NORTHWESTERN UNIVERSITY

Protocols, Architectures and Applications of Multi-Hop Wireless Networks

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Electrical Engineering and Computer Science

By

Alan R. Wolff

EVANSTON, ILLINOIS

December 2007

ABSTRACT

Protocols, Architectures and Applications of Multi-Hop Wireless Networks

Alan R. Wolff

Multi-hop wireless networks hold promise for increasing network capacity, lowering power requirements, and improving coverage over traditional cellular networks. However, their widespread adoption is hampered by challenges that include: 1) unreliable and complex routing protocols due to the transient nature of wireless nodes, 2) difficulties in guaranteeing quality of service to real-time applications, and 3) inefficiencies of medium access control in a dynamic wireless networking environment. In this dissertation, we tackle these problems in some specific multi-hop network environments.

First, we design low-complexity self-organizing and routing protocols for a large multi-hop network. We develop cell cluster-based routing trees and associated novel hierarchical routing and addressing approaches. Near optimum routing is achieved with a complexity of

essentially O(1), versus O(n^3) for conventional optimal routing, and performance is validated by simulation. We then design a real-time vehicle guidance solution with a dense wireless sensor network utilizing our routing approach. It features a communication subsystem and a vehicle routing subsystem; the former gathers real-time vehicle data and distributes guidance information while the latter processes vehicle data and makes guidance decisions. Very small communication bandwidth is shown to be required to deliver true real-time vehicle guidance.

Second, we develop a virtual circuit communication protocol that supports connection-oriented applications such as voice and streaming video that exploits load balancing multiple-routing trees to minimize connection blocking rate. Lower bounds on blocking rate are analyzed and used to evaluate the performance, and simulations verify the results.

Finally, we consider a medium access control problem in a two-hop wireless network where the base station, fixed relays, and mobiles within a cell share a reservation-based TDMA channel. We present a scheduling solution that seeks to identify the optimum uplink path from each mobile while allocating time slots in such a manner that queueing delay at the relays is essentially negligible. We quantify conditions whereby relays offer performance advantages and show analytically that throughput gain due to optimum relay increases dramatically as path loss becomes more severe. Simulations also show that optimum relay yields a significant throughput advantage compared with a one-hop approach and un-optimized two-hop relay approach.

To my beloved wife Vivian, and my children Sarah,

Esther, Grace and David: this one is yours too.

ACKNOWLEDGEMENTS

First of all, I thank my Lord Jesus Christ who has enabled this. All honor goes to him.

I thank my advisor Dr. Chung-Chieh Lee who gave me this opportunity. I have no words to express how valuable his encouragement, challenge, and wisdom have been. I also want to acknowledge my committee members, Dr. Abraham Haddad and Dr. Randy Berry who have been supportive and have offered good words of advice

My wife Vivian made many sacrifices while I was working on my studies. She also pushed me and our kids to "study hard" and strongly encouraged me to keep at it when I wanted to give up. I am a lucky man.

I must acknowledge the support I have had from my employers at Northwestern University. I thank Rebecca Dixon who allowed me to pursue this and take time for classes during the day, and work into the evening. I am thankful to Michael Besancon, who understood during the especially stressful periods of doing two full-time jobs. I genuinely appreciate the kind words of encouragement from John Birge on a few occasions. Jay Walsh gave me great counsel many times in the last couple years of this journey and seemed to know exactly what I was going through - that was a huge help. Julio Ottino has been supportive as well. I am very thankful for the many great coworkers I've had over the years who have accepted me in my sleep-deprived state and who have enabled me to be a part of interesting things at work too.

The love and support I have received from my church family has also been instrumental, starting from Dr. Samuel Lee and Sarah Barry, my church pastors. Dr. Lee planted hope in me to do something great when I only wanted to be a mediocre person. Sarah Barry has been there to listen to me and give me wise advice: “just pray about it.” I have always felt her strong support, love and prayers for me. I have also felt a strong bond with many people in our fellowship, which has helped to sustain me. Among them are Chris and Christine Kelly, Dr. Nathan and Mary Walker, Tony and Joy King, Hosea and Annie Lee, Grace Jeon, Dr. Sarah Han, Viola Koti, Francesca Park, Paul Chung Jr., Richard Choi Jr., Agnes Sohn, Dr. Oleg, Dr. Jung, Taneisha Robinson, Brian Annear, Kristin Lawson, Dr. Gae’s family and many others. Jacob A. Lee and David Brogi in Washington have been an inspiration to me. Pastor Ron Ward, Pastor Kevin Albright and Dr. James Kim have been an encouragement. I owe a debt of gratitude to them all.

I am appreciative of working with several fellow students during my studies. I want to acknowledge Brad Weyers and Jason Perlewitz who were undergraduates when we took classes together and let me join their study group. Two students who finished their Ph.D. before me and helped me tremendously with the class material are Yishen Sun, and Sak Santipach. I am especially grateful to Hyunjeong Lee whose research forms some of the basis for my work. I am thankful that Aaron Ballew was there to cheer me on as I was finishing this quest. I also wish to thank Ron Widitz who has been very mindful of this adventure that I’ve been on.

My mother and father, Jeanette and William Wolff, have something to be proud of their son for. I am thankful for the good influence they gave me; their love and support over many years cannot be understated. I really wish that my mother could be here to witness this but believe that she somehow knows. I am thankful to my brother, Eric, his wife Kathy, and their children Timothy, Hannah, Jonathan, Daniel for their support, prayer and encouragement. I am thankful also to Norah Wolff who has shown her love in tangible ways.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction and Background

1.1 Wireless Technology and Trends

It has been stated that the vision of computing in the 21^{st} century is to provide pervasive computing environments that can seamlessly and ubiquitously enable users to communicate at any time, anywhere and from any device [1]. Today wireless communications continues to rapidly develop. Applications of broadcast radio, television, cellular phones, short-range point-to-point communication, wireless sensor networks, and wireless computer networks continue to further permeate our day to day lives. The FCC reports that U.S. cell phone usage now exceeds landline usage, with close to 200 million American cell phone lines in operation. Laptop computers with wireless connectivity now outnumber desktops and, as a result, businesses continue to integrate wireless technology into their daily operations. The growing usage of smart phones such as Blackberries, Treos and iPhones has brought mobile e-mail and Internet access into the mainstream. However, in spite of these recent advances, the vision of universal access to a ubiquitous wireless network that adapts itself to challenges in its environment, provides multiple easy to use services for people at a low cost, and simplifies service provider management by seamlessly integrating the underlying complexity is very far off.

