geographical distance between them (i.e., geographical distance may not be true representative of hops or connectivity). Obviously, by "overprovisioning" the TTL, we can improve our chances of reaching all the nodes within the target geocast area, but it will be achieved at the expense of increased forwarding overhead in the network, which can be detrimental for resource constrained wireless networks.

The intent of this work is to conduct a systematic study exploring the suitability of using TTL forwarding to achieve geocasting in situations when some (or all) nodes in the geocast area do not have access to reliable location information. Our objective is to quantify, as a function of the TTL, the achievable coverage, i.e., the percentage of nodes in the geocast area that receive a copy of the original broadcast from the source, and the broadcast overhead, i.e., the total number of transmissions needed for each message to achieve the geocast.

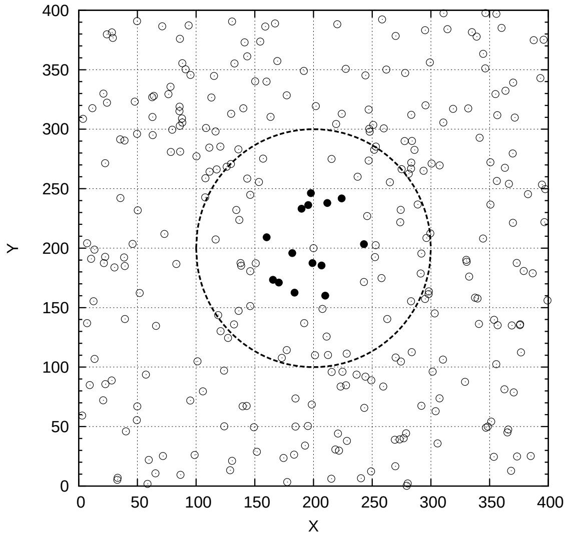
The achievable coverage and broadcast overhead given a TTL depends on the geographical distance covered by each hop (referred as hop distance). In our first part of work, we have analyzed the hop distance under the consideration of both ideal radio model and realistic radio model. Based on those results, we develop an analytical model to accurate quantify coverage and broadcast overhead.

The rest of this chapter is organized as follows. Section 5.1 presents the analytical model for computing coverage and forwarding overhead. Section 5.2 describes the implementation of TTL-based geocasting in the widely used ns-2 simulation platform. The performance results, obtained from the analytical model and the simulation experiments, are discussed in Section 5.3. We conclude the chapter in Section 5.4.

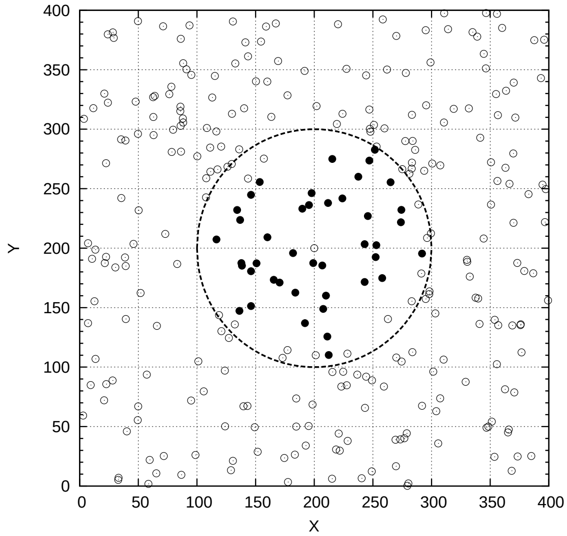
5.1 System Model

In this work, we focus on local geocasting where the goal is to disseminate messages to the nodes inside the geocast region, which is a circle of distance d centered at the message source (as shown in Fig. 5.1). In the proposed scheme, we use a predefined hop count as TTL value to restrict flooding within the desired area. The TTL-based flooding has been previously used in routing protocols, e.g. the expanding ring search in AODV [18]. In this mechanism, the source fills a flooding packet with a predefined TTL value and then broadcasts the packet. Upon receiving the packet, each intermediate node decrements the packet's TTL by one. If the TTL is nonzero and the node has not received the packet before, the node rebroadcasts the packet. Otherwise, the node drops the packet.

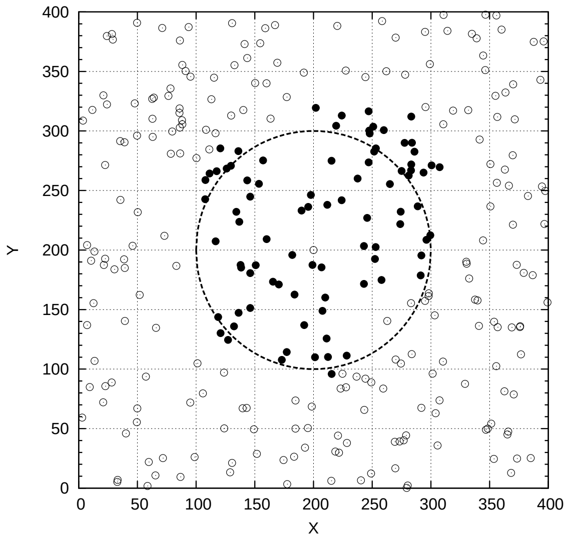
The estimation of appropriate TTL is vital in this scheme. There is a tradeoff between the coverage (percentage of nodes within geocast region receiving the packet) and broadcast overhead. For example, Fig. 5.1 illustrates this tradeoff where the TTL value varies from 1 to 4. In this example, the radio range is $50m$. The required geocast region is a circle with radius of $100m$, as depicted in the figure. The nodes that have received the flooding message are marked with dark solid circle. This figure shows that, selecting a small TTL value can reduce the broadcast overhead but may compromise the coverage of geocast region. For example, in Fig. 5.1(a) where $TTL = 1$ is used, the packet is only being broadcast once at the source. While the broadcast overhead is minimum, the percentage of nodes within the geocast region that receive the packet is only 22%. On the contrary, using a large value can ensure that all nodes will recieve the packet however may introduce unnecessary broadcasts for nodes beyond the geocast region. For example, in Fig. 5.1(d), the selction of $TTL = 4$ can cover all the nodes within the geocast region. However it incurs around 30 unnessary broadcasts at the nodes outside the geocast region.



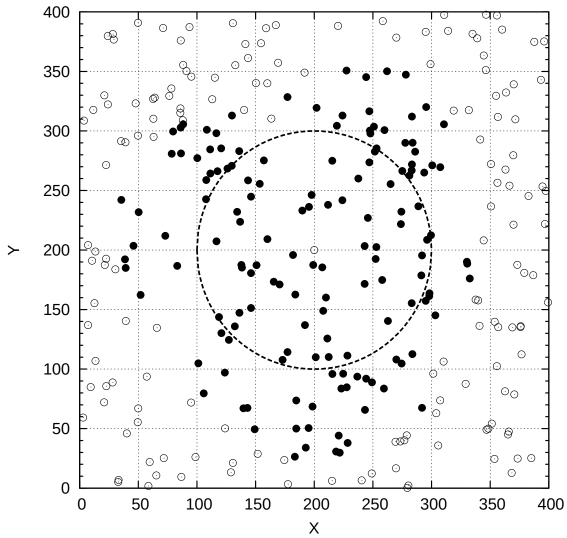
(a) TTL=1 (number of broadcast=1,coverage=22%)



(b) TTL=2 (number of broadcast=14,coverage=69%)



(c) TTL=3 (number of broadcast=42,coverage=97%)



(d) TTL=4 (number of broadcast=85,coverage=100%)

Figure 5.1: Illustrative example of TTL-based geocasting (R is 50m and geocast radius is 100m)

Given a distance d, we use $f(h|d)$ to denote the coverage that can be achieved by TTL value of h, i.e. the percentage of nodes within the distance radius d that can receive the geocast message by using h. Similar, we use $g(h|d)$ to denote the broadcast overhead incurred by using h, i.e. the average number of times each message is broadcast (aggregated among all nodes). In order to select an appropriate TTL value that balances the coverage and the broadcast overhead, it is necessary to accurately estimate $f(h|d)$ and $g(h|d)$. In doing so, the key step is to study how far a packet can progress at each broadcasting, referred to as *hop distance*. In this section, we first focus on analyzing the average hop distance (Section 5.1.1). Then we analyze the coverage (Section 5.1.2) and the broadcast overhead given the pair (d, h) (Section 5.1.3).

5.1.1 Analysis of Hop Distance

For mathematical tractability, we assume that node distribution follows a homogenous Poisson point process with a density of ρ. This distribution can approximate a large region with uniformly distributed nodes and has been widely used in analyzing multi-hop wireless ad-hoc networks [90, 91].

The hop distance is dependent on the network topology, which again is determined by the radio characteristics at the physical layer. We consider two popularly used radio models here so that we can study the impact of radio model on the proposed scheme. We first consider an ideal radio model where radio coverage of each node is a perfect circle with a radius of R. This radio model has been proved to be far from realistic [27, 28]. Therefore, we also consider a **log-normal shadowing** radio propagation model, which takes account of random signal fading observed in most wireless communication environments. The radio coverage of each node in this model is irregular, which resembles realistic situations [27, 28]. More formally, given a distance s that separates two nodes, the probability that these two nodes have a direct connection, referred as link probability [90], is,

$$P_{\wedge}(s) = \frac{1}{2}\left[1 - erf(\frac{10}{\sqrt{2}\xi} log_{10}\frac{s}{R})\right] \tag{5.1}$$

where R is referred to as **average radio range**, which is the radio range of a node in the absence of random fading. ξ is the fading randomness parameter and it is the ratio of path loss rate to the standard deviation of random fading. The

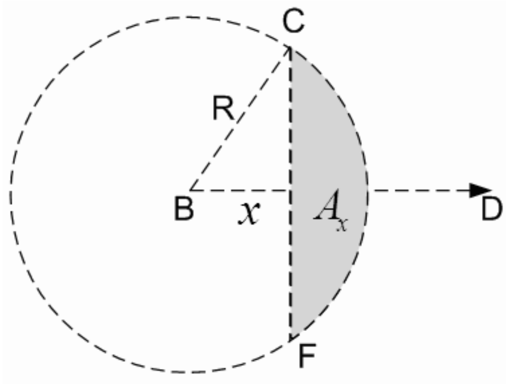


Figure 5.2: Illustration used to prove Theorem 7

function $erf(.)$ is defined as follows,

$$erf(z) = \frac{2}{\sqrt{\pi}} \int_0^z \exp(-x^2)dx \tag{5.2}$$

Based on geometric and probabilistic calculations, we can estimate the average hop distance for ideal radio model. The analytical result has exactly same form as the one presented in Theorem 4 in Section 3.4.3, where we have analyzed the converged value of average progress per hop for greedy routing. However, the proof is slightly different since we focus on broadcasting routing here. We restate the theorem and proof the theorem for broadcasting scenarios.

Theorem 7. *In ideal radio model, the average hop distance λ for each broadcasting is given by,*

$$\lambda = R\left[1 - \int_0^1 \exp(-\rho R^2(\arccos(t) - t\sqrt{1-t^2})dt)\right] \tag{5.3}$$

Proof:

Assume that a packet is currently at node B. Let X be the hop distance in the direction of BD (X is a random variable). When node B broadcasts a packet, all neighbors within its radio range can receive the packet. The hop distance in the direction of BD is the largest distance between B and its neighbors' projections on line BD. As shown in Fig. 5.2, the probability that the hop distance is x, is the probability that a neighbor is located along line CF and no neighbor exists in the area to the right of CF, referred to as area A_x. Equivalently, the cumulative probability that the hop distance is less than x, is the probability that no neighbor exists in the region A_x. We have,

$$F_X(x) = P(X < x) = Pr(\text{no nodes in region } A_x) \tag{5.4}$$

Recall that, we have assumed that the node distribution follows a homogeneous Poisson point process with density ρ. As a property of this assumption, the number of nodes in any region of area A follows a Poisson distribution with mean ρA. Therefore, the probability that no neighbor exists in the region A_x, is $\exp(-\rho A_x)$. The area A_x can be calculated as,

$$A_x = R^2 \arccos\frac{x}{R} - x\sqrt{R^2 - x^2} \tag{5.5}$$

Thus, the cumulative density function (cdf) of X is given by,

$$F_X(x) = \exp(-\rho A_x) = \exp(-\rho(R^2 \arccos\frac{x}{R} - x\sqrt{R^2 - x^2})) \tag{5.6}$$

The average hop distance λ follows,

$$\begin{aligned}
\lambda = \quad & E(X) = \int_0^R x f_X(x)dx = \int_0^R x dF_X(x) \\
= \quad & \big[xF_X(x)\big]_0^R - \int_0^R F_X(x)dx \\
= \quad & R - \int_0^R \exp(-\rho(R^2 \arccos\frac{x}{R} - x\sqrt{R^2 - x^2}))dx \\
\overset{x=Rt}{\Longrightarrow} \quad & R\left[1 - \int_0^1 \exp(-\rho R^2(\arccos(t) - t\sqrt{1-t^2}))dt\right]
\end{aligned} \tag{5.7}$$

Hence the theorem is proved. ♠

Next, we use a similar approach to analyze the average hop distance for log-normal shadowing radio model. The analytical result has same form as the one presented in Theorem 5 in Section 3.4.3. We have,

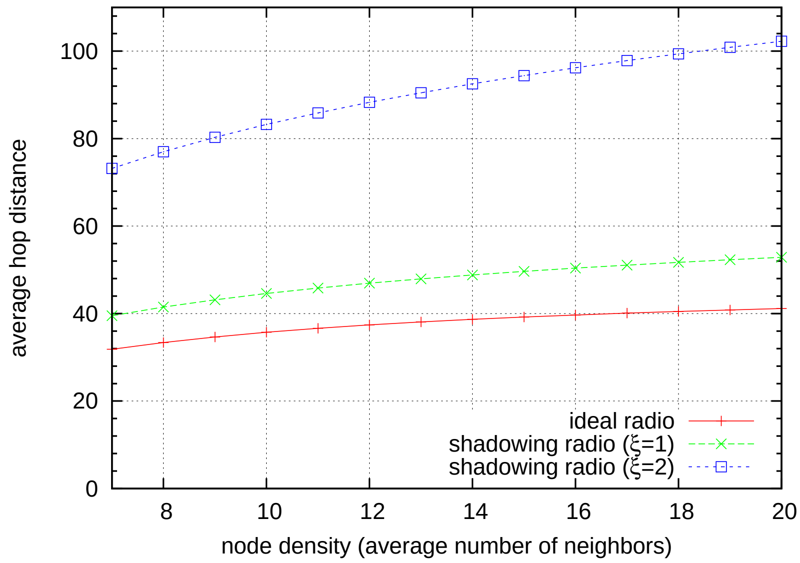


Figure 5.3: Average hop distance as a function of node density ($R = 50$)

Theorem 8. *Under the consideration of log-normal shadowing radio model, the average hop distance λ is given by,*

$$\lambda = R'\left(1 - \int_0^1 \exp(-\rho R'^2(\arccos(t) - t\sqrt{1-t^2})) \cdot g(t)dt\right) \tag{5.8}$$

where R' is the approximation of maximum radio range, and $g(t)$ is,

$$g(t) = \frac{10}{\sqrt{2\pi}\ln(10)\cdot\xi(\arccos t - t\sqrt{1-t^2})} \cdot \int_t^1 \frac{u^2\arccos\frac{t}{u} - t\sqrt{u^2-t^2}}{u}\exp(-(\frac{10}{\sqrt{2}\xi}log_{10}\frac{R'u}{R})^2)du \tag{5.9}$$

For the detailed proof, please refer to the Theorem 5 in Section 3.4.3. According to Theorem 7, hop distance is a function of radio range R and node density ρ for ideal radio model. For the case of log-normal shadowing radio, the hop distance is a function of average radio range R, node density ρ and fading randomness ξ. Fig. 5.3 illustrates an example of average hop distance as a function of node density for the different radio models assuming $R = 50m$ (here the node density is transformed to the average number of nodes within πR^2). It shows that, in all radio models, the hop distance increases with the node density. This is due to the fact that each node is more likely to find neighbors at the edge of radio coverage in a denser network. Thus, on average, the packet can progress further in each hop. Comparing different radio models, it is evident that the distance traveled in one hop is greater with the shadowing radio model. Further, this distance increases with an increase in the random fading, ξ. This is a direct result of the link probability distribution as illustrated in Fig. 3.4. Since the link probability curve has a longer tail for a larger value of ξ, each broadcast packet can cover a greater area and therefore has a longer hop distance.