The wireless network architectures and protocols described in this dissertation are intended to capitalize on the continued development of low-cost, low-power, and low-complexity devices that will enable broader and more easily managed network access. This is achieved with a multiple hop cellular architecture that takes advantage of the benefits of wireless mesh networking and the use of cellular-type base stations.

1.1.1 Mobile Ad Hoc Networks (MANets)

There has been a tremendous amount of research activity in Mobile (multi-hop) ad hoc networks (MANets) that has generated an ocean of literature in recent years [2]. A MANET is a kind of wireless network where mobile nodes, which may be hosts or routers or both, move freely and organize themselves into an arbitrary and temporary topology. The goal is to extend the reach of conventional networks and allow for seamless internetworking. Figure 1.1 is a depiction of a set of nodes that form a topology based on their wireless communication radius. As they move, nodes in a MANet need to dynamically establish communications. Examples of applications of MANets are Department of Defense (DoD) applications that require communications on the battlefield and specialized civilian applications such as emergency communications for disaster recovery. There is also much research into vehicular ad hoc communications where automobiles are both wireless network nodes and routers. In such networks challenges arise because node mobility causes frequent failure through the activation and deactivation of links, leading to complexities and routing problems.

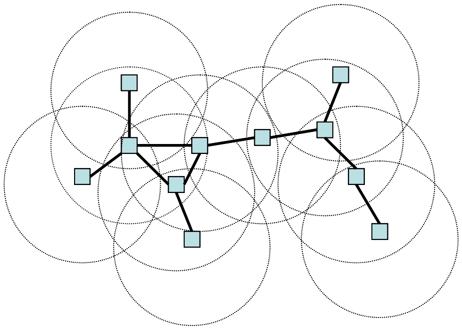


Figure 1.1: Mobile Ad Hoc Network where nodes within range of each other can communicate and construct a dynamic wireless typology.

1.1.2 Wireless Mesh Networks (WMNs)

Strictly speaking, MANets are made up of user devices only with no pre-existing infrastructure. A new class of networks, called *mesh networks* [3, 4], has emerged that relaxes this constraint. Wireless mesh networks (WMNs) are built on a mix of fixed and mobile nodes interconnected via wireless links to form a multi-hop ad hoc network. These WMNs are made up mesh routers and clients, where the routers have minimal mobility and form the backbone of the WMN. The integration of WMNs with other networks such as the Internet, cellular, IEEE 802.11, IEEE 802.15, IEEE 802.16, sensor networks and the like can be accomplished through gateway and bridging functions in the mesh routers. Mesh clients can be either stationary or mobile, and form a client mesh network among themselves and with mesh routers.

Figure 1.2 shows a generic example of a three-tier architecture for wireless mesh networks. Here, nomadic users are provided a service by other nodes that are dedicated routers. The nomadic users may form a multi-hop MANet amongst themselves. In the same way, routers may form their own multi-hop network and they forward their messages to and from an access point. The network of wireless routers forms a wireless backbone which provides multi-hop connectivity between nomadic users and wired gateways. The meshing among the wireless routers and access points creates a wireless backhaul communication system, which provides each mobile user with a low-cost, high bandwidth, and seamless multi-hop connection service [4].

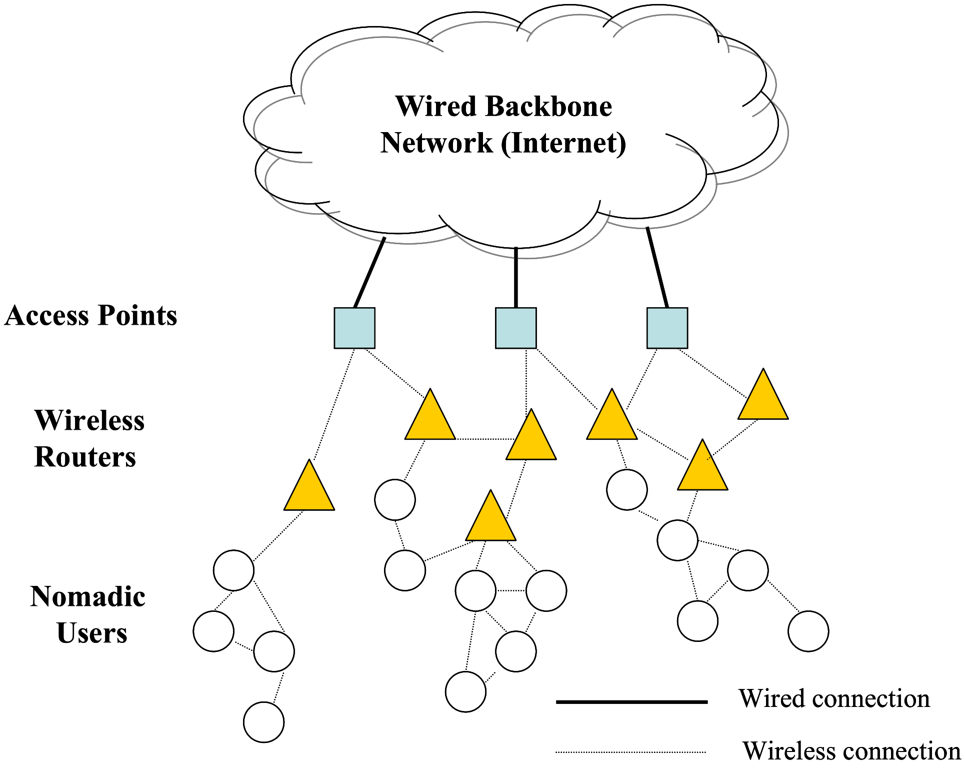


Figure 1.2: A 3-tier architecture for wireless mesh networks.

1.1.3 Multi-hop Cellular Networks (MCNs)

The specific WMN considered in this dissertation is an implementation of a *multi-hop cellular* architecture with fixed relay nodes. A Multi-hop Cellular Network (MCN) is an architecture originally proposed by Lin and Hsu for wireless communications [5]. MCNs combine the benefits of a fixed cellular infrastructure of base stations and the flexibility of ad hoc networks. The further development of adding fixed relay nodes to MCNs has enabled more stable routing, as well as improved load balancing among cells [6, 7]. The result is an architecture that is roughly analogous to that of Figure 1.2, with the base stations, fixed routing nodes and mobile users corresponding to the 3 levels of the mesh network architecture. An implementation of this is shown in Figure 1.3 where a mobile user in the vehicle may receive messages from a cellular base station (BS1), and then send a message upstream through the relay nodes and then to the Mobile Switching Center (MSC).