5.1.2 Analysis of Coverage

Next, we analyze the coverage and broadcast overhead as a function of the TTL value. We assume a worst case scenario where all the nodes in the network do not have access to their location information. In such situation, we apply the TTL-based scheme to all the nodes. Recall that the coverage $f(h|d)$ is defined as the percentage of nodes within the geocast radius d that can receive the geocast message by the TTL value of h. To estimate $f(h|d)$, we need to know the number of nodes that can receive the packet by using h and the total number of nodes within the geocast region. Given a TTL value h of a packet and the average hop distance λ, the actual distance that the packet can travel on average is λh. Since the node distribution follows a Poisson point process, the average number of nodes within distance λh is given by $\rho\pi\lambda^2 h^2$, which is also the number of nodes that receive the packet. Note that, the total number of nodes within the geocast region is $\rho\pi d^2$. Therefore, the coverage can be calculated as $\frac{\lambda^2 h^2}{d^2}100\%$. Note that, this ratio can be possibly greater than 100% if a very large value of h is used. Hence, we limit the maximum value to 100%, in accordance with our definition of coverage. We have,

$$f(h|d) = \min\left\{\frac{\lambda^2 h^2}{d^2}, 1\right\} \cdot 100\% \tag{5.10}$$

5.1.3 Analysis of Broadcast Overhead

Now, we calculate the broadcast overhead $g(h|d)$, i.e. the average number of times that each message is broadcast (aggregated among all nodes) by using h as the TTL value. Note that, when a node receives a packet with TTL value of 1, this node is the last recipient of this packet and therefore does not rebroadcast this message. Thus, only nodes that are within hop count $h-1$ from the source are involved in rebroadcasting the message (if $h=1$, only the source needs to broadcast). Given λ, the average number of nodes that broadcast the packet, i.e. the broadcast overhead, is given by,

$$g(h|d) = \rho\pi\lambda^2(h-1)^2 \tag{5.11}$$

Note that, the above analysis model is independent of the radio model under consideration. One simply has to substitute the appropriate hop distance λ as derived in Theorem 1 and 2 for the radio model under consideration.

5.2 Practical Considerations in Implementing TTL-based Geocasting

Next, we proceed to perform simulation-based evaluations of TTL-based geocasting. We use the popular NS2 simulator for our evaluations. In this section, we highlight the challenges encountered in implementing TTL-based geocasting in NS2 and discuss practical solutions to overcome the same. This discussion will be of particular interest to network practitioners who wish to instantiate the ideas presented in this chapter in real-world systems.

Recall that, in TTL-based geocasting, a node will immediately broadcast a packet that it is has received, if the TTL is non-zero after being decremented. In a practical scenario, this is very likely to cause significant collisions, since neighbouring nodes that receive a broadcast packet will synchronously rebroadcast the packet. Our simulations have indeed confirmed that this synchronization leads to significant collisions. To avoid this problem, we introduce a small random delay at each node, prior to broadcasting. We assume that this delay is a random value which is uniformly selected from a range (a, b). Note that, a similar approach has been used in other protocols which rely on the broadcast primitive. For example, in the AODV routing protocol, each node waits for a random duration prior to broadcasting the periodic HELLO message to its neighbors.

Note that, our mathematical model does not account for this additional random delay incurred at each node. In our analysis, we implicitly assume that a node will always receive the first copy of the geocast packet along the shortest path between the source and the node. Subsequently, any packets arriving along the other paths (i.e. not along the shortest path) are copies, which are discarded. More importantly, this allows us determine the geocast coverage $f(h|d)$ that can be achieved when the TTL value is h (see Equation 5.10). However, with the addition of the random delay at each node, it may be possible that the packet that follows the shortest path to a node may no longer be the first copy to reach the node. Subsequently, this packet will be dropped as it is a copy. As a result, the geocast coverage achieved in practice may be lower than that determined from the analytical model.

Hence, it is important that the values of the range (a, b) are chosen so as to ensure that there is a minimal chance of the occurrence of the above problem, while still reducing collisions due to node synchronization. In our simulations (detailed in the subsequent section), we find that choosing a range such that $b \leq 2a$ meets these requirements. We have used a range of $(4ms, 8ms)$ in our simulations.

5.3 Performance Results

In this section we present results from our simulation-based evaluations. Our goal is two-fold. First, we attempt to validate our analytical model. Second, we seek to compare the performance of the TTL-based approach to location-based geocasting in different situations.

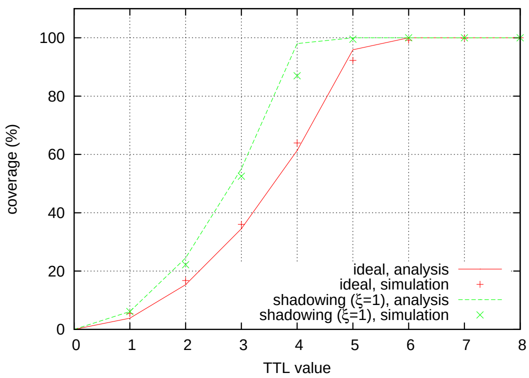
We consider three different scenarios. In the first scenario, we simulate a random network topology and validate our analysis by comparing the numerical results derived in Sections 5.1.2 and 5.1.3 with results from the simulations. In our analysis, for mathematical tractability, we assumed an idealized MAC, which does not result in any packet collisions. However, in all the simulations we have implemented the 802.11 MAC at the link layer. This allows us to investigate if our analytical results still hold in realistic settings. In addition, we also compare the performance of our TTL-based approach with location-based geocasting, assuming that all nodes have perfect knowledge of their location. In the second scenario, we investigate if the TTL-based approach can complement traditional location-based geocasting. We consider the same random network topology as in the first scenario, but assume that a variable percentage of nodes are unaware of their location coordinates. These nodes employ TTL-based geocasting, while all other nodes that know their geographical coordinates use location-based geocasting. In the final scenario, we repeat the above experiment for a realistic vehicular ad hoc network using mobility traces of a metropolitan public transport bus network.

5.3.1 Comparison with Location-based Geocasting

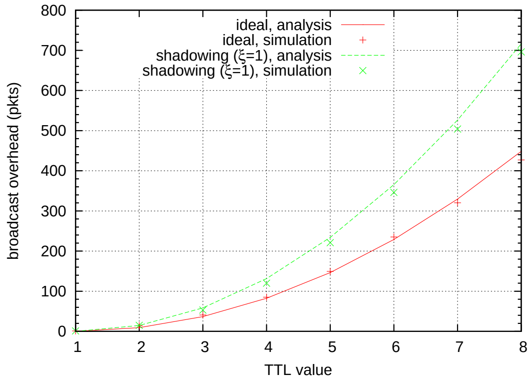
In the first scenario, we evaluate the performance of the proposed TTL-based geocasting scheme. In particular, we are interested in determining the coverage achieved and the corresponding broadcast overhead as a function of TTL. We also compare the simulation results with the corresponding results from our analysis in Section 5.1. Finally, we compare our scheme with location-based geocasting.

We consider a network of dimension 500m $\times$ 500m and assume that the node density, ρ is 0.0019, which results in a total of 475 nodes. The geocast source is assumed to be located at the center of the network. We simulate the IEEE 802.11b MAC at the link layer. We simulate the following radio models (to be consistent with our analysis in Section 5.1): (i) ideal model (i.e., two ray ground) - wherein the received power of a packet depends on the Euclidean distance between the sender and the receiver, the path loss and the transmission power, (ii) shadowing model - wherein the received power is also affected by an additional parameter, random fading. In both cases, the packet is assumed to be successfully received only if the received power is greater than a threshold, $7.69113e-08W$. The transmit power is set to $0.281W$ and the path loss exponent and standard deviation for the shadowing model, are both set to 2. The radio range, R for each node is set to 50m and the geocast distance, d is assumed to be 200m. We vary the TTL value from 1 to 8 and observe its impact on the geocast coverage and broadcast overhead in Fig. 5.4(a) and Fig. 5.4(b), respectively. Here each point of result is the average value of 20 repeated simulations (i.e. 20 random generated networks). The confidence intervals of simulation results are very narrow. For the coverage, the confidence intervals are within $\pm 1\%$ from the corresponing average value. For broadcast overhead, the confidence interval are within ± 8 from the corresponing average value.

Fig. 5.4(a) shows that the coverage increases rapidly with an increase in the TTL and finally converges to the maximum value of 100% after a certain TTL value. Fig. 5.4(a) also shows that the radio model has a significant impact on the coverage. Particularly, with the realistic shadowing radio model, the coverage converges to its maximum value for a lower value of the TTL than that of the ideal radio model. As expected, the broadcast overhead consistently increases with an



(a) coverage



(b) broadcast overhead

Figure 5.4: Comparing analytical and simulation results (assuming a geocast radius of 200m)

increase in the TTL. The results imply that beyond a certain TTL threshold, any further increase in the TTL value does not improve the coverage (beyond 100%), but instead merely introduces additional overhead. In order to achieve maximum coverage, it is thus prudent to use this threshold value as the TTL. The threshold (for either radio model) can be estimated as $\lceil \frac{d}{\lambda} \rceil$. This is because, the average distance traversed in one hop is λ, which implies that the total number of hops to cover a distance, d is d/λ.

To obtain the corresponding analytical results, we substitute the parameters for the scenario under consideration in the appropriate equations (Equation (5.10) and (5.11), derived in Section 5.1. The analytical results are plotted alongside the simulation results in Figs. 5.4(a) and 5.4(b). Observe that the results match closely (for both radio models), thus validating our analytical model. Note again, the difference between the results for the ideal and log-normal model. An important lesson to be learnt here is that the results from the ideal scenario are not suitable in realistic situations. For example, the TTL that achieves maximum coverage under the ideal model would be more than what is required in a practical scenario (which is consistent with the log-normal model) and would thus create excessive broadcast overhead.

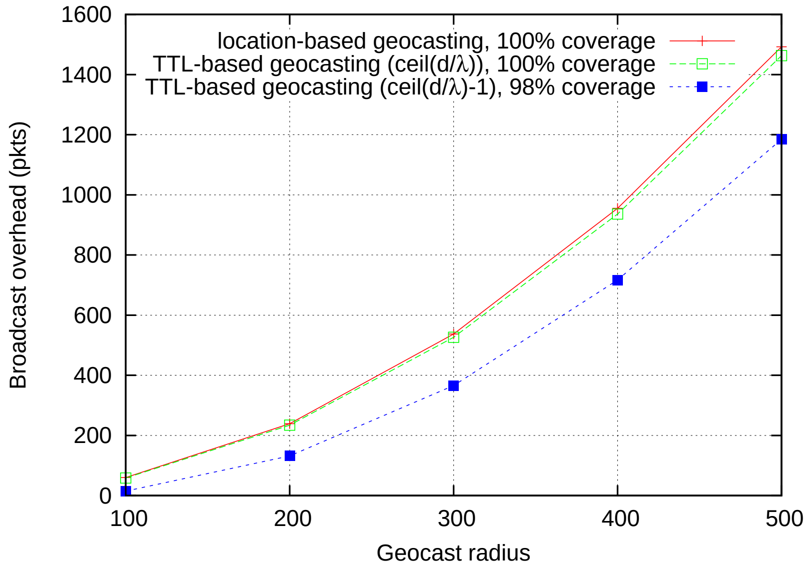


Figure 5.5: The comparison of TTL-based geocasting with location-based geocasting as a function of geocast radius ($R = 50m, \xi = 1$)

Next, we compare the performance of our TTL-based scheme with location-based geocasting. We use analytical results in this comparison. In particular, we focus on the broadcast overhead of the two schemes for achieving 100% coverage. As discussed above, we select the TTL value for achieving 100% coverage using the formula $\lceil \frac{d}{\lambda} \rceil$. The broadcast overhead given the geocast radius, d can be calculated using Equation (5.11). We also compute the broadcast overhead resulting from using a TTL value that is one less than this threshold (i.e., $\lceil \frac{d}{\lambda} \rceil - 1$). This choice of the TTL value still achieves 98% coverage (according to Equation (5.10). In the location-based geocasting scheme, since we assume that all nodes have perfect knowledge of their location, the broadcast overhead is simply equal to the total number of nodes contained within the geocast region. This is because, each node that is located within the target geocast region will broadcast the packet exactly once. The broadcast overhead is thus computed as $\pi \rho d^2$. We assume that the physical layer is represented by the log-normal shadowing model.

Fig. 5.5 plots the broadcast overhead as a function of the geocast radius for

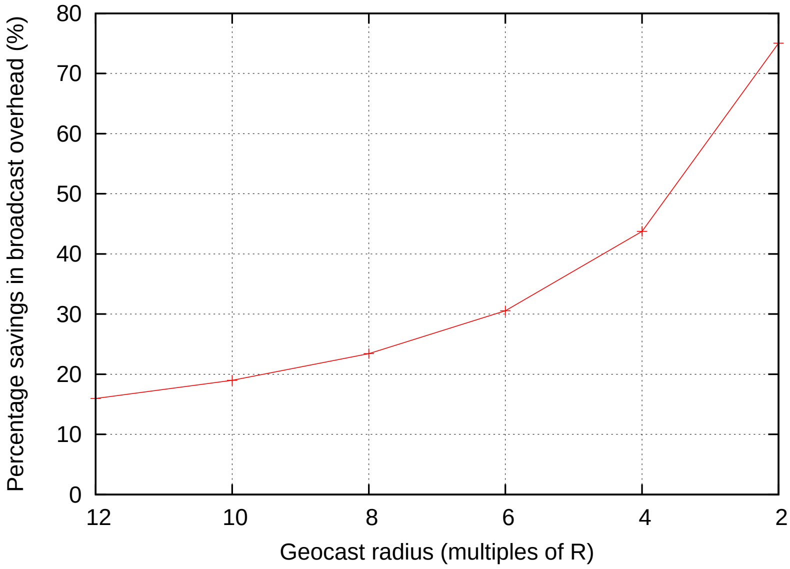


Figure 5.6: Percentage savings in broadcast overhead when using a TTL value one less than $\lceil \frac{d}{\lambda} \rceil$

the aforementioned schemes. One can readily observe that the overhead incurred by the proposed TTL-based approach to achieve 100% coverage closely matches that of location-based geocasting. Interestingly, if we choose the TTL to be one less than the threshold, $\lceil \frac{d}{\lambda} \rceil$, then the broadcast overhead can be reduced significantly, while still ensuring that the message is received by 98% of the nodes. Fig. 5.6 shows that the percentage savings in broadcast overhead increases exponentially with decreasing geocast radius and the actual savings can be as high as 75% when the geocasting radius is twice the radio range. The exponential increase in broadcast overhead can be explained as follows. When we use a TTL one smaller than the required, we are saving the broadcast near the periphery of geocast region. With the decrease of geocast radius, the ratio of the periphery area to the whole geocast area becomes larger, boosting the percentage savings in broadcast.

To fully appreciate the coverage-overhead tradeoff of TTL-based geocasting, let us further analyze the 2% loss in coverage which yields upto 75% reduction in broadcast overhead. As illustrated in Fig. 5.1(c), the 2% of the nodes, which do not receive a copy of the packet, are all found to be located along the edge of the geocast region. For applications where the significance of packet delivery is proportional to the distance from the source, e.g., in vehicular crash avoidance applications (vehicles farther from the source have less possibility of crashing) [98], the coverage-overhead tradeoff can be a very useful feature of a geocasting protocol.

5.3.2 Evaluating the Coexistence of TTL-based and Location-based Geocasting

In this scenario, we seek to investigate if TTL-based geocasting can work hand-in-hand with location-based geocasting. We use the same simulation parameters as in Section 5.3.1. We assume that a certain variable fraction of the nodes are unaware of their location coordinates. These nodes employ our TTL-based approach, while all other nodes in the network, which know their position coordinates use location-based geocasting. One can readily envision that such situations often arise in realistic ad hoc networks. For example, when vehicles are inside a tunnel or in urban roads, access to one or more GPS satellites may be temporarily blocked. We vary the percentage of nodes that do not have location information from 0% to 100% and observe the effect on the geocast coverage and broadcast overhead. The results are presented in Fig. 5.7, with the left axis reflecting the coverage and the right axis denoting the overhead. Note that, at the two extremes (i.e. 0% and 100%) all nodes homogeneously use location-based and TTL-based geocasting, respectively. We assume that the geocast radius is 200m and the TTL value is equal to the threshold, $\lceil \frac{d}{\lambda} \rceil$.

The graph illustrates that, even when a significant fraction of the nodes aren't aware of their location, it is still possible to maintain near 100% coverage

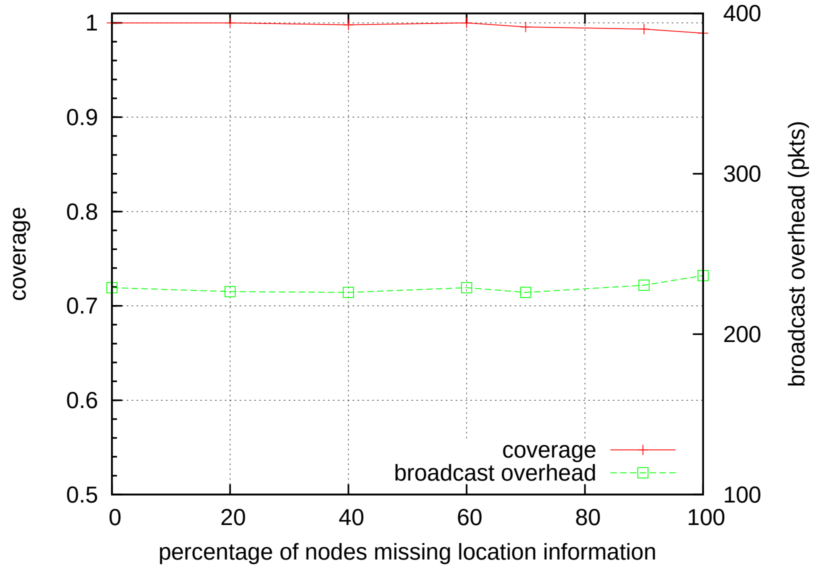


Figure 5.7: The performance of geocasting when some nodes are missing location information in a random network $(R = 50, \xi = 1, d = 200m)$

without any noticeable increase in the broadcast overhead, by employing TTL-based geocasting. This implies, that our TTL-based approach is a simple yet effective strategy for achieving geocasting in practical multi-hop wireless networks.

5.3.3 Vehicular Network Scenario

In the previous simulations, we assumed a random network topology. In this section, we relax this assumption and consider a more realistic scenario. We simulate a vehicular ad hoc network generated from the movement traces of public transportation buses in a metropolitan area. We have used location traces from the King County Metro bus system in Seattle, Washington [97]. This transport network consists of close to 1160 buses plying over 236 distinct routes and covering an area of 5100 square kilometers. The traces were collected over a three week period in November 2001. The traces are based on location update messages sent by each bus. Each bus logs its current location using an Automated Vehicle Location system [110], its bus ID and route ID along with a timestamp. The typical update frequency is 30 seconds. We have not simulated the entire bus network. This is because the network is quite sparse (for example, only 1 or 2 buses) in several regions of the city, which would not lead to meaningful results. Rather, we focus on the business district, which has a consistently high density of buses. In particular, we focus on a rectangular region of size 4km x 7km in the central business district. The duration of this trace spans 30 minutes. We assume that the radio range of each node is 1000m, which is consistent with that for DSRC [98] and the results from [99].

In addition, we simulate a practical road safety application [111]. We assume that municipal workers are conducting road maintenance at certain locations in the business district. The maintenance sites are equipped with wireless devices that periodically geocast safety messages within the immediate neighborhood of the work zone to warn drivers of the roadwork and revised speed restrictions. We assume that the messages are transmitted periodically every 10 seconds and that the geocast radius is 3000m. As in Section 5.3.2, we vary the percentage of nodes that do not have location information from 0% to 100%. The nodes without location coordinates employ TTL-based geocasting, while all other nodes use location-based geocasting. The average node density for the network under consideration is found to be $1.03 * 10^{(-5)}$. We assume the realistic log-normal shadowing radio model at the physical layer. The TTL value according to $\lceil \frac{d}{\lambda} \rceil$ is 3.

Fig. 5.8 plots the coverage (left axis) and broadcast overhead (right axis) as a function of the percentage of nodes that do not know their location information. The graph again confirms that, even when a large fraction of the nodes don't have their location coordinates, TTL-based geocasting ensures that there is no drop in the geocast coverage. However, the TTL-based scheme reduces the broadcast overhead in the network. In fact, the greater the number of nodes employing

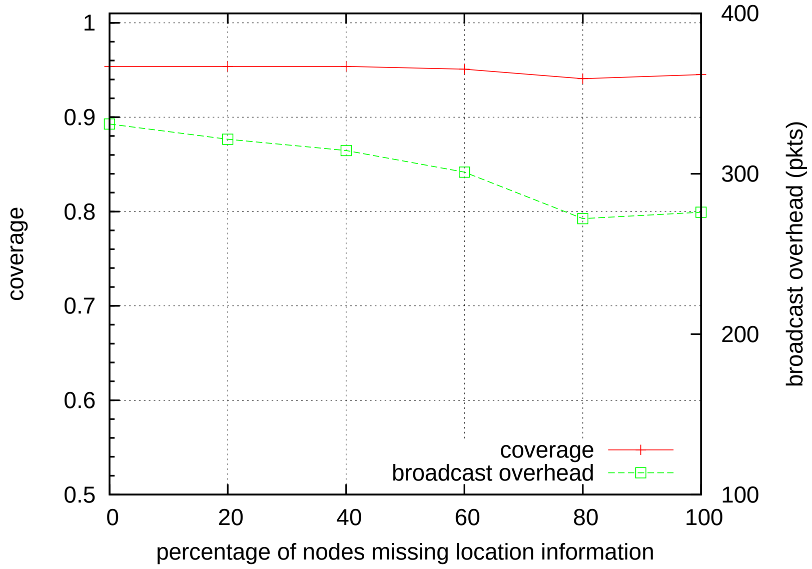


Figure 5.8: The performance of geocasting when some nodes are missing location information in a vehicular network $(R = 1000m, \xi = 1, d = 3000m)$

TTL-based geocasting, the more the decrease in the overhead. The reason for this is that, in the TTL-based scheme, the last hop recipients of a packet (i.e., when TTL=1) do not rebroadcast the packet. These last hop recipients are often located near the edge of the geocast region. Due to the non-uniform distribution of the nodes, we find that more nodes are located near the edge of the geocast region than in the center. Therefore, the large number of nodes near the edge do not broadcast the geocast packet further if they are the last hop recipients. However, for the nodes that employ location-based scheme, as long as the node is within the geocast region (including nodes located near the edge), it has to broadcast the packet. Therefore, the broadcast overhead reduces as the percentage of nodes in the network that use TTL-based geocasting increases.

5.4 Summary

We have motivated the need for wireless packet forwarding techniques that can achieve geocast in the event that some nodes lose access to their location coordinates. We have shown that TTL forwarding, which does not use location information, can be used to achieve geocasting without sacrificing performance in terms of coverage and forwarding overhead. We have demonstrated that, for known network density and radio propagation model, the TTL value required to achieve effective and efficient geocast can be derived analytically. Our analytical model has been verified using simulation. We have demonstrated that for applications where the significance of packet delivery decreases with distance from the source (e.g., vehicular crash avoidance), TTL-based geocasting provides attractive tradeoff between coverage and forwarding overhead. The overhead can be significantly reduced by accommodating non-delivery of geocast messages to a tiny fraction of total node population, all located near the boundary of the geocast area (furthest from the source).

Chapter 6

Mobility and Traffic Adaptive Position Update for Geographic Routing

With the growing popularity of positioning devices (e.g. GPS) and other localization schemes [44], geographic routing protocols are becoming an attractive choice for use in mobile ad hoc networks [29], [33], [54], [55]. The forwarding strategy employed in the geographic routing protocols requires the following information: (i) the position of the final destination of the packet and (ii) the position of a node's neighbors. The former can be obtained by querying a *location service* such as the Grid Location System (GLS) [49] or Quorum [50]. To obtain the latter, each node exchanges its own location information (obtained using GPS or the localization schemes discussed in [44]) with its neighboring nodes. This allows each node to build a local map of the nodes within its vicinity, often referred to as the **local topology**.

However, in situations where nodes are mobile or when nodes often switch off and on, the local topology rarely remains static. Hence, it is necessary that each node broadcasts its updated location information to all of its neighbors. These location update packets are usually referred to as **beacons**. In most geographic routing protocols (e.g. GPSR [29], GeoCast [41], [51], [52]), beacons are broadcast periodically for maintaining an accurate neighbor list at each node.

Position updates are costly in many ways. Each update consumes node energy, wireless bandwidth, and increases the risk of packet collision at the medium access control (MAC) layer. Packet collisions cause packet loss which in turn affects the routing performance due to decreased accuracy in determining the correct local topology (a lost beacon broadcast is not retransmitted). A lost data packet does get retransmitted, but at the expense of increased end-to-end delay. Clearly, given the cost associated with transmitting beacons, it makes sense to adapt the frequency of beacon updates to the node mobility and the traffic conditions within the network, rather than employing a static periodic update policy. For example, if certain nodes are frequently changing their mobility characteristics (speed and/or heading), it makes sense to frequently broadcast their updated position. However, for nodes that do not exhibit significant dynamism, periodic broadcasting of beacons is wasteful. Further, if only a small percentage of the nodes are involved in forwarding packets, it is unnecessary for nodes which are located far away from the forwarding path to employ periodic beaconing because these updates are not useful for forwarding the current traffic.

In this work, we propose a novel beaconing strategy for geographic routing protocols called **Adaptive Position Updates strategy (APU)** [53]. Our scheme eliminates the drawbacks of periodic beaconing by adapting to the system variations. APU incorporates two rules for triggering the beacon update process. The first rule, referred as **Mobility Prediction (MP)**, uses a simple mobility prediction scheme to estimate when the location information broadcast in the previous beacon becomes inaccurate. The next beacon is broadcast only if the predicted error in the location estimate is greater than a certain threshold, thus tuning the update frequency to the dynamism inherent in the node's motion.

The second rule, referred as **On-Demand Learning (ODL)**, aims at improving the accuracy of the topology along the routing paths between the communicating nodes. ODL uses an on-demand learning strategy, whereby a node broadcasts beacons when it overhears the transmission of a data packet from a **new** neighbor in its vicinity. This ensures that nodes involved in forwarding data packets maintain a more up-to-date view of the local topology. On the contrary, nodes that are not in the vicinity of the forwarding path are unaffected by this rule and do not broadcast beacons very frequently.

We model APU to quantify the beacon overhead and the local topology accuracy. We apply the hop count analysis results derived in chapter 3 to estimate the average hop count of a communication pair in a mobile ad-hoc network, based on which we are able to analyze the beacon overhead incurred in APU scheme. The local topology accuracy is measured by two metrics, **missing neighbor ratio** and **false neighbor ratio**. The former measures the percentage of new neighbors a forwarding node is not aware of but that are actually within the radio range of the forwarding node. On the contrary, the latter represents the percentage of obsolete neighbors that are in the neighbor list of a node, but have already moved out of the node's radio range. Our analytical results are validated by extensive simulations.

In the first set of simulations, we evaluate the impact of varying the traffic load and mobility dynamics on the performance of APU and also compare it with periodic beaconing and two recently proposed updating schemes: distance-based and speed-based beaconing [68]. In addition to the aforementioned metrics, we also evaluate the effect of the beaconing strategies on the routing performance by simulating GPRS (Greedy Perimeter Stateless Routing) [29]. We find that when the mobility characteristics of a node do not change significantly (i.e. when the average speed is 5 m/s) and under light traffic load (5 traffic flows), APU reduces the beacon overhead by 80% compared to that of PB, but achieves a similar high performance in terms of packet delivery ratio and average end-to-end delay. When the node mobility is highly dynamic (i.e. when the average speed is 25m/s) and under heavy traffic load (30 traffic flows), APU has similar beacon overhead as that of PB, but achieves considerably better packet delivery ratio and average end-to-end delay. The simulation results also show that the distance-based and speed-based beaconing strategies can reduce the beacon overhead under low dynamism. However, when the nodes frequently change their motion parameters, the performance of these schemes degrades substantially as compared with APU.

In the second set of simulations we evaluate the performance of the beaconing strategies in a real-world vehicular scenario using realistic movement patterns of buses in a metropolitan city. The results indicate that APU significantly reduces beacon overhead without having any noticeable impact on the data delivery rate. One of the reasons for this is that beacons generated in APU are more concentrated along the routing paths, while the beacons in all other schemes are more scattered in the whole network. As a result, in APU, the nodes located in the hotspots, which are responsible for forwarding most of the data traffic in the network have an up-to-date view of their local topology, thus resulting in improved performance.

The rest of this chapter is organized as follows. A detailed description of the APU scheme is provided in Section 6.1, followed by a comprehensive theoretical analysis in Section 6.2. Section 6.3 presents a simulation-based evaluation highlighting the performance improvements achieved by APU in comparison with other schemes. Finally, Section 6.4 summaries this chapter.

6.1 Adaptive Position Update (APU)

We begin by listing the assumptions made in our work: (1) all nodes are aware of their own position and velocity, (2) all links are bi-directional, (3) the beacon updates include the current location and velocity of the nodes, and (4) data packets can piggyback position and velocity updates and all one-hop neighbors operate in the promiscuous mode and hence can overhear the data packets.

Upon initialization, each node broadcasts a beacon informing its neighbors about its presence and its current location and velocity. Following this, in most geographic routing protocols such as GPSR, each node periodically broadcasts its current location information. The position information received from neighboring beacons is stored at each node. Based on the position updates received from its neighbors, each node continuously updates its local topology, which is represented as a neighbor list. Only those nodes from the neighbor list are considered as possible candidates for data forwarding. Thus, the beacons play an important part in maintaining an accurate representation of the local topology.

Instead of periodic beaconing, APU adapts the beacon update intervals to the mobility dynamics of the nodes and the amount of data being forwarded in the neighborhood of the nodes. APU employs two mutually exclusive beacon triggering rules, which are discussed in the following.

6.1.1 Mobility Prediction (MP) Rule

This rule adapts the beacon generation rate to the frequency with which the nodes change the characteristics that govern their motion (velocity and heading). The motion characteristics are included in the beacons broadcast to a node's neighbors. The neighbors can then track the node's motion using simple linear motion equations. Nodes that frequently change their motion need to frequently update their neighbors, since their locations are changing dynamically. On the contrary, nodes which move slowly do not need to send frequent updates. A periodic beacon update policy cannot satisfy both these requirements simultaneously, since a small update interval will be wasteful for slow nodes, whereas a larger update interval will lead to inaccurate position information for the highly mobile nodes.

In our scheme, upon receiving a beacon update from a node i, each of its neighbors, denoted by the set $N(i)$, records its current position and velocity and periodically track node i's location using a simple prediction scheme based on linear kinematics (discussed below). Based on this position estimate the neighbors $N(i)$, check whether node i is still within their transmission range and update their neighbor list accordingly. The goal of the MP rule is to send the next beacon update from node i when the error between the predicted location in $N(i)$ and i's actual location is greater than an acceptable threshold. To achieve this, node i must keep track of its own predicted location as in its neighbors, $N(i)$.

We use a simple location prediction scheme based on the physics of motion to esimate a node's current location. Note that, in our discussion we assume that the nodes are located in a two-dimensional coordinate system with the location indicated by the x and y coordinates. However, this scheme can be easily extended to a three dimensional coordinate system. Table 6.1 illustrates the notations used in the rest of this discussion.

Table 6.1: Notations for Mobility Prediction

Variables	Definition
(X_l^i, Y_l^i)	The coordinate of node i at time T_l (included in the previous beacon)
(V_x^i, V_y^i)	The velocity of node i along the direction of the x and y axes at time T_l (included in the previous beacon)
T_l	The time of the last beacon broadcast
T_c	The current time
(X_p^i, Y_p^i)	The predicted position of node i at the current time

As shown in Fig. 6.1, given the position of node i and its velocity along the x and y axes,at time T_l, its neighbors, $N(i)$ can estimate the current position of

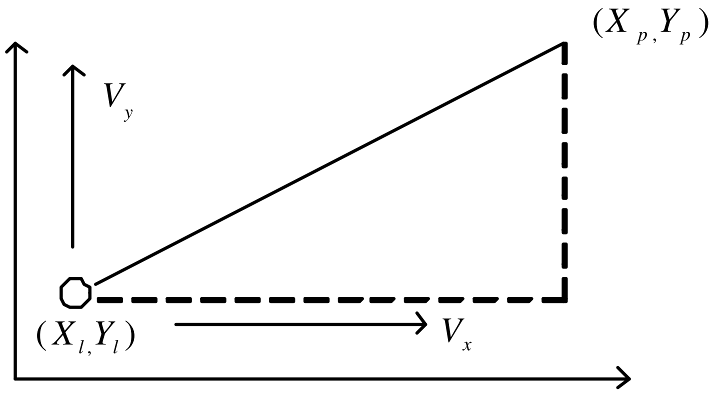


Figure 6.1: An example of mobility prediction

i, by using the following equations:

$$X_p^i = X_l^i + (T_c - T_l) * V_x^i$$
$$Y_p^i = Y_l^i + (T_c - T_l) * V_y^i \tag{6.1}$$

Note that, here (X_l^i, Y_l^i) and (V_x^i, V_y^i) refers to the location and velocity information that was broadcast in the previous beacon from node i. Node i uses the same prediction scheme to keep track of its predicted location among its neighbors. Let $(X_a,\ Y_a)$, denote the actual location of node i, obtained via GPS or other localization techniques. Node i then computes the deviation D_{devi}^i as follows:

$$D_{devi}^i = \sqrt{(X_a^i - X_p^i)^2 + (Y_a^i - Y_p^i)^2} \tag{6.2}$$

If the deviation is greater than a certain threshold, know as the **Acceptable Error Range (AER)**, it acts as a trigger for node i to broadcast its current location and velocity as a new beacon.

The MP rule, thus, tries to maximize the effective duration of each beacon, by broadcasting a beacon only when the predicted position information based on the previous beacon becomes inaccurate. This extends the effective duration of the beacon for nodes with low mobility, thus reducing the number of beacons. Further, highly mobile nodes can broadcast frequent beacons to ensure that their neighbors are aware of the rapidly changing topology.