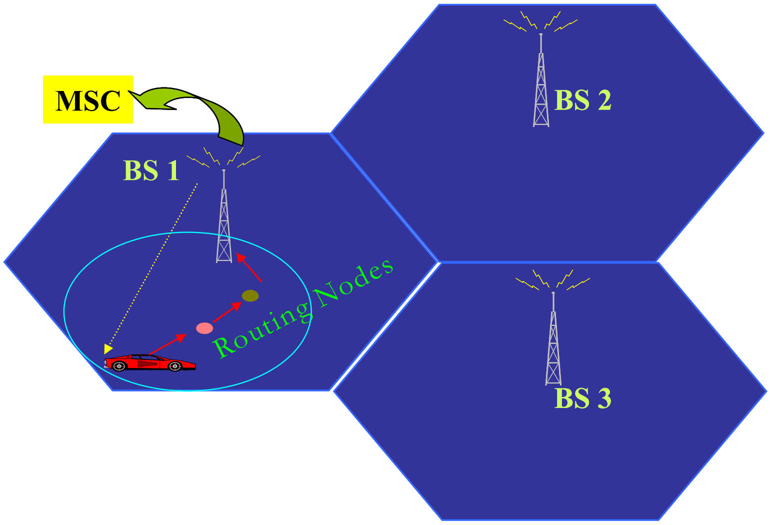


Figure 1.3: A Multi-hop Cellular Network (MCN) with fixed routing nodes.

1.2 Integrated Multi-hop Cellular Data Network (IMCDN)

Numerous types of ad-hoc multi-hop wireless protocols have been proposed, and the vast majority have a routing path to every other node in the network, as in a peer-to-peer network. Since there may be a very large number of nodes in the network and the state of the network may be time-varying, maintaining network tables could be extremely costly in power consumption and computational complexity. Therefore, we have built our research on a multi-hop self-organizing network with a routing hierarchy consisting of multiple routing trees. This hierarchical structure exhibits good economies of scale: as a network becomes large, routing table storage and processing overhead at each node is minimized. Our work uses a hierarchical routing protocol using routing trees developed in [8] and the protocols, architectures, and algorithms that optimize performance, increase performance and enhance reliability introduced in [9].

The focus of this work is on network layer and MAC-layer schemes while making general assumptions about the physical layer characteristics of the network environment. The proposed network is designed to operate in any RF spectrum using any modulation format, any data rate and any coding technique. The only requirement placed on the shared medium is that neighboring nodes are able to detect each other’s transmissions with sufficient accuracy. While many of the designs in this thesis were developed with the IEEE 802.16 protocol in mind and may be applicable to other protocols such as IEEE 802.11, we aimed to design protocols and architectures with more general applicability.

1.2.1 Self-Organizing Network with Hierarchical Routing

The self-organizing network introduced in [8] has nodes that self-organize to create a routing infrastructure whereby multi-hop communication is supported using rooted spanning trees. The network is densely populated with a large number of quasi-stationary nodes, where the cost per node and power per node are low. The nodes are equipped with wireless transceivers that allow them to communicate directly over a short transmission range. In order to transmit data to a final destination outside its range, a node must rely upon one or more intermediate nodes to relay messages. The rooted spanning trees are a routing topology where a high power node, such as a base station, is the root and is capable of transmitting directly to destination nodes in the network. The root is responsible for capturing and maintaining the routing hierarchy and the complete physical topology, and for coordinating medium access. An example of a possible physical topology and one possible associated routing topology is in Figure 1.4a and Figure 1.4b.

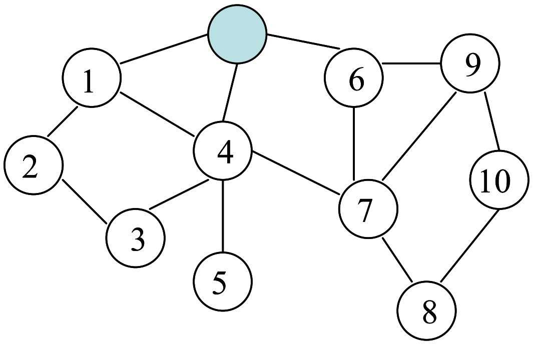


Figure 1.4a: Physical topology of a high power root node and 10 network nodes.

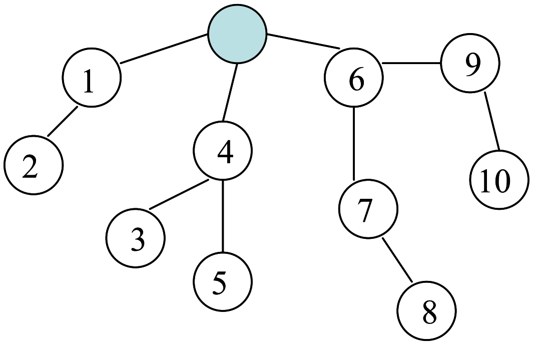


Figure 1.4b: A routing topology with spanning trees rooted at high power root node.

The rooted spanning tree routing topology allows every node in the network to route data upstream to the root. In the tree structure, every non-root node is connected to at least one parent node via a wireless link and every parent node may have multiple children, while every child node must have only one parent. Additionally, a root node is assumed to be of sufficient power to broadcast messages downstream to all of the non-root nodes. In order to communicate to a destination node, data messages must be relayed to the root following the routing hierarchy. Then the high power root broadcasts the message to the destination node within that cell. Thus, as an example, in Figure 1.4b, the shaded root node may directly broadcast downstream to Node 5 while Node 5 routes upstream data messages through its parent node (Node 4) and then to the root node. Each message contains two destination address fields: the parent node address for the next hop and the final destination address.

1.2.2 Multiple Spanning Tree Routing Topology

The work of [8] deals exclusively with a single routing tree, that is, there is only one hierarchical routing topology in the network. The routing tree topology construction algorithm developed in that work applies also to this work. This algorithm was proved to be optimal in terms of hop count from each node to the root and was proved to have minimal computational, communication, and signal complexities. However, the limited capacity of low-power nodes and the increased delay caused by the physical expansion of the network will ultimately place an upper bound on the capacity and size of the network. To make the network more scalable, a multiple tree routing topology was developed in [9] that utilizes multiple high power root nodes, each connected to a high bandwidth connection to a mobile switching center and capable of broadcasting user data messages downstream directly to the destination of the message. Each high power root is assumed to have its own carrier frequency and is the root of a spanning tree that encompasses all of the nodes in the network. Figure 1.5 shows a two-tree network, where the two trees are alternatively used to send successive packets. This enables better load balancing and reliability.