6.1.2 On-Demand Learning (ODL) Rule

The MP rule solely, may not be sufficient for maintaining an accurate local topology. Consider the example illustrated in Fig. 6.2, where node A moves from $P1$ to $P2$ at a constant velocity. Now, assume that node A has just sent a beacon while at $P1$. Since node B did not receive this packet, it is unaware of the existence of node A. Further, assume that the AER is sufficiently large such that when node A moves from $P1$ to $P2$ the MP rule is never triggered. However, as seen in Fig. 6.2 node A is within the communication range of B for a significant portion of its motion. Even then, neither A nor B will be aware of each other. Now, in situations where neither of these nodes are transmitting data packets, this is perfectly fine since they are not within communicating range once A reaches $P2$. However, if either A or B was transmitting data packets, then their local topology will not be updated and they will exclude each other while selecting the next hop node. In the worst-case, assuming no other nodes were in the vicinity, the data packets would not be transmitted at all.

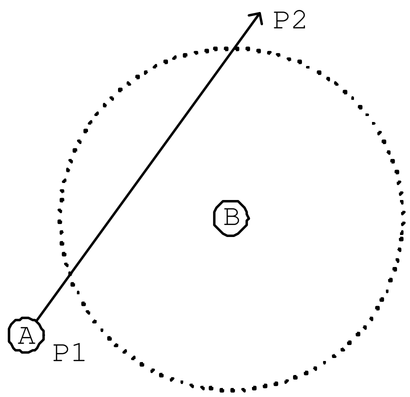


Figure 6.2: An example illustrating a drawback of the MP rule

Hence, it is necessary to devise a mechanism, which will maintain a more accurate local topology in those regions of the network where significant data forwarding activities are on-going. This is precisely what the **On-Demand Learning (ODL)** rule aims to achieve. As the name suggests, a node broadcasts beacons **on-demand**, i.e. in response to data forwarding activities that occur in the vicinity of that node. According to this rule, whenever a node overhears a data transmission from a **new** neighbor, it broadcasts a beacon as a response. By a **new** neighbor, we imply a neighbor who is not contained in the neighbor list of this node. In reality, a node waits for a small random time interval before responding with the beacon to prevent collisions with other beacons. Recall that, we have assumed that the location updates are piggybacked on the data packets and that all nodes operate in the promiscuous mode, which allows them to overhear all data packets transmitted in their vicinity. In addition, since the data packet contains the location of the final destination, any node that overhears a data packet also checks its current location and determines if the destination is within its transmission range. If so, the destination node is added to the list of neighboring nodes, if it is not already present. Note that, this particular check incurs zero cost, i.e. no beacons need to be transmitted.

We refer to the neighbor list developed at a node by virtue of the initialization phase and the MP rule as the **basic** list. This list is mainly updated in response to the mobility of the node and its neighbors. The ODL rule allows active nodes that are involved in data forwarding to enrich their local topology beyond this basic set. In other words, a **rich** neighbor list is maintained at the nodes located in the regions of high traffic load. Thus the rich list is maintained only at the active nodes and is built reactively in response to the network traffic. All inactive nodes simply maintain the basic neighbor list. By maintaining a rich neighbor list along the forwarding path, ODL ensures that in situations where the nodes involved in data forwarding are highly mobile, alternate routes can be easily established without incurring additional delays.

Fig. 6.3(a) illustrates the network topology before node A starts sending data to node P. The solid lines in the figure denote that both ends of the link

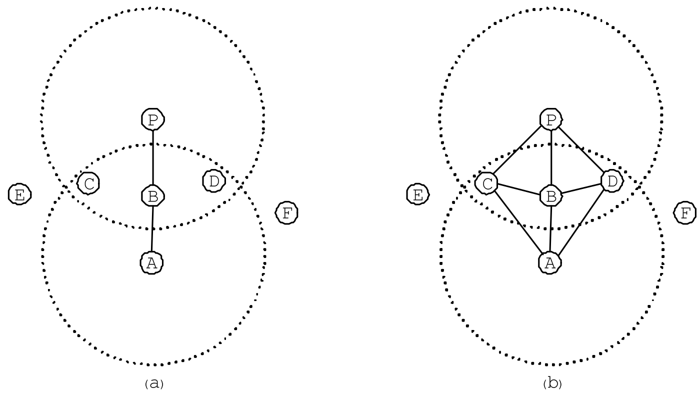


Figure 6.3: An example illustrating the ODL rule

are aware of each other. The initial possible routing path from A to P is A-B-P. Now, when source A sends a data packets to B, both C and D receive the data packet from A. As A is a new neighbor of C and D, according to the ODL rule, both C and D will send back beacons to A. As a result, the links AC and AD will be discovered. Further, based on the location of the destination and their current locations, C and D discover that the destination P is within their one-hop neighborhood. Similarly when B forwards the data packet to P, the links BC and BD are discovered. Fig. 6.3(b) reflects the enriched topology along the routing path from A to P.

Note that, though E and F receive the beacons from C and D, respectively, neither of them respond back with a beacon. Since E and F do not lie on the forwarding path, it is futile for them to send beacon updates in response to the broadcasts from C and D. In essence, ODL aims at improving the accuracy of topology along the routing path from the source to the destination, for each traffic flow within the network.

6.2 Analysis of Adaptive Position Update

In this section, we analyze the performance of the proposed beaconing strategy, APU. We focus on two key performance measures: (i) update cost and (ii) local topology accuracy. The former is measured as the total number of beacon broadcast packets transmitted in the network. The latter is collectively measured by the following two metrics:

- **Missing neighbor Ratio**: This is defined as the ratio of the new neighbors a node is not aware of, but that are within the radio range of the node to the total number of neighbors.
- **False neighbor Ratio**: This is defined as the ratio of obsolete neighbors that are in the neighbor list of a node, but have already moved out of the node's radio range to the total number of neighbors.

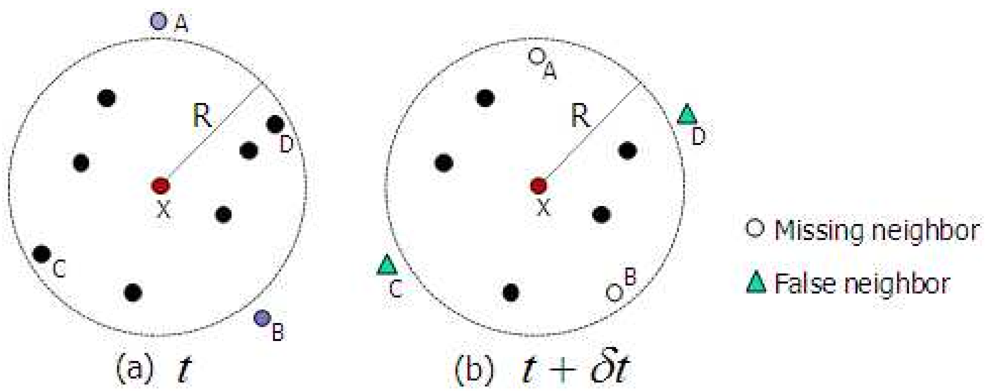


Figure 6.4: Example illustrating missing and false neighbors

The missing neighbors of a node are the new neighbors that have moved in to the radio range of this node but have not yet been discovered and are hence absent from the node's neighbor table. Consider the example in Fig. 6.4, which illustrates the local topology of a node X at two consecutive time instants. Observe that nodes A and B, are not within the radio range R of node X at time t. However, in the next time instant (i.e. after a certain period δt), both these nodes have moved into the radio range of X. If these nodes do not transmit any beacons, then node X will be unaware of their existence. Hence, nodes A and B are examples of missing neighbors.

On the other hand, false neighbors of a node are the neighbors that exist in the node's neighbor table but have actually moved out from the node's radio range (i.e., these nodes are no longer reachable). Consider the same example in Fig. 6.4. Nodes C and D are legitimate neighbors of node X at time t. However, both these nodes have moved out of the radio range of node X in the next time instant. But, node X would still list both nodes in its neighbor table. Consequently, nodes C and D are examples of false neighbors.

Note that, the existence of both missing and false neighbors adversely impacts the performance of the geographic routing protocol. Missing neighbors are ignored by a node when it makes the forwarding decision. This may lead to sub-optimal routing decisions, for example, when one of the missing neighbors is located closer to the destination than the chosen next-hop node. If a false neighbor is chosen as the next hop node, the transmitting node will repeatedly retransmit the packet without success, before realising that the chosen node is unreachable (in 802.11 MAC, the transmitter retransmits several times before signalling a failure). Eventually, an alternate node would be chosen, but the retransmission attempts waste bandwidth and increase the delay.

For mathematical tractability, we make the following simplifying assumptions:

- Nodes move according to the Random Direction Mobility (RDM) model, a popular model used in the analysis and simulations of wireless ad-hoc networks. This mobility model maintains a uniform distribution of nodes in the target region over the entire time interval under consideration [112].

Table 6.2: Notations used in the analysis

Symbol	Denotation
N	total number of nodes in network
$A \times B$	dimensions of the region of deployment
ρ	average nodes density, $\rho = AB/N$
Γ	network lifetime
R	radio range
ω	prediction periodicity, i.e. the period after which each node refreshes its neighbor list using the mobility prediction equations
M	number of flows in the network
λ	average packet arrival rate at each data source
$(0, v_{max})$	the speed of the node is randomly chosen from this range
$(0, \tau)$	the travel duration of each linear segment is randomly chosen from this range
$\overline{L}$	average distance between the source and destination nodes
$\overline{H}$	average number of hops between the source and destination nodes
χ	the total number of data packets being forwarded in the network
γ	average beacon overhead for each data packet forwarding operation
$\delta(t)$	The probability that the link between two neighboring nodes ceases to exist after a time interval t

- Each node has the same radio range R, and the radio coverage of each node is a circular area of radius R.
- The network is sufficiently dense such that the greedy routing always succeeds in finding a next hop node. In other words, we assume that a forwarding node can always find a one-hop neighbor that is closer to the destination than itself.
- The data packet arrival rate at the source nodes and the intermediate forwarding nodes is constant.

The notations used in the analysis are listed in Table 6.2.

6.2.1 Analysis of the Beacon Overhead

Recall, that the two rules employed in APU are mutually exclusive. Thus, the beacons generated due to each rule can be summed up to obtain the total beacon overhead. Let the beacons triggered by the MP rule and the ODL rule over the network lifetime be represented by O_{MP} and O_{ODL}, respectively. The total beacon overhead of APU, O_{APU}, is given by,

$$O_{APU} = O_{MP} + O_{ODL} \tag{6.3}$$

Next, we proceed to separately analyze O_{MP} and O_{ODL}.

6.2.1.1 Beacon Overhead due to the MP Rule (O_{MP})

Recall that, we have assumed that the nodes follow the RDM mobility model. According to this model, a node's trajectory consists of multiple consecutive linear segments. In each segment, the node randomly selects a direction (or heading), a speed and a travel duration from certain predefined ranges. The node moves at the selected speed in the chosen direction until the selected travel duration expires. At the end of the segment, the node pauses for a random time interval and then randomly selects another set of values for the next segment and changes its motion accordingly. For mathematical tractability, we neglect the pause time between successive segments (i.e., we assume that nodes instantly transition to the next segment).

Recall that, according to the MP rule, a node periodically predicts its own location using the motion parameters advertised in the last transmitted beacon, and compares the predicted location with its actual location. If this difference is greater than the threshold AER, a new beacon is broadcast (see Section 6.1.1). Consequently, the threshold AER directly influences the frequency and hence the number of beacon broadcasts. We seek to derive the upper bound of the beacon overhead and hence assume that the AER is zero (the lowest possible value).

In this case, a beacon will be broadcast immediately in response to any change in the node's motion characteristics (direction and speed). Since, in the RDM model, a node changes these characteristics at the end of every linear segment, the number of beacons transmitted by the node are equal to the total number of linear segments traversed by the node. Since, the travel duration of each segment is randomly selected from $(0, \tau)$, on average, a node completes traversing a linear segment after an interval of $t/2$. In other words, the average duration between two successive beacon broadcasts is $t/2$. Given that the network lifetime is Γ, on average, the number of beacons broadcast by a node is $2\Gamma/\tau$. Therefore, for a total of N nodes in the network, the total beacon overhead triggered by the MP rule, O_{MP} is given by,

$$O_{MP} = \frac{2N\Gamma}{\tau} \tag{6.4}$$

6.2.1.2 Beacon Overhead due to the ODL Rule (O_{ODL})

According to the ODL rule, whenever a node overhears a data transmission from a *new* neighbor, it broadcasts a beacon as a response (see Section 6.1.2). In other words, beacons are transmitted in response to data forwarding activities. Let χ denote the total number of data packet forwarding operations that occur over the lifetime of the network and let γ be the the average number of beacons that are triggered by each forwarding operation. Now, the total beacons triggered by the ODL rule, O_{ODL}, can be represented by,

$$O_{ODL} = \chi \cdot \gamma \tag{6.5}$$

Next, we proceed to derive χ and γ.

i. Analysis of χ: The total number of data packet forwarding operations can be represented as the product of the number of packets generated in the network and the number of times each packet is forwarded. The number of packets generated in the network can be expressed as $\lambda M \Gamma$, where λ is the packet generation rate (packets per second) at each source, M is the number of communication pairs (i.e. source-destination pairs) and Γ is the network lifetime. Let $\overline{H}$ be the average number of hops along the forwarding paths between the source and destination nodes. In other words, each packet is forwarded on average, $\overline{H}$ times, as it progresses from the source to the destination.Hence, χ can be represented as,

$$\chi = \lambda M \Gamma \cdot \overline{H} \tag{6.6}$$

Since, λ, M and Γ are known network parameters, we only need to derive $\overline{H}$.

In our previous work (see Chapter 3), we have analyzed the forwarding behavior of greedy geographic routing and derived the average number of hops along a forwarding path, given the Euclidean distance separating the source and destination node in a static multi-hop wireless network. However, in this chapter, we consider a mobile ad-hoc network, wherein, due to the mobility of the nodes, the distance between the source and destination nodes of a communicating pair is bound to change with time. This distance can be represented as a random variable. In the following, we first estimate the mean value of the source-destination distance. Then we use our previous analytical results to estimate the average hop count, $\overline{H}$.

Since, the nodes are uniformly distributed in the network (a property of the RDM model [112]), the distance between a source-destination pair is equivalent to the distance between two randomly selected points. In [113], Bettstetter et al, have analyzed the distance between two randomly select points, and formulated the average distance ($\overline{D}$) as,

$$\begin{aligned} \overline{D} = &\tfrac{1}{15}\left[\tfrac{A^3}{B^2} + \tfrac{b^3}{B^2} + \sqrt{A^2+B^2}(3 - \tfrac{A^2}{B^2} - \tfrac{B^2}{A^2})\right] \\ &+\tfrac{1}{6}\left[\tfrac{B^2}{A}\text{arccosh}(\tfrac{\sqrt{A^2+B^2}}{B}) + \tfrac{A^2}{B}\text{arccosh}(\tfrac{\sqrt{A^2+B^2}}{A})\right] \end{aligned} \tag{6.7}$$

, where $A \times B$ denotes the network dimensions. Based on our previous work in Section 3.4.3, given the Euclidean distance $\overline{D}$ between the source and destination node, the average number of hops between these nodes can be represented as follows,

$$\overline{H} = \frac{\overline{D}}{R \cdot \left[1 - \int_0^1 1 - \exp(\rho R(\arccos{(t)} - t\sqrt{1-t^2}))dt\right]} \tag{6.8}$$

, where ρ is the average node density, which is given by, $A \cdot B/N$.

Combining Equations (6.6), (6.7) and (6.8), we obtain the total number of data packet forwarding operations, χ.

ii. Analysis of γ***:*** According to the ODL rule, when a node forwards a data packet, the new neighbors that have moved in to the radio range of this forwarding node (and are hence unaware of the existence of the node forwarding the packet), broadcast beacons upon overhearing the packet transmission. This allows the forwarding node to maintain an up-to-date view of the local topology. Thus, the average number of beacons triggered by each packet forwarding operation, i.e. γ, is equal to the number of new neighbors that have entered the radio range of the forwarding node in the time interval between two successive data forwarding operations.

Recall that, one of the assumptions in our analysis is that the packet arrival rate at the source nodes and the intermediate forwarding nodes is constant, and is represented by λ. Thus, the time interval between two consecutive data forwarding operations at a node is $1/\lambda$. Since the nodes are uniformly distributed in the network, on average each node has the same number of one-hop neighbors, which is given by $\rho\pi R^2$ (where ρ is the nodes density). In steady state, the average number of new neighbors that enter the radio range of a node during the interval $1/\lambda$ is equal to the average number of neighbors that leave this region (this has been validated by simulations but have been omitted for brevity). Therefore, γ is equal to the average number of neighbors that move out of the radio range of the forwarding node during the interval $1/\lambda$.

Let $\delta(t)$ be the probability that a neighboring node moves out the radio range of a node during a small interval t. In other words, $\delta(t)$ denotes the link breakage probability. Given that a node has an average of $\rho\pi R^2$ neighbors, the number of neighbors that move out of the radio range of a node during the time $1/\lambda$ follows,

$$\gamma = \rho\pi R^2 \cdot \delta(\frac{1}{\lambda}) \tag{6.9}$$

Next, we derive $\delta(t)$. Intuitively, $\delta(t)$ is a function of the mobility pattern of the nodes. The faster the nodes move, the higher is the link breakage probability. We prove the following theorem:

Theorem 9. *The probability that the link between two neighboring nodes ceases to exists after a small time interval* t*, is given by,*

$$\delta(t) = \frac{1}{\pi a R^2}\int_0^R l \cdot \left[\int_0^{2\pi}\int_0^a g(r,\theta,l)drd\theta\right]dl \tag{6.10}$$

where $a = v_{max} \cdot t$*, and* $g(r,\theta,l)$ *is defined as,*

$$g(r,\theta,l) =$$
$$\begin{cases} 1-\alpha a + u\sin\alpha - \int_{\pi-\alpha}^{\pi}\sqrt{R^2-u^2\sin^2 v}dv & u \geq \sqrt{(R+a)^2}, \\ 1-\alpha a + u\sin\alpha - \int_{\pi-\alpha}^{\pi}\sqrt{R^2-u^2\sin^2 v}dv - 2\int_{\pi-a\sin\frac{R}{u}}^{\pi-\alpha}\sqrt{R^2-u^2\sin^2 v}dv & \\ & \sqrt{(R+a)^2} > u \geq R, \\ 1-\alpha a + u\sin\alpha - \int_0^{\pi-\alpha}\sqrt{R^2-u^2\sin^2 v}dv & R > u \geq R-a, \\ 0 & R-a > u \end{cases}$$

where $u = \sqrt{(l-r\cos\theta)^2 - r^2\sin^2\theta}, \alpha = \arccos\frac{u+a^2-R^2}{2ua}$,

Proof. A link between two neighboring nodes ceases to exist if the distance between the two nodes becomes greater than the radio range, R. Hence, the link breakage probability can be obtained by evaluating the probability that the distance between two adjacent nodes after time t becomes greater than R. Let L be

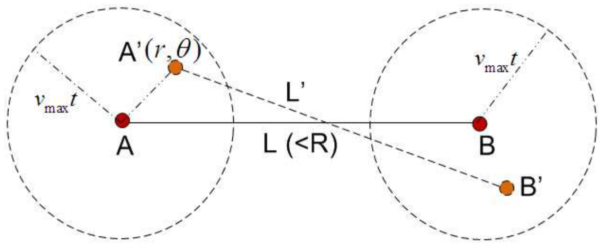


Figure 6.5: Example used to prove Theorem 9

the original distance between the two random neighboring nodes (e.g. node A and its random neighbor B) at the start of the interval, as shown in Fig. 6.5. Since node B is the one-hop neighbor of node A, node B is uniformly distributed within the radio coverage of node A. Therefore, the distance L is a random variable and we calculate its distribution as follows.

The probability that the distance L is less than a value l, is the probability that the node B is located within the circular region of radius l. Therefore, the cumulative density function (cdf) of L is given by,

$$\begin{aligned} F_L(l) &= Prob(L \leq l) = \frac{\pi l^2}{\pi R^2} \\ &= \frac{l^2}{R^2} \end{aligned} \tag{6.11}$$

The probability density function (pdf) of L follows,

$$f_L(l) = \frac{d}{dl} F_L(l) = \frac{2l}{R^2}, \qquad 0 < l \leq R \tag{6.12}$$

Let L' be the new distance between the two nodes after the small interval t. Thus $P(L' > R|L = l)$ is the link breakage probability given the original distance between the two nodes is l. By law of total probability, the link breakage probability over all possible values of l is,

$$\delta(t) = P(L' > R) = \int_0^R f_L(l) P(L' > R|L = l) dl \tag{6.13}$$

We now compute $P(L' > R|L = l)$. Without loss of generality, we assume that node A is located at the origin $(0, 0)$ at the start of interval. Recall that, in the RDM mobility model, a node is assumed to be moving along a randomly selected direction (represented as an angle) from $(0, 2\pi)$ and a randomly selected speed from $(0, v_{max})$. The maximum distance that a node can traverse during interval t is $v_{max} \cdot t$. We assume that nodes do not change their moving velocity during the small interval of t. Therefore, the possible new location of node A after interval t forms a circular area with a radius of $v_{max} \cdot t$, as shown in Fig. 6.5.

Let $a = v_{max} \cdot t$. Using the polar coordinates system, the *pdf* of the distance r to the new location is $1/a$, and the *pdf* of angle θ is $1/(2\pi)$. Therefore the joint pdf of the new location (r, θ) of node A is given by,

$$f(r, \theta) = \frac{1}{2\pi a} \tag{6.14}$$

Given that the original distance between node A and B is l, and the new location of node A is at (r, θ), we denote $g(r, \theta, l)$ as the link breakage probability over all possible new locations of B. The overall link breakage probability given the original distance of l can be expressed as

$$P(L' > R | L = l) = \int_0^{2\pi} \int_0^a f(r, \theta) g(r, \theta, l) dr d\theta \tag{6.15}$$

By some tedious calculations, the function $g(r, \theta, l)$ can be expressed as equation (6.11). The details are omitted here for brevity.

Finally, combining Equations (6.13), (6.12), (6.15) and (6.14), Theorem 9 is proved. □

Given the link breakage probability $\delta(t)$, we can use Equation (6.9) to estimate γ, i.e. the average number of beacons that are triggered by each data packet forwarding operation. Since, we have derived the total number of data packet forwarded ξ earlier, we can calculate the beacon overhead triggered by ODL rule using Equation 6.5.

Finally, according to Equation (6.3), the total beacon overhead generated by APU (O_{APU}) follows,

$$O_{APU} = O_{MP} + O_{ODL} = \frac{2N \cdot \Gamma}{\tau} + \chi \cdot \gamma \tag{6.16}$$

6.2.2 Analysis of the Local Topology Accuracy

Recall that, we have defined two metrics that collectively represent the neighbor table accuracy - (i) missing neighbor ratio and (ii) false neighbor ratio. The neighbor table maintained by a node is only referenced when the node has to forward a packet. Consequently, it only makes sense to calculate the neighbor table accuracy at the time instants when the node is forwarding a data packet.

We first analyze the missing neighbor ratio. In our earlier analysis (see analysis of γ), we have shown that, according to the ODL rule,the average number of new neigbors that enter the radio range of a node between two successive forwarding operations (i.e. the interval $1/\lambda$) is given by γ. The node will only become aware of these new neighbors when it forwards the next packet, since these neighbors will broadcast beacons announcing their presence in response to the packet transmission. According to Equation (6.9), on average $\rho\pi R^2 \cdot \delta(1/\lambda)$ new neighbors enter the radio range of a forwarding node during the interval $1/\lambda$. The number of actual neighbors is the total number of nodes within the radio range of the forwarding node, which is $\rho\pi R^2$ on average. Therefore, the missing neighbor ratio, represented by Λ^m_{APU}, can be computed as follows,

$$\Lambda^m_{APU} = \frac{\rho\pi R^2 \cdot \delta(\frac{1}{\lambda})}{\rho\pi R^2} = \delta(\frac{1}{\lambda}) \tag{6.17}$$

We now proceed to evaluate the false neighbor ratio. As per the MP rule, a node periodically estimates the current locations of its neighbors using Equation (6.1). Let ω denote the periodicity of this operation. At the beginning of each period, the node updates its neighbor list by removing all the false neighbors (i.e. those nodes that are estimated to have moved out of its radio range). Since, data packets arrive at the forwarding node at random during the interval ω, the average time of arrival of a packet is given by $\omega/2$. The number of false neighbors at time $\omega/2$ is the number of neighbors that have moved out of the radio range during $\omega/2$. Therefore, according to Equation 6.9, the false neighbor ratio, denoted by Λ^f_{APU}, is given by,

$$\Lambda^f_{APU} = \frac{\rho\pi R^2 \cdot \delta(\frac{\omega}{2})}{\rho\pi R^2} = \delta(\frac{\omega}{2}) \tag{6.18}$$

6.3 Simulation Results

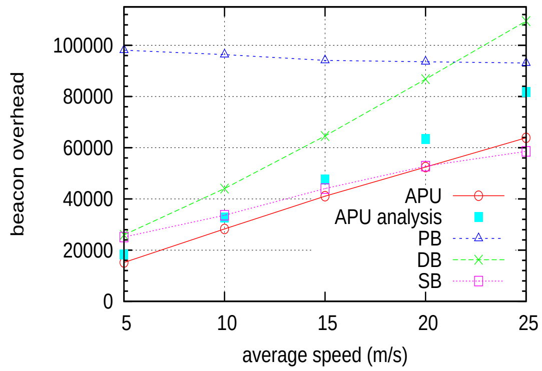
In this section, we present a comprehensive simulation-based evaluation of APU using the popular NS-2 simulator. We conduct two set of experiments. In the first set of simulations, we demonstrate that APU can effectively adapt the beacon transmissions to the node mobility dynamics and traffic load. In addition, we also compare the performance of APU with other beaconing schemes. These include PB and two other recently proposed adaptive beaconing schemes: (i) Distance-based Beaconing (DB) and (ii) Speed-based Beaconing (SB) (see Section 2.1.2). In addition, we also evaluate the validity of the analytical results derived in Section 6.2, by comparing the same with the results from the simulations. In the second set of experiments, we evaluate the performance of APU in a practical vehicular ad-hoc network (VANET) scenario that exhibits realistic movement patterns of public transport buses in a metropolitan city. This enables us to investigate if the benefits exhibited by APU still hold in a real-world scenario.

We use two sets of metrics for the evaluations. The first set includes the metrics used in our analysis, viz., beacon overhead and local topology accuracy (false and missing neighbor ratio), which directly reflect the performance achieved by the beaconing scheme. Note that, the beaconing strategies are an integral part of geographic routing protocols. The second set of metrics seek to evaluate the impact of the beaconing strategy on the routing performance. These include:

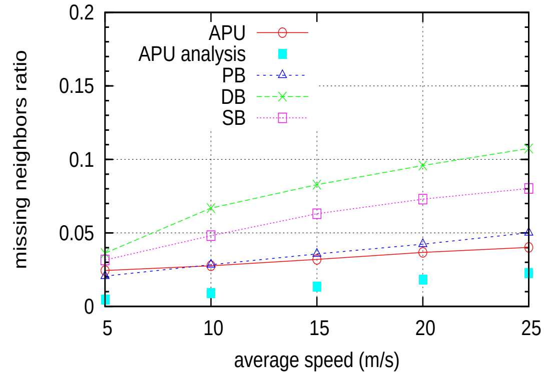
(i) packet delivery ratio, which is measured as the ratio of the packets delivered to the destinations to those generated by all senders (ii) MAC layer collisions, which reflect the interference caused due to the beacon transmissions and (iii) average end-to-end delay incurred by the data packets. In the simulations, we have implemented GPSR [29] as an illustrative example of a geographic routing protocol. We simulate IEEE 802.11b as the MAC protocol and assume a two-ray ground propagation model.

6.3.1 Impact of Node Mobility on Beaconing Schemes

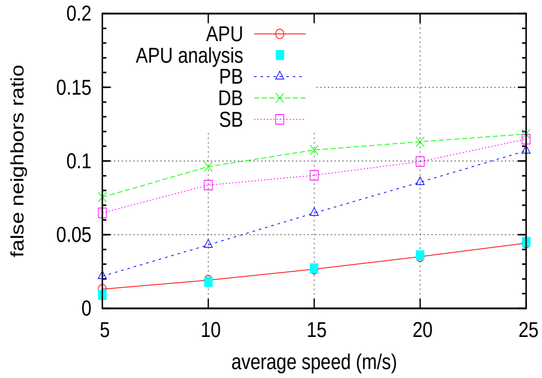
We first evaluate the impact of varying the mobility dynamics of the nodes on the performance on APU. In addition, we compare the performance of APU with other beaconing schemes. The simulations are conducted in NS-2 with each experiment being run for 1000 seconds. The results represented here are averaged over 12 runs (The standard deviation achieved is on average less than 5% of the mean value). The confidence intervals are narrow and the size of intervals are only within $\pm 7\%$ of the corresponding average value in most of cases. In each simulation, 150 nodes are randomly placed in a region of size 1500m*1500m. The radio range for each node is assumed to be 250 meters (thus the average number of one-hop neighbors for each node is 12). We use Constant Bit Rate (CBR) traffic sources with each source generating four packets per second. We simulate 15 traffic flows and randomly select nodes as source-destination pairs as the traffic flows. We have assumed that the nodes move according to the RDM model, to be consistent with our analytical results. First, we study the impact of changing the mobility dynamics of the nodes on the performance of APU and PB. Note that, the faster the node moves, the more frequently it changes its mobility parameters (i.e. speed and direction). We vary the average speed of the node from 5m/s (18km/hr, representing low dynamism) to 25m/s (90km/hr, representing high



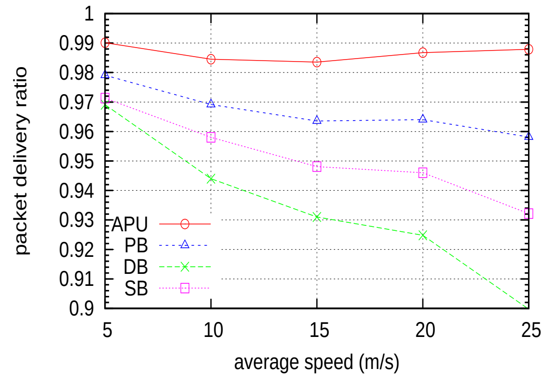
(a) Beacon Overhead (packets)



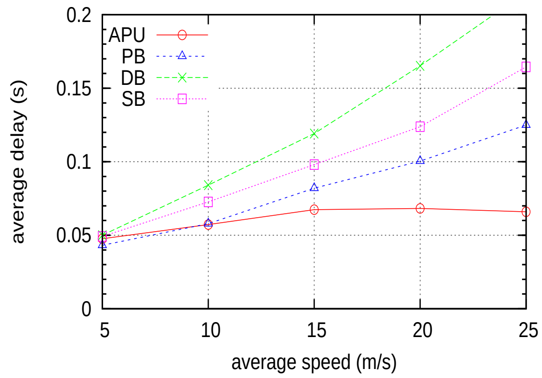
(b) Missing Neighbors Ratio



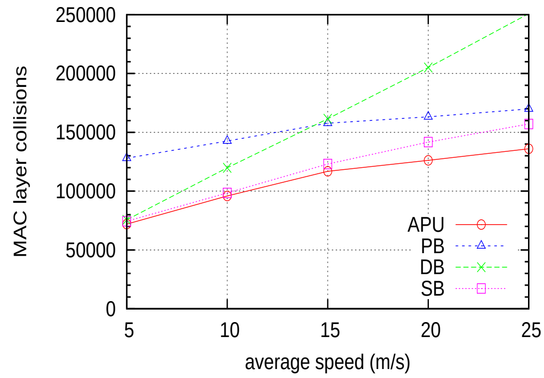
(c) False Neighbors Ratio



(d) Packet Delivery Ratio



(e) Average End-to-End Delay



(f) MAC Layer Collisions

Figure 6.6: Impact of node speed on the performance of beaconing schemes

dynamism). This range is consistent with typical vehicular mobility scenarios.

We assume that the prediction period in APU (ω) is 1s. The beacon period (ϵ) in PB is also assumed to be 1s, which is the default value in NS2 and also is recommended in [68]. The neighbor timeout interval in PB is set to 3s. In DB

[68], assuming that the distance parameter is d, and a node is moving at speed v, the beacon interval is given by, d/v. We have set the distance parameter, $d = 20m$ and the neighbor time-out interval as twice the beacon interval, as suggested in [68]. In SB, if the speed of the node is v, then its beacon interval is given by $B = a + (b - a) \cdot (\frac{v_{max} - v}{v_{max} - v_{min}})^n$, where $[a, b]$ is pre-defined beacon interval range; v_{min} and v_{max} are the minimal and maximal node speeds. We assume that the beacon interval range is [1s, 5s] and $n = 4$, as suggested in [68]. Since, the average speed is varied from 5m/s to 25m/s in the simulations, $v_{min} = 0$ and $v_{max} = 50$.

We initially focus on the first set of metrics, i.e., the beacon overhead and the missing and false neighbor ratios. Fig. 6.6(a) shows that the beacon overhead of APU increases linearly as a function of the average speed. This behavior is primarily attributed to the ODL rule. Recall that, in the OLD rule, when a node forwards a data packet, all of its new neighbors that overhear the data packet respond with beacons. When the network topology is highly dynamic, the local topology of a node frequently changes with several new neighbors entering the radio range. As a result, APU generates more beacons in order to keep up with the frequent changes of topologies. With DB, we observe a similar linear increase. This is expected, because, the beacon periodicity in DB is inversely proportional to the node speed. Finally, with SB, the beacon overhead also increases with increase in average speed, though not linearly. The beacon overhead tends to saturate as the average speed increases. This is because of the polynomial relationship that exists between the beacon update period and the node speed. In contrast, observe that PB results in very high beacon overhead, which does not vary significantly with the node speed. This is because in PB, the beacon broadcasts are independent of the node mobility.

Fig. 6.6(b) shows that APU can achieve a similar missing neighbor ratio as that of PB, despite the fact that APU generates significantly less beacon overhead.