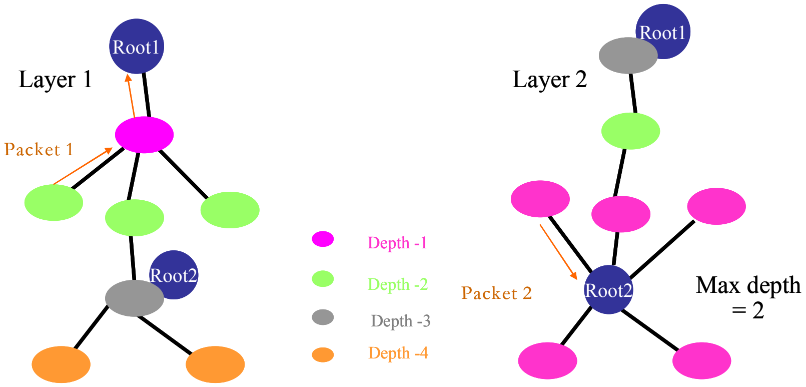


Figure 1.5: 2-layer spanning trees for routing messages.

Figure 1.6 shows an example of an implementation called Integrated Multi-hop Cellular Data Network (IMCDN). Here, there are three base stations, each with a high bandwidth connection to the Mobile Switching Center and with several other nodes within their broadcast range. The three base stations are each roots of a spanning tree that make up a routing topology that exists throughout the network. Thus, each non-root node in the network is part of 3 routing trees, one to each of the roots in the system. In the routing example shown, Root 1's tree is congested, but since there are three routing alternatives for the message a less congested route, through Root 2, to a base station is chosen. Once the message is relayed to Root 2, is it sent to the MSC and to Root 3, where it is broadcast to the destination that is within broadcast range of Root 3.

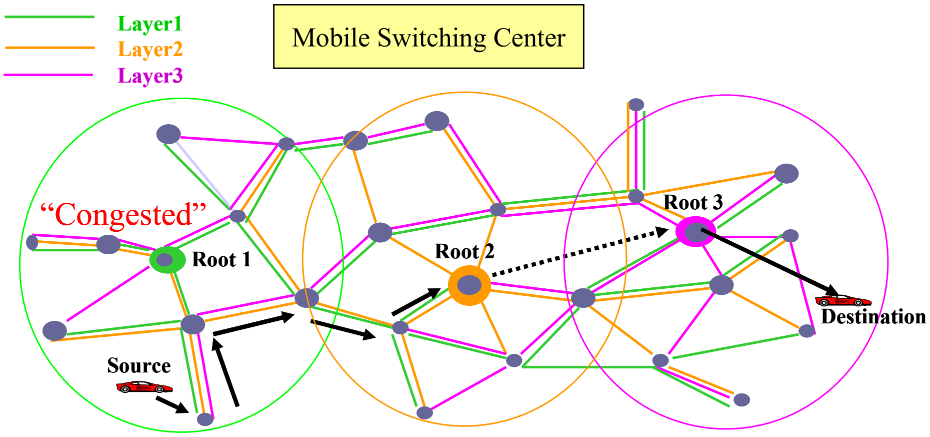


Figure 1.6: Integrated Multi-hop Cellular Data Network

1.3 Introduction to Large Scale Self-Organizing Network (LSSON)

The developments of [8] and [9] enable low-power, low-complexity nodes in a self-organizing multi-hop network to communicate with one another. The network also adapts to failures and changes. Moreover, the multiple trees facilitate load balancing and improve reliability. However, the design of network assumes that every root node is connected to a high bandwidth backhaul network. While this is a typical case for a conventional cellular network, such an assumption does not necessarily apply to sensor networks, networks that need to be rapidly deployed in the battlefield or temporarily deployed in the case of emergencies, and in networks of the future that will need to efficiently extend coverage, without a large upfront investment in a conventional cellular base station. In addition, an IMCDN does not limit the reach of the spanning trees in the system. Conceivably these trees could span the entire network and if there are many roots in the system this would present complexity issues for the presumably low-complexity and low-cost nodes. To better manage these problems, in this dissertation, we introduce a novel large scale multi-hop network that limits routing trees to cell clusters.

1.3.1 The Traditional Cell Cluster Concept

In cellular systems, a cell is defined as the area where signal power from a base station greater than some threshold for acceptable reception by users. Conceptually, base stations are placed at the center of each cell with the assumption of omni-directional antennas, uniform propagation, and flat terrain. With line-of-sight propagation, the coverage of a base station can be modeled by a circle centered at the base station. The coverage of each base station is overlapped with the coverage of neighboring base stations. Thus some nodes can be in one, two, three or more cells simultaneously depending on their location. Cells are grouped into clusters, taking advantage of the frequency re-use concept. Within a conventional cellular frequency reuse framework, if a cellular system has a total of *S* total channels for use, each cell is allocated *k* channels, where $k \sim S/N$, with *N* being the number of cells in a cluster. In other words, a group of *N* cells which collectively use the complete set of available frequencies is a cell cluster [10]. The *N* cells that make up a cell cluster each have coverage ranges that overlap so that a cell cluster is formed from a cell and all of its adjacent neighbors.

1.3.2 Cluster-Based Bounded Spanning Trees

Let C_k denote the cell associated with Base Station (BS) *k*, and Θ_k the cell cluster centered around C_k. That is, Θ_k consists of C_k and all of its neighboring cells and covers an area in which each point belongs to one or more cells that make up Θ_k.

In IMCDN, each routing tree spans the entire network such that every routing node is a member of every routing tree. That is, each routing tree is a spanning tree rooted at a unique base station. The root of each spanning tree is a base station which is responsible for supervising the construction and maintenance of its own spanning tree using algorithms developed in [8]. As mentioned before, the routing and routing topology maintenance may become prohibitively complex when the size of the network increases and/or when the number of trees increases.

In this dissertation, we present a multiple-layer hierarchical routing strategy in which a routing tree is bounded to a “cell cluster.” When the system is initially set up, each routing tree is created only for a cell and is rooted at a high power base station. Only routing nodes within the range of a particular root join that tree. The fixed routing nodes are assumed to be of low power, limited in their transmission range and processing capability. Each fixed routing node must be within range of at least one another active fixed routing node. In order to use the multiple-hop network to communicate with other mobile nodes, a mobile in the system must be within signal range of at least one fixed routing node. When a mobile has a message to send, it is sent to a fixed routing node which then forwards the message along a selected routing tree toward a base station.

To maintain routing node simplicity and reign in the routing complexity, we limit the reach of each routing tree to a cell cluster associated with the root of tree as shown in Figure 1.7. This diagram shows a routing tree rooted at Base Station A extending no further than the neighboring cells that make up the cell cluster associated with A. As a result, a routing node does not belong to all routing trees in the system but only belongs to a limited number of cluster-based routing trees, where the number of trees is essentially the number of cells in a cell cluster. In the case of Figure 1.7, all routing nodes in Cell A belong to seven routing trees, one for each base station in the cell cluster. Only one of the seven routing trees for the cluster is shown in Figure 1.7.