Recall that, the beacon broadcasts in APU are more concentrating around the routing paths due to the ODL rule. Therefore, these beacons are highly effective in maintaining an up-to-date view of the the local topology at the nodes involved in forwarding most of the traffic. On the contrary, both DB and SB exhibit higher missing neighbor ratio as compared to APU. In particular, when the average node speed is 25m/s, the missing neighbor ratio for DB and SB is more than twice as that of APU. We attribute this increase in the missing neighbors to the fact that in both DB and SB, when a fast moving node passes a slow node, the fast node may not detect the slow node due to the infrequent beacon transmissions by the slow node. Note that, in APU, due to the ODL rule, if either of these nodes are involved in forwarding packets, beacons would be exchanged, thus reducing the likelihood of missing neighbors.

Fig. 6.6(c) illustrates that APU can achieve a very low false neighbor ratio as compared with the other three schemes. This can be explained as follows. Since each node in APU uses mobility prediction to track the locations of its neighbors (MP rule), the node can always quickly remove the obsolete neighbors, which have moved out of its radio range, from the neighbor list. On the contrary, a node in PB, DB or SB only passively removes an obsolete neighbor when the node has not heard any beacons from the neighbor during a certain time window . Therefore, the removal of obsolete neighbors is delayed resulting in a higher false neighbor ratio. In summary, APU succeeds in maintaining an accurate view of the local topology in the network, while keeping the beacon overheads to a minimum.

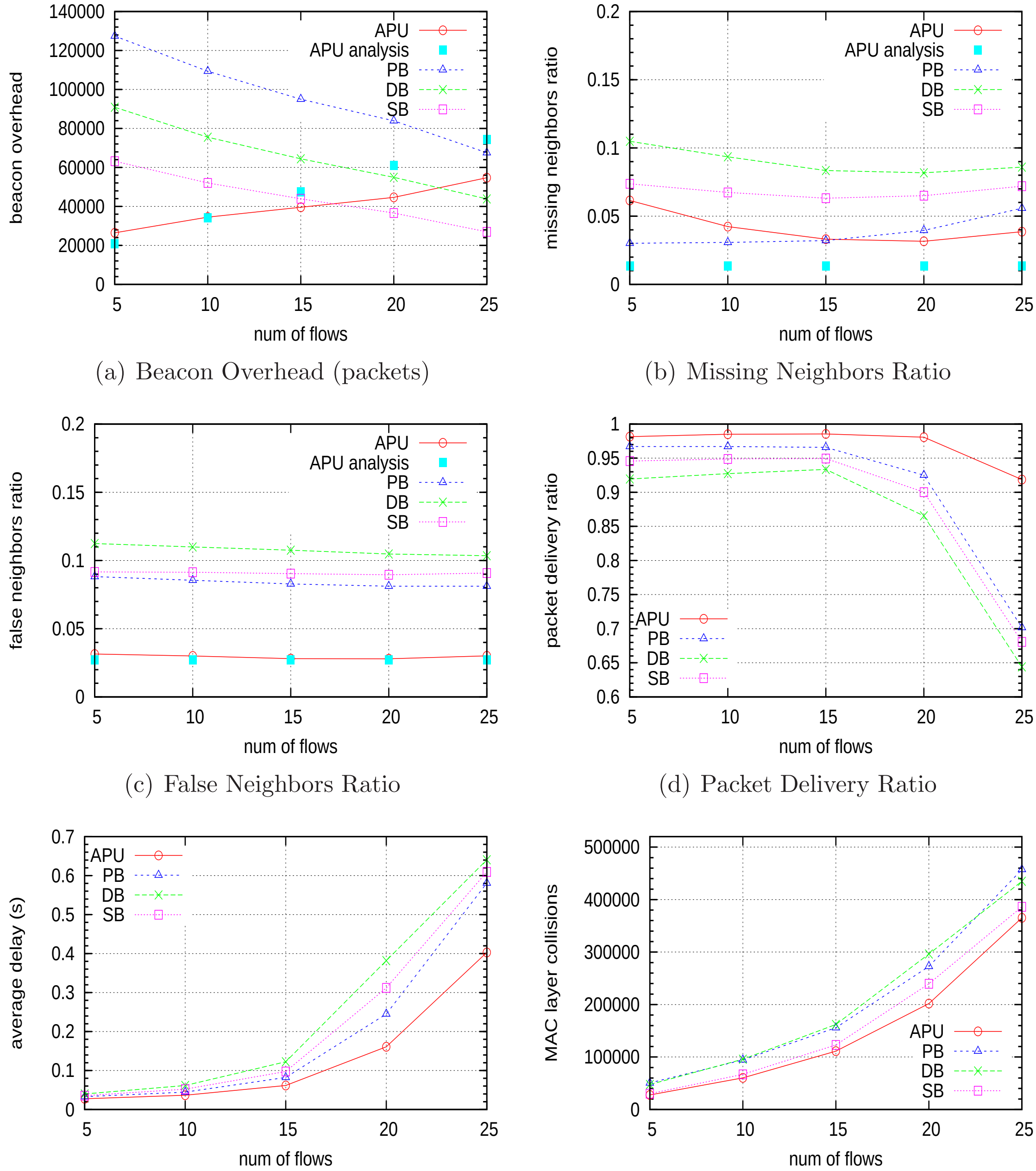
We also seek to validate the results from our analysis in Section 6.2. We obtain the analytical results for the beacon overhead, false neighbor ratio and missing neighbor ratio for APU by substituting the simulation parameters in the corresponding equations. These results are compared with the corresponding simulation results in Figs. 6.6(a)-(c). One can readily observe that the analytical model can provide an upper bound for the beacon overhead and an accurate approximation for the false neighbor ratio. The only exception is the missing neighbor, in which case, the analytical results are consistently lower than the simulation results. This is because, in our analysis, we assumed that when a node forwards a data packet, all its new neighbors (i.e. missing neighbors) respond with beacons to report of their existence. However, in simulations we observed that in some instances, not all missing neighbors reply with beacons. This is because, some of these nodes already have the forwarding node in their neighbor list (due to their previous proximity).

Next, we focus on the second set of metrics, which evaluate the impact of the beaconing strategies on the performance of the geographic routing protocols (GPRS in this case). These metrics include the packet delivery ratio, end-to-end packet delay and MAC collisions. Since, APU is successful in maintaining an up-to-date view of the local topology in the network, it also achieves a consistently high packet delivery ratio as illustrated in Fig. 6.6(d), independent of the speed, since each node involved in forwarding a packet is almost always able to find an appropriate next hop neighbor. Consequently, most packets are forwarded along the optimal paths, which in turn results in low end-to-end delay, as can be seen from Fig. 6.6(e). In comparison, all the other three schemes (i.e. PB, DB and SB) exhibit a decrease in their packet delivery ratio as the average speed of the nodes increases (Fig. 6.6(d)). Further, as seen from Fig. 6.6(e), the average end-to-end delay also increases linearly as a function of speed for these three schemes. This can be attributed to the fact that the false and missing neighbor ratios are considerably higher in all these schemes as compared to APU. The more missing neighbors a node has, the greater is the likelihood of sub-optimal routing decisions and even routing failures. In some instances, the missing neighbors could be better choices for the next hop node. In other instances, all the neighbors of a node may be missing, which leads to routing failures. This in turn increases packet loss in the network, thus reducing the packet delivery ratio. In addition, the end-to-end delay in the network also increases, since, there is a greater chance that packets follow longer routes to reach the destination. Similarly, the existence of false neighbors can lead to a situation where a node selects one of these false neighbors as the next hop node. In such situations, the MAC layer unsuccessfully retransmits several times and then reports the failure to the routing layer, which in turn results in the node selecting a new next hop node. Consequently, this increases the delay experienced by the packets.

MAC layer collisions are directly proportional to the beacon overheads. The greater the number of beacons transmitted, the higher is the chance of the beacons colliding with data packets. Further, as explained earlier, the existence of missing and false neighbors lead to frequent retransmissions of data packets, which in turn also increase the probability of MAC collisions. Consequently, as can be observed from Fig. 6.6(f), APU results in a minimum number of collisions compared to other three schemes.

6.3.2 Impact of Traffic Load on Beaconing Schemes

In the second set of simulations, we evaluate the impact of varying the traffic load on the performance of APU and also compare APU with the three beaconing schemes under consideration. We use the same scenario as in the first set of experiments. We fix the average node speed to 15m/s. We vary the number of flows from 5 (low load) to 25 (high load). We initially focus on the first set of metrics, i.e., the beacon overhead and the missing and false neighbor ratios. As the number of traffic flows increase, more nodes in the network are involved in forwarding packets. Since, the ODL rule in APU aims at maintaining an accurate view of the local topology for nodes involved in forwarding packets, we expect the beacon



(a) Beacon Overhead (packets)
(b) Missing Neighbors Ratio
(c) False Neighbors Ratio
(d) Packet Delivery Ratio
(e) Average End-to-End Delay
(f) MAC Layer Collisions

Figure 6.7: Impact of traffic load on the performance of beaconing schemes

overhead to increase with the traffic load. Fig. 6.7(a) confirms our hypothesis. On the contrary, the beacon overhead of the other three schemes (PB, DB and SB) decrease with an increase in the traffic load. This is because, in these schemes the beacon information is piggybacked with data packets whenever possible. When the traffic load is high, the opportunities for piggybacking increase, thus reducing the explicit transmission of beacons. However, the beacon overhead of APU is still lower than that of PB. For low traffic load, the beacon overhead of APU is also lower than that of DB and SB. However, when the traffic load is high, DB and SB outperform APU.

The results for the local topology accuracy (i.e. missing and false neighbor ratios) in Figs. 6.7(b) and 6.7(c)) exhibit similar characteristics as those observed in the previous set of simulations (Figs. 6.6(b) and 6.6(c)). APU has a comparable missing neighbor ratio as that of PB, while significantly outperforming DB and SB. Further, APU reduces the false neighbor ratio by more than 60% in comparison with the other three schemes. One can observe from Figs. 6.7(b) and 6.7(c), that the local topology accuracy remains relatively unaffected by changes in the traffic load. This is expected, since these metrics are significantly influenced by the topology dynamics rather that the traffic load.

As in the previous scenario, we also plot the analytical results for APU derived in Section 6.2 as a function of the traffic load in Figs. 6.7(a)-(c). We observe a similar difference in the results for the missing neighbor ratio (reasons explained earlier). In addition, Fig. 6.7(a) shows that the simulation results for the beacon overhead are inconsistent with the analytical results, when the traffic load is high. The reason for this is that, in the analysis, we have assumed that the packet arrival rate at all intermediate nodes is constant (λ). However, this assumption may not hold if multiple flows share some common forwarding nodes, which is highly likely when the traffic load is high. For example, if an intermediate node forwards data packet from multiple flows, the packet arrival rate at this node would be greater than λ. In other words, the packet inter-arrival duration at such nodes would be less than $1/\lambda$. Consequently, in this shorter interval, fewer new neighbors would enter the radio range of these nodes. As a result, the number of beacons transmitted according to the OLD rule would be lower as compared to when the routing paths for multiple flows are completely disjoint (as assumed in the analysis). Hence, our analytical results overestimate the beacon overheads for APU, particularly when the load is high. However, the analytical results can still serve as an upper bound.

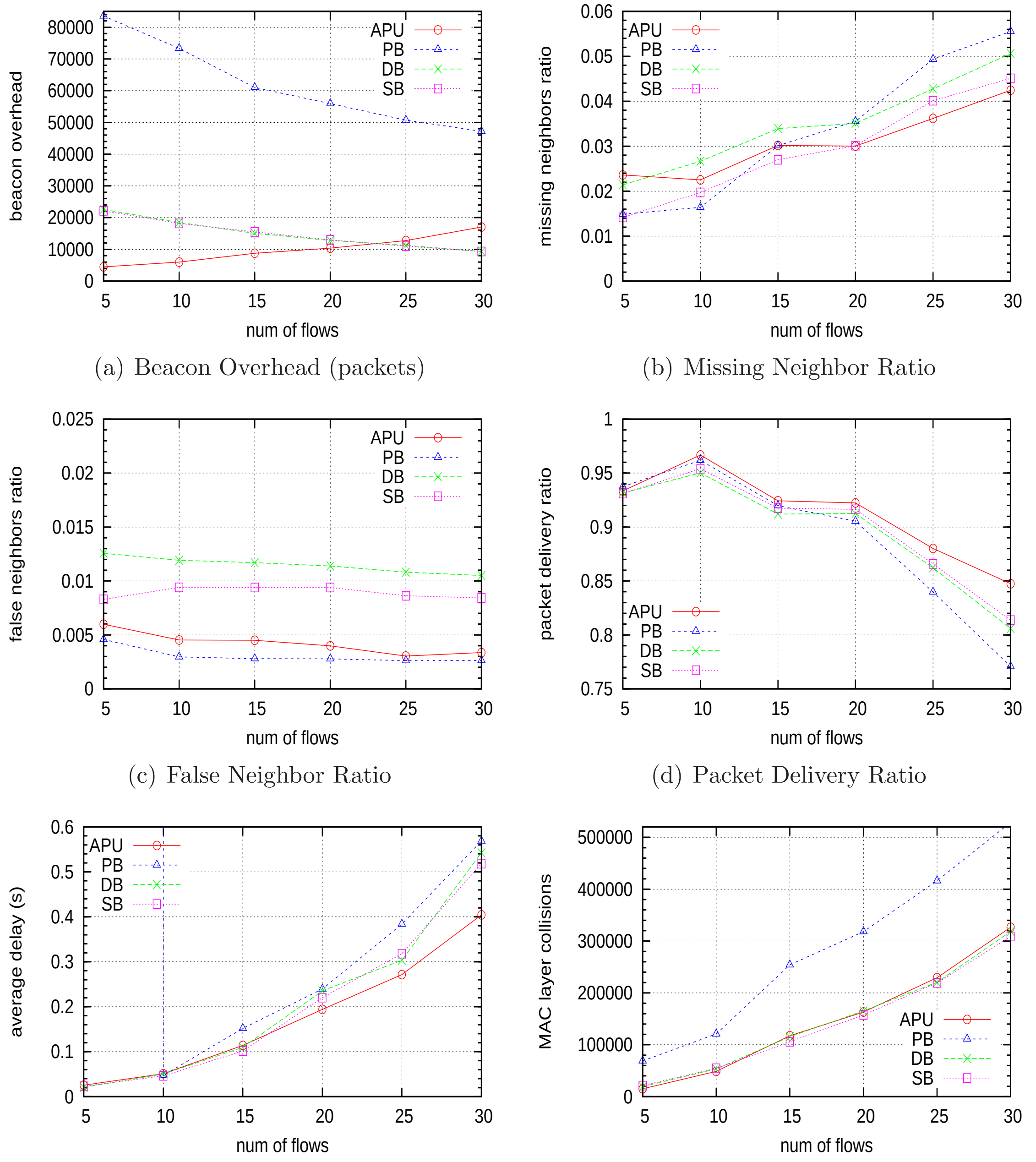
Next, we focus on to the second set of metrics, which evaluate the performance of GPSR. As seen from Fig. 6.7(e), when the traffic load is low to moderate (5 to 15 flows), the packet delivery ratio for all schemes is fairly constant. However, APU does outperform all the other schemes marginally. Observe that when the traffic load is very high, the packet delivery ratio reduces 30% when there are 25 flows). When the traffic load is very high, the higher number of packet retransmissions in these schemes due to their relatively poor local topology accuracy (Figs. 6.7(b)-(c)) results in a significant increase in routing failures, thus leading to a significant drop in the packet delivery ratio. Further, in the case of APU, most packets can be forwarded along optimal paths, which results in improved end-to-end delay (Fig. 6.7(e)). As expected, all schemes exhibit an almost linear increase in the number of MAC collisions as a function of the traffic load, as seen in Fig. 6.7(f). However, APU for the most part, marginally outperforms all the other schemes.

Overall, the simulation results show that APU is significantly better at adapting to network mobility and traffic load as compared to PB, DB and SB. We find that when the mobility characteristics of a node do not change significantly (i.e. when the average speed is 5 m/s) and under light traffic load (5 traffic flows), APU reduces the beacon overhead by 80% compared to that of PB, but achieves a similar high performance in terms of packet delivery ratio and average end-to-end delay. When the node mobility is highly dynamic (i.e. when the average speed is 25m/s) and under heavy traffic load (30 traffic flows), APU has similar beacon overhead as that of PB, but achieves considerably better pakcet delivery ratio and average end-to-end delay. The simulation results also show that the distance-based and speed-based beaconing strategies can reduce the beacon overhead under low dynamism. However, when the nodes frequently change their motion parameters, the performance of these schemes degrades substantially as compared with APU. The fundamental reason for the improvement of APU is that the beacons generated in APU are more concentrated in the network hotspots, where they are most useful in maintaining an accurate representation of the local neighborhood.

6.3.3 Results for a Realistic VANET Scenario

In the previous set of simulations, we have assumed that the nodes move according to the RDM mobility model, in order to be consistent with our analytical results. However, in a real-world scenario, the mobility dynamics of the nodes can be significantly different. We conduct a second set of simulations using a real-world Vehicular Ad hoc Network (VANET) to confirm if the findings from our previous experiments with synthetic mobility models hold true in a realistic scenario. We use realistic movement patterns of public transport buses in a metropolitan city to simulate the VANET.

We have used mobility traces that capture the actual movement of public transport buses from the King County Metro bus system in Seattle, USA [114]. This transport network consists of close to 1163 buses plying over 236 distinct routes covering an area of 5100 square kilometers. The traces were collected over a three week period in November 2001. The traces are based on location update messages sent by each bus. Each bus logs its current location using an Automated Vehicle Location (AVL) system [110], its bus id and route id along with a timestamp. The typical update frequency is 30 seconds. We have not



(a) Beacon Overhead (packets)

(b) Missing Neighbor Ratio

(c) False Neighbor Ratio

(d) Packet Delivery Ratio

(e) Average End-to-End Delay

(f) MAC Layer Collisions

Figure 6.8: Impact of traffic load on the performance of beaconing schemes (realistic VANET scenario)

simulated the entire bus network. This is because the network is quite sparse (for example, only 1 or 2 buses) in several regions of the city, which would not lead to meaningful results. Rather, we focus on the downtown area, which has a consistently high density of buses. We focus on a rectangular region of size 4km x

6km in the downtown. We create three scenarios from a weekday trace (Thursday, Nov 8, 2001), each lasting for 1000 seconds, but at different times of the day; 8am (morning peak), 12pm (afternoon off-peak) and 5pm (evening peak).

The simulations were conducted in NS-2 with the node movement patterns being read from a file. We assumed a radio range of 1km, which is consistent with that for the DSRC (Dedicated Short Range Communications) [98] standard proposed for vehicular communication. We used CBR traffic sources with the sender transmitting at 4 packets per second. We study the impact of varying the traffic load from 5 to 30 flows on the performance of the beaconing schemes. The source and destination nodes were randomly selected. The results presented here are averaged over 9 runs, with each scenario being executed thrice with different random seeds. Note that, since we use real vehicular traces to simulate the node mobility, we are unable to systematically study the impact of mobility dynamics on the performance. However, the traces capture the typical dynamism that would exist in a typical urban VANET scenario.

Fig. 6.8(a) illustrates that in most situations, APU achieves significantly lower beacon overhead as compared to the other three beaconing schemes. For example, with 15 traffic flows, APU reduces the beacon overhead by 50% as compared with distance-based beaconing. However, with an increase in the traffic load, we notice a slight increase in the beacons exchanged in APU. This is primarily due to the ODL rule, which tries to maintain an accurate topology along the forwarding paths. On the contrary, with PB, DB and SB, since the beacons are piggybacked on the data packets, the number of explicit beacon packets that need to be broadcast decreases with an increase in the load. Figures 6.8(b) and 6.8(c) show that APU can maintain a slightly better local topology accuracy than the other three schemes at higher traffic load. Consequently, APU achieves a better packet delivery ratio and lower end-to-end delay than other schemes at higher trafic load cases, as shown in Fig. 6.8(d) and Fig. 6.8(e). Fig. 6.8(f) illustrates that the MAC collisions in APU are comparable to that of SB and DB, but significantly lower than that of PB. These simulations confirm that even in a real-world scenario, APU significantly outperforms all other beaconing schemes.

6.4 Summary

In this chapter, we have identified the need to adapt the beacon update policy employed in geographic routing protocols to the node mobility dynamics and the traffic load. We proposed the **Adaptive Position Update (APU)** strategy to address these problems. The APU scheme employs two mutually exclusive rules. The MP rule uses mobility prediction to estimate the accuracy of the location estimate and adapts the beacon update interval accordingly, instead of using periodic beaconing. The ODL rule allows nodes along the data forwarding path to maintain an accurate view of the local topology by exchanging beacons in response to data packets that are overheard from new neighbors. We mathematically analyzed the beacon overhead and local topology accuracy of APU and showed that APU significantly outperforms the traditional periodic beaconing strategy. We have embedded APU within GPSR and have compared it with other related beaconing strategies using extensive ns-2 simulations for varying node speeds and traffic load. Our results indicate that the APU strategy significantly lowers the number of beacon updates while also achieving a better packet delivery rate. Further, with APU the packets are more likely to be routed along the shortest-hop path to the destinations, hence improving the end-to-end delay. We have also presented some initial results using realistic movement patterns of public transport buses within a city, which validates that the performance improvements of APU can be replicated in a real-world VANET scenario.

Chapter 7

Conclusion

Hop count in multi-hop wireless ad-hoc networks is an important performance metric that has a deterministic effect on throughput, energy consumption, routing overhead and end-to-end delay. Previous work on the hop count analysis do not give a very comprehensive insight, nor do they exploit the analysis results of hop count adequately enough to address the challenging issues in a wireless ad-hoc network.

In this thesis, we have analyzed the relationship of the source-to-destination distance to hop count under realistic scenarios. At the routing layer, we have focused on popular greedy geographic routing, which can provide a scalable routing solution for large-scale networks and also be able to be quickly adapted to topology dynamics. At the physical layer, we have studied the ideal radio model and a more realistic radio model, i.e. log-normal shadowing radio model. We have accurately analyzed the hop count distribution and the mean value under both radio models in a network where nodes are uniformly distributed. We also have proposed approximations to reduce the computation complexity. The analytical model is validated by an extensive set of simulations, including a real-world vehicular network that exhibits realistic topology. Both the analysis and simulation results show that greedy routing generally takes less hop count in shadowing radio model than in an ideal radio model for a same source-to-destination distance. Our hop count analysis reveals that the widely used approach of taking the ratio of source-to-destination Euclidean distance to radio range may significantly overestimate the actual hop count if random signal fading is present. Our analysis also challenges the current belief that greedy geographic routing can (approximately) find the shortest path between a source and destination in a multi-hop wireless network. We have also shown that this is true only if no random fading is present. Under the presence of random fading, greedy routing can take paths significantly longer than the shortest path.

Furthermore, we have proposed three applications to exploit our analytical results of hop count to address some challenging issues for a multi-hop wireless adhoc network. In the first application, we analyzed the per-node traffic load for a random wireless sensor network where each node periodically sends a data packet to the sink. We have consider both an idealistic and realistic radio model. Our results confirm that irrespective of the radio models, the traffic load generally increases as a function of the node's proximity to the sink. In the immediate vicinity of the sink, however, the ideal radio model shows the existence of a volcano region near the sink, where the traffic load drops significantly. Whereas, on the other hand, with the log-normal shadowing model, the traffic load actually increases at a much higher rate as one approaches the sink, resulting in the formation of a mountain peak.

In the second application, we proposed a hop count based local geocasting. Our scheme uses hop count as TTL value to restrict flooding within a local area around the source. We estimated the hop count radius given a flooding distance radius based on the aforementioned distance to hop count relationship. Our scheme thus eliminates the location requirement that is imposed on the previous location-based geocasting scheme. These simulation results show that our scheme can achieve similar performance as that of a location-based scheme which requires a precise location information.

In the last application, we focused on the position update mechanism that is used to maintain neighbors' location in geographic routing. We proposed an Adaptive Position Update (APU) algorithm that enables a position update which is adaptive to nodes mobility and traffic pattern. Based on the hop count analysis results,we analyzed the performance of APU, including the beacon overhead and the accuracy of neighbors' location. The simulation results validated the analytical model and demonstrated that APU performs significantly better than other schemes under various cases of topology dynamics and traffic load.

The hop count value, given a pair of communicating nodes, depends on the network topology and the routing protocol employed. In this thesis, we have considered the popularly used greedy geographic routing. In future work, we can apply the similar techniques presented in this thesis to analyze the hop count metric for other routing protocols, e.g. energy efficient greedy routing [52, 115]. Our work reveals that the greedy routing in the shadowing radio model can take a longer path than the shortest path. In future, we are interested in finding out how to modify greedy routing in order to enable it to be comparable to the shortest path routing. The per-node traffic load identified in this work can be used to estimate the per-node energy consumption and therefore the operational lifetime of a wireless sensor network. Furthermore, the per-node energy consumption can be useful in identifying **hotspots** in the network, i.e., the regions where sensor nodes are expected to drain their energy at a faster rate (due to a higher traffic load). Additional sensor nodes can be deployed at these hotspots to extend the operational lifetime of the network. The per-node traffic load analysis results can also be incorporated into MAC protocol design. Last but not least, the hop count knowledge can be used as a fundamental component to analyze other performance metrics such as packet delivery ratio and throughput.

Bibliography

[1] I. Chlamtac, M. Conti, and J. J. N. Liu. Mobile ad hoc networking: imperatives and challenges. Ad Hoc Networks, 1(1):13–64, 2003.

[2] I. F. Akyildiz and X. Wang. A survey on wireless mesh networks. IEEE Communications Magazine, 43(9):23–30, September 2005.

[3] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. Computer Network ISDN System, 47(4):445–487, 2005.

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. Computer Network, 38(4):393–422, 2002.

[5] P. Baronti, P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta, and Y. F. Hu. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards. Computer Communications, 30(7):1655–1695, 2007.

[6] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, pages 88–97, Atlanta, Georgia, USA, 2002. ACM.

[7] J. Li, C. Blake, D.S.J. De Couto, H.I. Lee, and R. Morris. Capacity of ad hoc wireless networks. In Proceedings of ACM MOBICOM, pages 61–69, Rome, Italy, July 2001.

[8] P. Gupta and P.R. Kumar. The capacity of wireless networks. IEEE Transactions on Information Theory, 46(2):388–404, March 2000.

[9] G. K. Holland and N. K. Vaidya. Analysis of TCP performance over mobile ad hoc networks. Wireless Networks, 8(2-3):275–288, March 2002.

[10] J. H. Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. In Proceedings of IEEE INFOCOM, volume 1, pages 22–31, Tel Aviv, Israel, 2000.

[11] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In Proceedings of ACM MOBICOM, pages 114–128, 2004.

[12] M. Grossglauser and D. N. C Tse. Mobility increases the capacity of ad hoc wireless networks. IEEE/ACM Transactions on Networking, 10(4):477–486, August 2002.

[13] C.-K. Toh, M. Delwar, and D. Allen. Evaluating the communication performance of an ad hoc wireless network. IEEE Transactions on Wireless Communications, 1(3):402–414, July 2002.

[14] F. Bai, N. Sadagopan, B. Krishnamachari, and A. Helmy. Modeling path duration distributions in manets and their impact on reactive routing protocols. IEEE Journal on Selected Areas in Communications, 22(7):1357–1373, September 2004.

[15] P. P. Pham and S. Perreau. Performance analysis of reactive shortest path and multipath routing mechanism with load balance. In In Proceeding of IEEE INFOCOM, volume 1, pages 251–259, March 2003.

[16] D. B. Johnson, Y. Hu, and D. Maltz. The dynamic source routing (DSR) protocol for mobile ad hoc networks for ipv4. February 2007.

[17] A. Gani, Q. Han, N. B. Anuar, and O. Zakaria. Enhancing DSR protocol performance in mobile ad hoc network using ack reply. WSEAS TRANSACTIONS on COMMUNICATIONS WTOC, 8(2):227–236, 2009.

[18] Y. Kim, I. L. Y. Moon, and S. J. Cho. A comparison of improved AODV routing protocol based on IEEE 802.11 and IEEE 802.15.4. Journal of Engineering Science and Technology, 4(2):132–141, 2009.

[19] L. Kleinrock and J. A. Silvester. Optimum transmission radii for packet radio networks or why six is a magic number. In Proceedings of the IEEE National Telecommunications Conference, Birmingham, Alabama, USA, December 1978.

[20] D. Lebedev and J. M. Steyaert. Path lengths in ad hoc networks. In 2004 international workshop on wireless ad hoc network, pages 207–211, May 2004.

[21] S. De, A. Caruso, T. Chaira, and S. Chessa. Bounds on hop distance in greedy routing approach in wireless ad hoc networks. International Journal on Wireless and Mobile Computing, 1(2):131–140, 2006.

[22] L. Zhao and Q. Liang. Hop-distance estimation in wireless sensor networks with applications to resources allocation. EURASIP Journal on Wireless Communications and Networking, 2007:Article ID 84256.

[23] C. Bettstetter and J. Eberspacher. Hop distances in homogeneous ad hoc networks. In Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual, volume 4, pages 2286–2290, April 2003.

[24] S. Dulman, M. Rossi, P. Havinga, and M. Zorzi. On the hop count statistics for randomly deployed wireless sensor networks. International Journal of Sensor Networks, 1:89–102, 2006.

[25] S. Mukherjee and D. Avidor. On the probability distribution of the minimal number of hops between any pair of nodes in a bounded wireless ad-hoc network subject to fading. In Proceedings of International Workshop Wireless Ad-Hoc Networks (IWWAN), May 2005.

[26] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic. Models and solutions for radio irregularity in wireless sensor networks. ACM Transactions on Sensor Networks TOSN, 2(2):221–262, 2006.

[27] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic. Impact of radio irregularity on wireless sensor networks. In Proceedings of MobiSys, pages 125–138, Boston, MA, USA, 2004.

[28] M. Zuniga and B. Krishnamachari. Analyzing the transitional region in low power wireless links. In Proceedings of the First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks SECON, pages 517–526, October 2004.

[29] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing forwireless networks. In Proceedings of MobiCom, pages 243–254, Boston, MA, USA, 2000.

[30] H. Frey and K. Pind. Dynamic source routing versus greedy routing in a testbed sensor network deployment. In EWSN '09: Proceedings of the 6th European Conference on Wireless Sensor Networks, pages 86–101, Cork, Ireland, 2009.

[31] H. Luo, F. Ye, J. Cheng, S. Lu, and L. Zhang. TTDD: Two-tier data dissemination in large-scale wireless sensor networks. Wireless Networks, 11:161–175, March 2005.

[32] G. Xing, C. Lu, R. Pless, and Q. Huang. On greedy geographic routing algorithms in sensing-covered networks. In MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing, pages 31–42, 2004.

[33] S. Basagni, I. Chlamtac, VR. Syrotiuk, and BA. Woodward. A distance routing effect algorithm for mobility (DREAM). In Proceedings of ACM MOBICOM, pages 76–84, 1998.

[34] T. Camp amd J. Boleng, B. Williams, L. Wilcox, and W. Navidi. Performance comparison of two location based routing protocols for ad hoc networks. In Proceedings of IEEE INFOCOM, volume 3, pages 1678–1687, 2002.

[35] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, volume 8, page 8020, Hawaii, USA, January 2000.

[36] K. Sha and W. Shi. Revisiting the lifetime of wireless sensor networks. In SenSys 2004: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, pages 299–300, Baltimore, MD, USA, November 2004.

[37] J. Li and P. Mohapatra. An analytical model for the energy hole problem in many-to-one sensor networks. In VTC-2005-Fall: Proceedings of IEEE Vehicular Technology Conference, volume 4, pages 2721–2725, Dallas, Texas, USA, September 2005.

[38] D. Wang, Y. Cheng, Y. Wang, and D. P. Agrawal. Lifetime enhancement of wireless sensor networks by differentiable node density deployment. In MASS 2006: Preccedings of IEEE International Conference on Mobile Adhoc and Sensor Systems, pages 546–549, Vancouver, BC, Canada, October 2006.

[39] E. Hyytiä and J. Virtamo. On traffic load distribution and load balancing in dense wireless multihop networks. EURASIP Journal on Wireless Communications and Networking, 2007:Article ID 16932, 15 pages, 2007.

[40] Intelligent transportation system (ITS). http://www.its.dot.gov/index.htm.

[41] J. C. Navas and T. Imielinski. Geocast-geographic addressing and routing. In MobiCom '97: Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking, pages 66–76, Budapest, Hungary, 1997.

[42] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. IEEE Personal Communications Magazine, 7(5):28–34, October 2000.

[43] Crossbow, inc. wireless sensor networks. http://www.xbow.com/Products/wproductsoverview.aspx.

[44] J. Hightower and G. Borriello. Location systems for ubiquitous computing. Computer, 34(8):57–66, 2001.

[45] D. Chen and P. K. Varshney. Geographic routing in wireless ad hoc networks. In Guide to Wireless Ad Hoc Networks, Computer Communications and Networks, pages 1–38. Springer London, 2009.

[46] S. Subramanian, S. Shakkottai, and P. Gupta. Optimal geographic routing for wireless networks with near-arbitrary holes and traffic. In INFOCOM 2008. The 27th Conference on Computer Communications. IEEE, pages 1328–1336, Phoenix, AZ,, April 2008.

[47] M. Z. Zamalloa, K. Seada, B. Krishnamachari, and A. Helmy. Efficient geographic routing over lossy links in wireless sensor networks. ACM Transaction on Sensor Network, 4(3):1–33, 2008.

[48] K. Zeng, K. Ren, W. Lou, and P. J. Moran. Energy aware efficient geographic routing in lossy wireless sensor networks with environmental energy supply. Wireless Network, 15(1):39–51, 2009.

[49] J. Li, J. Jannotti, D. S. J. D. Couto, D. R. Karger, and R. Morris. A scalable location service for geographic ad hoc routing. In MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking, pages 120–130, Boston, Massachusetts, United States, 2000.

[50] Z. J. Haas and B. Liang. Ad hoc mobility management with uniform quorum systems. IEEE/ACM Transaction Network, 7(2):228–240, 1999.

[51] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location information. In MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking, pages 96–108, San Diego, CA, USA, 2003.