Intra-cluster message delivery is very simple using this scheme. When there is message to be sent to a destination mobile within a particular cell, the candidate routing trees associated with that cell are examined. The routing tree centered around the cell that has the least number of hops or least loaded to the root in which the mobile is located will be selected and the message will be forwarded along that routing tree until it gets to the root whose transmission range covers the destination mobile; that root then delivers the message to the destination mobile via single-hop broadcast. On the other hand, inter-cluster routing is not so simple. We assume that roots may not be inter-connected by a high bandwidth backhaul infrastructure such as a broadband connection to the Internet. In this case, the message must be forwarded from one routing tree to another until the message gets to the cell cluster that contains the destination. Once it is there, the message can be delivered by the appropriate root to the destination. We shall discuss means for solving this large scale routing problem using the architecture we have presented.

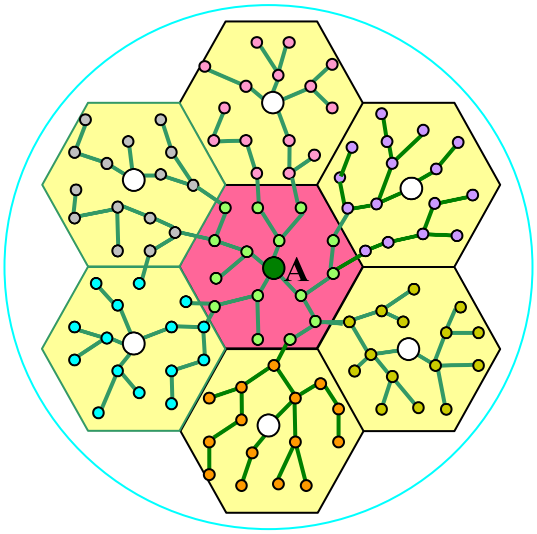


Figure 1.7: Cluster-based routing tree for Root A

1.4 Thesis Overview and Contributions

This work expands on previous research and presents new solutions to address challenges of large-scale routing, servicing real-time traffic, and medium access control in multi-hop wireless networks. In addition, we also discuss an application of large-scale wireless sensor network to real-time vehicle guidance, a hot issue in Intelligent Transportation Systems.

We introduce the basics of our network architecture, dubbed “Large Scale Self-Organizing Network” (LSSON) in Chapter 3 by showing how it is constructed and describing the protocols for its operation and maintenance. A hierarchical routing methodology for LSSON is described that scales well to a very large number of nodes. The hierarchical routing scheme involves two-tiers: a lower tier for node-level paths using tree-based routing and an upper tier for large scale cell-level paths. Lower-tier routing involves simply routing to the parent node of the tree with respect to a base station. Once there, the message is broadcast to the destination user. The upper-tier routing is explored using both a hierarchical geographical partitioning approach and a radial geometric approach. A significant complexity reduction compared to other routing methods is quantified and the efficiency is demonstrated to be excellent through a series of very large scale routing simulations using nearly 700,000 routing nodes.

In Chapter 4 we tackle the problem of servicing connection-oriented traffic, such as voice and streaming audio/video, in the multi-hop network by allocating network resources through the use of virtual circuits. We leverage the tree routing structure of LSSON to efficiently establish, track, and tear down virtual circuit connections in the network when necessary. Dynamic routing trees are utilized to re-configure virtual circuits in order to maximize the capacity of the network. The capacity of the LSSON architecture using virtual circuits is found to be limited by the nodes that are one hop from the base station, called “top level nodes,” assuming that all routing nodes have the same capacity. Expressions for blocking rate are developed as theoretical limits on the performance of LSSON virtual circuits. The results of a series of simulations of the LSSON virtual circuit environment are presented to compare performance to the theoretical limits. We determine that while network topology issues and load distribution may pose significant challenges to performance, the use of multiple routing trees in LSSON may effectively distribute traffic from “hot spots” in the network among several base stations, providing robust load balancing capability.

Chapter 5 applies the LSSON network architecture and associated protocols to the area of Intelligent Transportation Systems involving real-time vehicle guidance. The wireless multi-hop network we have devised is proposed as an effective means of gathering real-time vehicle data, communicating it to a vehicle guidance subsystem, and then delivering the processed up-to-the-second vehicle guidance information to motorists. In our system architecture, communication functions and vehicle route processing are separated into distinct subsystems, providing ubiquitous communication through a roadside mesh-like sensor network and a design framework for low-complexity real-time vehicle routing table computation. An analysis of the bandwidth requirements associated with the proposed architecture and communication protocols demonstrates the feasibility of the automated real-time vehicle routing system. Through the use of roadside sensor locations as entries in vehicle routing tables, processing is minimized within the vehicle. Vehicle routing complexity is drastically reduced by using an on-demand routing strategy and a hierarchical vehicle routing approach which separates local routing from long-distance routing.

In Chapter 6 we study medium access control (MAC) scheduling problem in the multi-hop communication network, where the wireless channel is shared by the base station, relay station, and the mobile stations using reservation-based TDMA. The conditions under which there are improvements in user data throughput by the effective use of relay nodes in a multi-hop cellular network are discussed. It is shown that environments of high path loss are where relay stations can provide the most benefit to subscribers. A MAC scheduling solution for a relay based two-hop wireless network is then presented. In this centralized scheduling design, the BS chooses the best relay station for each SS/MS to communicate through, it properly allocates the TDMA data slots that are shared by all the active nodes in the cell and effectively manages the queues in the relay nodes so as to render queueing delay in the relay nodes a negligible part of overall delay for users. We show analytically and by simulation that this can improve system throughput through significantly improving channel qualities, compared to a single-hop cellular network, despite entailing two transmissions for delivering each packet.

Chapter 7 summarizes the contributions of this thesis and outlines areas for future research.