[52] S. Lee, B. Bhattacharjee, and S. 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, Urbana-Champaign, IL, USA, 2005.

[53] Q. Chen, S. S. Kanhere, M. Hassan, and K.-C. Lan. Adaptive position update in geographic routing. In Proceedings of the International Conference on Communications (ICC 2006), pages 4046–4051, 2006.

[54] Silvia Giordano and Jean yves Le Boudec. Boudec. self-organizing wide-area routing. In In Proceedings of SCI 2000/ISAS 2000, 2000.

[55] Y.-B. Ko and N. H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. Wireless Network, 6(4):307–321, 2000.

[56] I. Stojmenovic. Home agent based location update and destination search schemes in ad hoc wireless networks. UTechnical Report TR-99-10, Computer science, SITE, University Ottawa, September 1999.

[57] G. G. Finn. Routing and addressing problems in large metropolitan-scale internetworks. Technical Report ISI/RR-87-180, Information Sciences Institute, March 1987.

[58] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. IEEE Transactions on Communications, 32(3):246–257, March 1984.

[59] E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In In Proceeding of 11th Canadian Conference on Computational Geometry, pages 51–54, 1999.

[60] I. Stojmenovic and X. Lin. GEDIR: Loop-free location based routing in wireless networks. In International Conference on Parallel and Distributed Computing and Systems, pages 1025–1028, 1999.

[61] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. Wireless Network, 7(6):609–616, 2001.

[62] F. Kuhn, R. Wattenhofer, and A. Zollinger. Worst-case optimal and average-case efficient geometric ad-hoc routing. In MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, pages 267–278, Annapolis, Maryland, USA, 2003.

[63] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric ad-hoc routing: of theory and practice. In PODC '03: Proceedings of the twenty-second annual symposium on Principles of distributed computing, pages 63–72, Boston, Massachusetts, 2003.

[64] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. Geographic routing made practical. In NSDI'05: Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation, pages 217–230, Berkeley, CA, USA, 2005.

[65] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. On the pitfalls of geographic face routing. In DIALM-POMC '05: Proceedings of the 2005 joint workshop on Foundations of mobile computing, pages 34–43, Cologne, Germany, 2005.

[66] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. Geographic routing made practical. In NSDI'05: Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation, pages 217–230, Berkeley, CA, USA, 2005. USENIX Association.

[67] D. Son, A. Helmy, and B. Krishnamachari. The effect of mobility-induced location errors on geographic routing in mobile ad hoc and sensor networks: Analysis and improvement using mobility prediction. IEEE Transactions on Mobile Computing, 3(3):233–245, 2004.

[68] M. Heissenbüttel, T. Braun, M. Wälchli, and T. Bernoulli. Evaluating the limitations of and alternatives in beaconing. Ad Hoc Netw., 5(5):558–578, 2007.

[69] K. Seada and A. Helmy. Efficient geocasting with perfect delivery in wireless networks. In Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE, volume 4, pages 2551–2556, March 2004.

[70] Q. Zhang and D. P. Agrawal. Dynamic probabilistic broadcasting in MANETs. Journal of Parallel and Distributed Computing, 65(2):220–233, February 2005.

[71] Y. C. Tseng, S. Y. Ni, Y. S. Chen, and J. P. Sheu. The broadcast storm problem in a mobile ad hoc network. Wireless Networks, 8(2):1022–0038, March 2002.

[72] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing, pages 194–205. ACM Press, 2002.

[73] Y.-B. Ko and N. H. Vaidya. Anycasting-based protocol for geocast service in mobile ad hoc networks. Computer Network, 41(6):743–760, 2003.

[74] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In INFOCOM '97: Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution, page 1405, 1997.

[75] I. Stojmenovic. Geocasting with guaranteed delivery in sensor networks. IEEE Wireless Communications, 11(6):29–37, December 2004.

[76] L. E. Miller. Probability of a two-hop connection in a random mobile network. In Conference on Information Sciences and Systems, Baltimore, MD, USA, 2001.

[77] X. Ta, G. Mao, and B. D. O. Anderson. On the probability of k-hop connection in wireless sensor networks. IEEE Communications Letters, 11(8):662–664, August 2007.

[78] X. Ta, G. Mao, and B. D. O. Anderson. Evaluation of the probability of k-hop connection in homogeneous wireless sensor networks. In Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE, pages 1279–1284, Washington, DC,, November 2007.

[79] I. Haque, I. Nikolaidis, and P. Gburzynski. Expected path length for angle and distance-based localized routing. In Proceedings of SPECTS 2009, pages 137–141, Istanbul, Turkey, July 2009.

[80] S. De. On hop count and euclidean distance in greedy forwarding in wireless ad hoc networks. IEEE Communications Letters, 9(11), November 2005.

[81] M. Zorzi and R. R. Rao. Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Multihop performance. IEEE Transactions on Mobile Computing, 2(4):337–348, 2003.

[82] J.-C. Kuo and W. Liao. Hop count distance in flooding-based mobile ad hoc networks with high node density. IEEE Transactions on Vehicular Technology, 56(3):1357–1365, May 2007.

[83] J.-C. Kuo and W. Liao. Hop count distribution of multihop paths in wireless networks with arbitrary node density: Modeling and its applications. IEEE Transactions on Vehicular Technology, 56:2321–2331, July 2007.

[84] S. Vural and E. Ekici. Analysis of hop-distance relationship in spatially random sensor networks. In MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing, pages 320–331, Urbana-Champaign, IL, USA, 2005.

[85] S. Vural and E. Ekici. Probability distribution of multi-hop-distance in one-dimensional sensor networks. Computer Network, 51(13):3727–3749, 2007.

[86] S. Vural and E. Ekici. On multihop distances in wireless sensor networks with random node locations. Accepted by IEEE Transaction on Mobile Computing, 2009.

[87] S. Nath and A. Kumar. Performance evaluation of distance-hop proportionality on geometric graph models of dense sensor networks. In ValueTools '08: Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools, pages 1–10, ICST, Brussels, Belgium, Belgium, 2008.

[88] D. Niculescu and B. Nath. DV-based positioning in ad hoc networks. Telecommunication Systems, 21(1-4):267–280, 2003.

[89] S. Dulman and P. Havinga. Statistically enhanced localization schemes for randomly deployed wireless sensor networks. In Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004. Proceedings of the 2004, pages 403–410, December 2004.

[90] C. Bettstetter. Connectivity of wireless multihop networks in a shadow fading environment. Wireless Network, 11(5):571–579, 2005.

[91] R. Hekmat and P. Van Mieghem. Connectivity in wireless ad-hoc networks with a log-normal radio model. Mobile Networks and Applications, 11(3):351–360, 2006.

[92] D. Miorandi and E. Altman. Coverage and connectivity of ad hoc networks presence of channel randomness. INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, 1:491–502, March 2005.

[93] C. Bettstetter and O. Krause. On border effects in modeling and simulation of wireless ad hoc networks. In Proceedings of IEEE MWCN, Recife, Brazil, 2001.

[94] C. Lantuejoul. Book: Geostatistical Simulation: Models and Algorithms. Springer, page 121, 2001.

[95] A. Baddeley, P. Gregori, J. Mateu, R. Stoica, and D. Stoyan. Book: Case Studies in Spatial Point Process Modeling. Springer, page 18, 2006.

[96] C. Bettstetter, H. Hartenstein, and X. Prez-Costa. Stochastic properties of the random waypoint mobility model. Wireless Networks, 10(5):555–567, September 2004.

[97] J. G. Jetcheva, Y. C. Hu, S. Palchaudhuri, A. K. Saha, and D. B. Johnson. Design and evaluation of a metropolitan area multitier wireless ad hoc network architecture. In Proceedings of Fifth IEEE Workshop on Mobile Computing Systems and Applications, pages 32–43, 2003.

[98] Dedicated short range communications (dsrc). http://grouper.ieee.org/groups/scc32/dsrc/index.html.

[99] G. Setiwan, S. Iskander, S. S. Kanhere, Q. J. Chen, and K. C. Lan. Feasibility study of using mobile gateways for providing internet connectivity in public transportation vehicles. In Proceedings of the ACM International Wireless Communications and Mobile Computing Conference (IWCMC), pages 1097–1102, Vancouver, Canada, 2006.

[100] G. Holland and N. Vaidya. Analysis of TCP performance over mobile ad hoc networks. Wireless Networks, 8(2):275–288, 2005.

[101] S. Bansal, R. Gupta, R. Shorey, I. Ali, A. Razdan, and A. Misra. Energy efficiency and throughput for TCP traffic in multi-hop wireless networks. In INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, volume 1, pages 210–219, 2002.

[102] Q. Chen, S. S. Salil, and M. Hassan. Analysis of per-node traffic load in multihop wireless sensor networks. IEEE Transaction on Wireless Communication, 8:958–967, 2009.

[103] Q. Chen, S. S. Salil, M. Hassan, and Y. K. Rana. Distance-based local geocasting in multi-hop wireless networks. In IEEE Wireless Communications and Networking Conference WCNC 2007, pages 4074–4079, Hong Kong, China, March 2007.

[104] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, and W. Hong. A macroscope in the redwoods. In SenSys 2005: Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems, pages 51–63, San Diego, CA, USA, November 2005.

[105] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler. An analysis of a large scale habitat monitoring application. In SenSys 2004: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, pages 214–226, Baltimore, MD, USA, November 2004.

[106] Y. Yu, R. Govindan, and D. Estrin. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023, May 2001.

[107] F. Kuhn, R. Wattenhofer, and A. Zollinger. Worst-case optimal and average-case efficient geometric ad-hoc routing. In MobiHoc '03: Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing, pages 267–278, Annapolis, Maryland, USA, June 2003.

[108] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. GHT: a geographic hash table for data-centric storage. In WSNA 2002: Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, pages 78–87, Atlanta, GA, USA, September 2002.

[109] Y. Chen and Q. Zhao. On the lifetime of wireless sensor networks. IEEE Communications Letters, 9(11):976–978, 2005.

[110] D. J. Dailey. Smart trek: A model deployment initiative. Technical Report WA-RD 505.1, Washington State Transportation Center, 2001. http://www.wsdot.wa.gov/Research/Reports/500/505.1.htm.

[111] Applications overview for federal intelligent transportation systems (ITS). http://www.itsoverview.its.dot.gov/.

[112] P. Nain, D. Towsley, Benyuan Liu, and Zhen Liu. Properties of random direction models. In INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, volume 3, pages 1897–1907, March 2005.

[113] C. Bettstetter, H. Hartenstein, and X. Pérez-Costa. Stochastic properties of the random waypoint mobility model. Wireless Network, 10(5):555–567, 2004.

[114] J. G. Jetcheva, Y. C. Hu, S. PalChaudhuri, A. K. Saha, and D. B. Johnson. Design and evaluation of a metropolitan area multitier wireless ad hoc network architecture. In Mobile Computing Systems and Applications, 2003. Proceedings. Fifth IEEE Workshop on, pages 32–43, Oct 2003.

[115] K. Seada, M. Zuniga, K. Helmy, and B. 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, Baltimore, MD, USA, 2004.

Appendix A

Simulation Algorithms

Recall that, in Chapter 3, we presented analytical models to estimate the hop count distribution and the mean values given the distance between a source and a destination. In Chapter 4, we analyzed the per-node traffic load in a wireless sensor network. In these two chapters, we have used custom C++ simulators to validate the analytical results. In this appendix, we give more details about the simulations. We first present the algorithms used in the simulator that generate the hop count results. Then we introduce the algorithm used for the simulation of per-node traffic load in a wireless sensor network.

A.1 Simulation of Mean Hop Count and hop count distribution

This set of simulation aims to calculate the average hop count and the distribution given the distance between a source and a destination in a random network. The simulation results are based on many iterations of individual simulation. In each iteration, the simulation follows the algorithm as below,

1. Randomly determine the number of nodes, which follows a Poisson distribution with mean of ρA (A is the area of the simulated two-dimensional

network).

2. Randomly assign the location for each node (nodes are uniformly distributed).

3. Construct the neighbor list for each node. The algorithm is outlined in Algorithm 1.

4. Find the hop count incurred in greedy routing for each pair of nodes. The algorithm is outlined in Algorithm 2.

5. group the pairs that have same source-to-destination distance together and record their hop count values. The algorithm is outlined in Algorithm 4.

Algorithm 1 Construct the neighbor lists for each node

for each pair of nodes (i, j) **do**
  $\beta \leftarrow$ computing signal attenuation from node i to j using Equation (3.11)
  if $\beta \leq$ the predefined threshold **then**
    //the threshold is the maximum attenuation that grants a direct link
    add node i to the neighbor list of j
    add node j to the neighbor list of i
  end if
end for

In Algorithm 3, for each pair of nodes that has a distance of d, we record its hop count to the set of Ω_d. Here the collective sets of all $\Omega_d, d \in N$ are global variables shared by all the iterations of individual simulations. At the end of all simulations, each set Ω_d contains the hop count values for all the sample pairs that have distance d between them. Based on these results, we can compute the mean hop count and the confidence interval. The algorithm is listed in Algorithm 4. Note that, the confidence interval is calculated based on 95% of confidence level. Therefore, with confidence level 95% the actual mean hop count lies in the confidence interval of the simualtion results.

Algorithm 2 Finding the hop count from node i to j

```
hopCount ← 0
currNode ← i
dstNode ← j
while currNode ≠ dstNode do
    if currNode is a neighbor of dstNode then
        nextNode ← dstNode;
    else
        nextNode ← the neighbor of currNode that is closest to dstNode
    end if
    currNode ← nextNode
    hopCount ← hopCount + 1
end while
return hopCount
```

Algorithm 3 Record hop count values

```
for each pair of nodes (i, j) do
    d ← the distance between the two nodes
    h ← the hop count between the two nodes (using Algorithm 2)
    add h to the set Ω_d, which consists of the hop count of all pairs having the
    same distance d
end for
```

The simulation of hop count distribution is similar as the simulation of mean hop count. Since we know all the hop count values for a distance d, we can easily get the probability distribution for each hop count value. The algorithm is described in Algorithm 5. Note that, Algorithm 5 is run after a certain iterations, e.g. m times, of individual simulations. However Algorithm 5 only gets one sample of hop count distribution. In order to get the average value (and confidence interval) of the hop count distribution, we need to repeatedly run the whole simulation again (each one includes m times of iteration). For example, in Chapter 3, we run 100 times of iterations to get one sample of distribution. Then we repeat the process 50 times and finally get the average value of hop count distribution and the confidence interval. Similar, we have used the same approach to simulate the forwarding probability (i.e. state transition probability) for Chapter 3.

Algorithm 4 Calculate the mean hop count

for each distance case d **do**
  n = the number of elements in set Ω_d
  denote h_k as the value of k-th elements in Ω_d
  compute the sample average hop count: $\overline{h} = \frac{\sum_{k=1}^{n} h_k}{n}$
  compute the sample standard deviation: $\delta = \sqrt{\frac{\sum_{k=1}^{n}(h_k - \overline{h})^2}{(n-1)}}$
  compute the 95% confidence interval as $[\overline{h} - 1.96 * \frac{\delta}{\sqrt{n}}, \overline{h} + 1.96 * \frac{\delta}{\sqrt{n}}]$
end for

Algorithm 5 Calculate the hop count distribution for a distance d

n = the total number of elements in set Ω_d
for each hop count value h **do**
  $n(h)$ = the number of elements in set Ω_d that have value of h
  hop count probability given distance d is $P(H = h|d) = \frac{n(h)}{n}$
end for

A.2 Simulation of per-node traffic load

This set of simulation aims to calculate the per-node traffic load in a circular random network. In the network, each node periodically sends a packet to the sink node that is located at the center of network. Packets are forwarded to the sink along greedy routing paths. The simulation results are based on many iterations of individual simulation. In each iteration, the simulation follows the algorithm as below.

1. Randomly determine the number of nodes, which follows a Poisson distribution with mean of ρA (A is the area of the simulated two-dimensional network).

2. Randomly assign the location for each node (nodes are uniformly distributed).

3. Construct the neighbor list for each node. The algorithm is outlined in Algorithm 1.

4. Find the greedy routing path from each node to the sink and update

the traffic load of each forwarding node. The algorithm is outlined in Algorithm 6.

5. Group the nodes that have same distance to the sink together and record their traffic load. The algorithm is outlined in Algorithm 7.

Algorithm 6 Find the greey routing path from a node i to the sink node

currNode $\leftarrow i$
while currNode $\neq$ sinkNode **do**
  if currNode is a neighbor of sinkNode **then**
    nextNode $\leftarrow$ sinkNode;
  else
    nextNode $\leftarrow$ the neighbor of currNode that is closest to sinkNode
  end if
  increase the traffic load of currNode by 1
  currNode $\leftarrow$ nextNode
end while

Algorithm 7 Record per-node traffic load

for each nodes i **do**
  $d \leftarrow$ the distance between from i to the sink
  $w \leftarrow$ the traffic load of node i (using Algorithm 6)
  add w to the set Γ_d, which consists of the traffic load of all nodes having the same distance d to the sink
end for

Once we have enough samples in Γ_d, we can use the algorithm similar to Algorithm 4 to compute the average value of the traffic load and the confidence interval.