Chapter 2

Related Research

2.1 “Hybrid” Networks

Among the most attractive features of peer to peer multi-hop wireless networks are that they obviate the need for a costly fixed infrastructure, they can extend the reach of a network by the addition of new nodes, they can lower the power requirements of nodes since transmissions only need to be sent to immediate neighbors, and the raw capacity of the network may be increased due to high spectral efficiency. However, these advantages may be offset by decreased end-to-end throughput with many hops, inefficient and complex large scale routing protocols especially with mobile routing nodes, and greater difficulty in providing quality of service guarantees. Traditional fixed node wireless networks, such as those with cellular base stations, have strengths at precisely the points where peer to peer multi-hop wireless networks are the weakest. They provide a one-hop path to a base station which can then forward the message along a high speed network infrastructure to the destination. Thus, they can support mobile users through a static and centralized routing scheme and can also provide a quality of service based on bandwidth guarantees within the system. In recent years, there has been much work done to exploit the benefits of both peer-to-peer multi-hop wireless networks and cellular networks through a “hybrid” approach: networks which contain a central access point and which also contain multi-hop routing nodes. In 2004, the IEEE 802.16 Working Group integrated a “mesh mode” specification into its standard [11] that allows for multi-hop message forwarding within the reach of the network. Moreover, multi-hop relaying has been incorporated into the European HIPERLAN/2 standard [12]. Examples of working multi-hop wireless networks are those developed by MeshNetworks Inc [13], which was recently acquired by Motorola and MIT's Roofnet project [14], among many others.

2.1.1 Theoretical Justification of "Hybrid" Networks

A "hybrid" model has been justified on many fronts. In Gupta and Kumar's seminal work "Capacity of Wireless Networks," they analyzed the maximum throughput achievable by wireless networks. When n identical randomly located nodes, each capable of transmitting at W bits per second and using a fixed range, form a wireless network, the throughput obtainable by each node for a randomly chosen destination is $\Theta(W / \sqrt{n}$ log n) bits per second under a noninterference model [15]. Further, they found that under optimal circumstances the throughput is only Θ (W / $\sqrt{n}$) bits per second for each node to a destination, making it clear that an ad hoc wireless network is not exceptionally scalable since as the number of nodes becomes large, the capacity diminishes rapidly. If some fixed wireless infrastructure is added to such an ad hoc network, it has been found that per node capacity can be increased significantly to $W /$ log$n$ [16]. The number of base stations in the hybrid network is an important variable. In a large hybrid network, a network with both ad hoc nodes which communicate with each other via shared wireless links of capacity W bits/second and infrastructure nodes which are interconnected with each other via high capacity links, analysis has been done on the optimal number of base stations relative to ad hoc nodes [17]. With m infrastructure nodes and n ad hoc nodes, when $m \leq \sqrt{n/\log n}$ there are a relatively few base stations and the per user throughput is only W/$\sqrt{n/\log n}$. This could be realized by only ad hoc nodes and the few infrastructure nodes are not needed. However, when $\sqrt{n/\log n} \leq m \leq n/\log n$ per user throughput improves to Θ(Wm/n) and the total additional bandwidth provided by *m* infrastructure nodes is effectively shared among the ad hoc nodes. Finally when there are more base stations, $m \geq n/\log n$, per user throughput has an upper bound of Θ(W/logn), suggesting further investments in infrastructure nodes will not lead to improvement in throughput.

Percolation theory [18] describes the connectivity of nodes in a random graph. Based on this theory, Gilbert [19] studied the multi-hop connectivity of wireless broadcasting stations using point locations generated by a two-dimensional Poisson point process to represent wireless transmitting stations of a certain range he defined as 2*r*. He showed that there exists a certain critical value $\lambda_{c,}$ the density of transmitters, such that, for $\lambda > \lambda_c$, a connected component of transmitters is formed (the network "percolates") with probability of one. Therefore, every node in the network is accessible to every other node and multi-hop communication is possible among the nodes. Since then, there has been significant further study in network percolation [20], [21] with different topologies. In a one-dimensional line topology, it has been shown that as $\lambda < \infty$ the probability of a vacant interval in an ad hoc network to occur between 2 consecutive nodes is strictly positive, whatever the value of λ and the range of the transmitters. In this case, since there are an infinite amount of intervals and since their lengths are independent, the probability of having no holes is 0. In recent years, there has been study on how the addition of fixed nodes in the network, such as base stations that characterize a hybrid model, affects percolation. In the two dimensional model, the introduction of base stations has been shown to have only a minimal improvement in connectivity, but in the one dimensional model, such as along a highway for example, a dramatic improvement in connectivity has been demonstrated [22]. Further, it has been shown that the placement of nodes in particular locations rather than in random locations increases the connectivity and reliability of a network. Moreover, in the face of possible node failures, placement of nodes in a grid topology can ensure a high level of connectivity, with the maximum number of hops to travel from any active node to another on the order of $\sqrt{n / \log n}$ [23].

2.1.2 Examples of "Hybrid" Networks

In recent years, there have been numerous proposals for next generation networks which exploit the qualities of both conventional cellular networks and peer-to-peer wireless networks. In [24] the authors studied the performance trade-offs between the two and then propose a hybrid network with a dual mode of operation that uses a peer to peer model in tandem with a conventional cellular network model. Simulations were done in terms of throughput, delay, power consumption, per-flow fairness, impact of mobility, impact of traffic locality, and impact of node distribution on network performance. While the pure peer-to-peer network performed better in terms of throughput, delay and power, the cellular network proved superior with per-flow fairness, performance in the event of mobility and when large scale routing was involved. The hybrid network presented in [24] used TDM. By default the network operated in the peer to peer mode during all of the TDM slots but as flows increased on nodes, nodes began to use more of their TDM slots to operate in cellular mode. Simulations showed that this dual mode hybrid network can overcome the disadvantages of each of the two different kinds of networks separately. More recently there has been much work done on what has become known as “Multi-hop Cellular Networks.” These differ from the hybrid network in that there is only one mode of operation instead of two. In [25] the authors defined single hop cellular networks (SCN) as a regular cellular network where base stations can be reached by mobiles in a single hop (and vice versa) and defined a Multi-Hop Cellular Network (MCN) as a cellular network where base stations may be reached by mobiles in multiple hops. In a MCN, relaying is performed by the mobile nodes if a mobile cannot reach a base station in a single hop directly. Moreover, mobiles are allowed to communicate with one another without the involvement of a base station. This approach can extend the reach of a base station and allows for mobile nodes within a cell to communicate with each other without use of the base station, thus augmenting network capacity. Other similar network architectures using the Multi-hop Cellular approach include Unified Cellular and Ad-hoc Network [26] (UCAN), Pervasive Ad-hoc Relaying for Cellular Systems [27] (PARCelS), and Cellular Aided Mobile Ad hoc Network [28] (CAMA). A common element in these systems is that they use mobile nodes as routers, requiring a high level of complexity in the routing protocols and significant sophistication in the mobile nodes compared to using fixed routing nodes. In [29] a multi-hop cellular architecture called, “Hierarchical Cellular Multi-hop Networks” was presented which uses a multi-hop downstream channel to improve system throughput using a hierarchy of different types of wireless networks, including those with base stations, access points and mobile nodes. The option of fixed nodes is also mentioned.

2.1.3 Multiple-Hop Wireless Networks with Fixed Routing Nodes

As in this dissertation, there has been some work done in multi-hop cellular systems with fixed routing nodes, which may also include nomadic routing nodes. A relatively early idea that incorporated both the use of fixed wireless relay nodes and base stations was the Integrated Cellular and Ad-Hoc Relaying System [30] (iCAR). The basic idea of iCAR is to place a number of ad hoc routing stations (ARSs) at strategic locations, which can be used to relay signals between mobile hosts and base stations. Not only does this relieve mobile nodes from the routing function, but by using ARSs it becomes possible to divert traffic from one possibly congested cell to another non-congested cell. The benefits of this system include load balancing among cells in a cellular network, extending a cell's reach and enhancing network reliability. The authors analyzed the blocking ratio through load balancing among cells using the Erlang-B model and considered only circuit-switched traffic. Moreover, they allowed the use of mobile hosts as routing nodes in a "secondary relaying" mode in which there are no ARSs accessible to a given mobile host. In order to route messages without the use of mobile relaying nodes, the authors estimated that with R being the range of the base station, and r being the range of the ARSs, and the value n being R^2/r^2, then approximately $2n$ ARSs would be required in a cell to provide this coverage [31]. In [32] a centralized downlink for a cellular network with multi-hop transmission through fixed wireless relays is considered. They found that performance is dramatically increased over a system without relays due to spatial re-use. With relays placed in high traffic locations large gains could be expected. In [33] the authors analyzed relay node placement algorithms based on the data rates received from the base station in different locations.

They quantified improved data rates through simulation that can be realized by deploying fixed relay nodes in locations of weak reception from the base station. A “Wireless Media System” (WMS) is proposed in [34] and [35] that uses fixed relay nodes to provide coverage to “shadowed” areas. It makes use of different forwarding techniques and positions fixed wireless routers to improve urban coverage. The fixed nodes, used as forwarding nodes to enable access to “shadowed” areas such as around the corner of a building, are shown to significantly improve network capacity and coverage. In [36] several wireless topologies are considered in order to support real-time multimedia services. They found that using fixed wireless routers in the network greatly improved the quality of real-time traffic compared to purely ad hoc routing nodes, and was especially important in large ad hoc networks, which are more vulnerable to path breaks. In their experiments, mobile relays experienced a significant drop in packet delivery ratio, as well as an increase in latency due to link breaks, while fixed wireless routers avoided frequent links breaks, caused by node mobility, thus improving packet delivery rates and decreasing delay.

2.1.4 Scale Free Networks

Many real world self-organized networks can be categorized as “scale-free” networks. These networks have the characteristic that a large proportion of nodes being of low degree while a relatively few nodes, like hubs, are of very high degree. A few examples include complex networks in social science where vertices are individuals or organizations and the edges are the social interactions between them [37], or in a large network formed by a nervous system [38] whose vertices are the nerve cells connected by axons, and a huge genetic network whose vertices are proteins and genes, while the chemical interactions between them are the edges [39].

Barabasi and Albert [40] further observed that pages on the World Wide Web also exhibited a similar pattern and showed that in such systems the probability of P(*k*) that a vertex in the network interacts with *k* other vertices decays as a power law, following $P(k) \sim k^{-\gamma}$, emulating the characteristic of scale-free networks that some nodes act as "highly connected hubs" while most nodes are of low degree. Thus, scale-free networks tend to contain centrally located, highly connected "hubs" and exhibit the “small world phenomenon,” in which two average nodes are separated by a very small number of hops, as in airline networks. The quantifiable values in these networks are small average path length (*L*) defined as the average number of hops in the shortest path between two nodes, a high average clustering coefficient (*C*), defined as the average fraction of pairs of neighbors of a node that are also neighbors of each other, and a node degree (*k*) of a node, defined as the number of links connecting that node to the neighboring nodes. In [41] and [42] the authors present an overlay network with fixed routing nodes and a base station as a centralized controller. They use this as an example of applying “small world” patterns to wireless networks in order to achieve a robust, scalable and efficient wireless network. Placement of fixed wireless nodes (called fixed relay nodes or FRNs) then becomes an important issue. Ideally, FRNs should reach a base station in a small amount of hops. At the same time FRNs should be placed according to some criteria such as covering a given geographical area with the fewest number of FRNs. These may be conflicting parameters, so a trade-off between average path length and geographic coverage is needed. As a solution they proposed attaching FRNs to already placed FRNs that have at most *K* neighbors at the time of attachment instead of placing them near highly connected FRNs in the network. They also proposed a routing algorithm based on destination base station and load balancing FRNs which we will discuss in more detail later. The overall theme of the work is that an overlay network with fixed routing nodes in a cellular system exhibits scale-free characteristics and can self-organize itself into a load balanced state.

2.2 Routing in Ad Hoc Networks and Multi-hop Cellular Networks

Routing in ad hoc networks falls into two general categories: proactive routing and reactive routing. Many of the proactive routing protocols stem from conventional link state routing protocols. They maintain routes to all nodes in the network rather than only those that may be presently active. They rely heavily on tables to keep track of route costs or lengths, and are sometimes referred to as table driven algorithms. The advantage of proactive protocols is that latency is low when sending the first packet to a new node since they do not need to construct a new route but only fetch the next hop node from a routing table. This is most beneficial in networks where a large number of nodes communicate with each other for short periods of time. The disadvantages are the relatively large communication overhead and storage cost. Each node has to handle large data structures - routing tables and secondary tables for calculating the routing tables. There is also communication overhead due to the need to maintain routes to all nodes in the network, regardless of whether they are being used or not. Reactive protocols use on-demand routing and routes are only calculated when they are actually needed and only active routes are kept in memory. Their advantage is that they reduce communication overhead, ultimately saving bandwidth. The disadvantage is that when a node needs a new route,

route discovery imposes latency and overhead in initiating the communication. Another disadvantage is that when route requests are flooded, “broadcast storms” may result. Other types of routing protocols developed for large scale routing are hierarchical routing protocols and geographic routing. Hierarchical routing reduces the size of routing tables and route calculation by aggregating remote host information. Geographic routing uses location as the basis for routing.

Proactive routing protocols covered in Section 2.2.1 include Fisheye State Routing (FSR) [43], [44], Optimized Link State Routing (OLSR) [45], [46], Topology Broadcast based on Reverse Path Forwarding (TBRPF) [47], [48]. Reactive protocols covered in Section 2.2.2 include Dynamic Source Routing (DSR) [49] and Ad hoc On Demand Distance Vector Routing (AODV) [50]. The hierarchical protocols summarized in Section 2.2.3 are Hierarchical State Routing [54] and Zone Routing Protocol [55]. The geographic routing protocols covered in Section 2.2.4 are Geographic Perimeter Stateless Routing for Wireless Networks [58] (GPSR), Distance Routing Effect Algorithm for Mobility [59] (DREAM) and Location-Aided Routing [60] (LAR). Finally, recently developed routing protocols related to hybrid and multi-hop cellular topologies are described in Section 2.3.

2.2.1 Proactive Routing Protocols

Fisheye State Routing (FSR) is a simple link state routing protocol which maintains a topology map at each node and propagates link state updates. It is based on Global State Routing (GSR) [51]. In GSR, link state packets are not flooded. Instead nodes maintain a link state table based on up-to-date info received from neighbors and only exchange it with neighbors periodically. A drawback of GSR is that the large size update message consumes a lot of bandwidth and the latency of the link state propagation, which depends on the update period. In FSR, the principle is that entries corresponding to nodes within a certain limited scope are propagated to neighbors with the highest frequency. The entries further away (outside the predefined scope) are propagated at lower frequency. FSR produces accurate distance and path information about the immediate neighborhood of a node, and less timely information and more imprecise information about nodes far away. This imprecision is overcome by the fact that as a packet gets closer to its destination, the route on which it travels becomes more accurate. Theoretical analysis on routing overhead for this type of routing can be found in [52].

The Optimized Link State Routing Protocol [45] (OLSR) is a link-state based routing protocol that collects data about which network members can communicate, and then calculates an optimized routing table. The main idea is to reduce the bandwidth cost of maintaining routes by limiting the broadcast of link-state updates to only certain nodes. To accomplish this, OLSR uses multipoint relays (MPRs) and MPR selectors. Each node selects a subset of its neighbor nodes as MPRs, and these nodes are the only ones that forward packets from this node. In turn each of these MPRs record that the node has selected it as a MPR and calls that node its MPR Selector. Link state packets are exchanged among MPRs for large scale routing. This limits the bandwidth consumption because topology information is only exchanged among MPRs rather than all nodes in the network. OLSR is particularly well-suited to dense networks. When the network is sparse, every neighbor of a node becomes a MPR, reducing OLSR to a purely link state protocol.

The Topology Broadcast based on Reverse Path Forwarding [47] (TBRPF) is also a link state protocol but it uses a very different approach from OLSR. It utilizes a routing tree to forward topology updates. For each node, a minimal hop tree is constructed and topology updates are sent along this tree. This drastically reduces the overhead in the network compared to flooding it with link state packets. However, receiving topology updates along the tree describing the topology has obvious inherent problems when links change. To account for this, the topology tree at each node is recomputed when links change. This has the disadvantage that there will be a stabilization period whenever links change, before routing to the changed section can be resumed. Moreover, TBRPF transmits only the differences between the previous network state and the current network state. Therefore, routing messages are smaller, and can be sent more frequently, helping nodes' routing tables to be more up-to-date. TBRPF is used commercially in "Firetide" brand mesh network routers [53].

2.2.2 Reactive Routing Protocols

Reactive routing protocols are also known as “on demand” routing. Two flavors of on-demand routing that are representative are Dynamic Source Routing Protocol (DSR), and Ad-hoc On Demand Distance Vector (AODV) routing.

The Dynamic Source Routing Protocol [50] (DSR) is a source-routed reactive routing protocol. Each node maintains a cache containing the source routes that they are aware of. They update entries in the route cache as they learn about new routes. The two major phases of the DSR protocol are route discovery and route maintenance. When a source node needs to send a packet to a destination, it looks in its route cache to determine if it already contains a route to the destination. If it finds an unexpired route to the destination, then it uses that route to send the packet. But if the node does not have the route, it then initiates a route discovery process by broadcasting a route request packet (RREQ) containing the address of the source and the destination, and a unique identification number. Each intermediate node checks whether it has the route to the destination. If not, the node appends its address to the route record of the packet and then forwards the packet to its neighbors. To limit the number of route requests propagated, a node processes the route request packet only if it has not already seen the packet and its address is not present in the route record of the packet. A route reply (RREP) is generated when either the destination or an intermediate node with current information about the destination receives the route request packet. A route request packet reaching such a node already contains, in its route record, the sequence of hops taken from the source to this node. On the way back, the RREP is cached by all intermediate nodes it passes on the way. When the RREP reaches back to the source, the source can send the data and caches the new route for future use.

Ad-hoc On Demand Distance Vector Routing (AODV) broadcasts a RREQ packet that is similar to DSR, but instead of recording the route on the intermediate nodes, the intermediate hosts on the way to the destination maintain back-pointers to node from which they received the packet. This sets up a reverse path from destination to source. When the RREQ packet reaches the destination (or a node with a current path to it), the destination sends a route reply (RREP) packet back along the reverse path that validates the route and sets up a forward path. All the nodes that have back-pointers but don’t receive a RREP packet within a specified time, will delete the pointer so that the result is that there is only one route between source and destination.

This is a major difference compared to DSR, since the algorithm first builds up a tree of possible paths with the route information stored at the nodes and then only validates one of the routes. This works better in larger networks since route request packets do not have to contain the entire path as in DSR. To ensure the shortest and most current path is selected, the intermediate nodes use the sequence numbers of the request and reply packets.

2.2.3 Hierarchical Routing Protocols

When wireless networks increase to a very large size, “flat” routing schemes become infeasible. In the case of proactive protocols, the quantity of link state messages flooding the network becomes inordinate, the complexity of the calculation of routing tables becomes overwhelming and the size of routing tables becomes unmanageable. For reactive protocols, the route discovery process becomes prohibitively costly and cached information about far away nodes becomes invalid very rapidly. One way to solve the problem and construct scalable solutions is through hierarchical routing, as has been used successfully for routing on the worldwide Internet. The principle is based on grouping or aggregating distant nodes in the network in some way to reduce shortest path calculation complexity so as to reduce routing table size. One way to do this in an ad-hoc network is to group nodes into clusters that are in relatively close proximity to one another. Each cluster of nodes has a cluster-head. Other logical hierarchies can be constructed. In this section we briefly summarize Hierarchical State Routing [54] and Zone Routing Protocol [55].

In Hierarchical State Routing (HSR), link state routing is the basis. The characteristic features of Hierarchical State Routing (HSR) are multilevel clustering and logical partitioning of nodes. A logical hierarchical topology is maintained by using a clustering scheme recursively. Nodes at the same logical level are grouped into clusters and nodes elected cluster-heads at the lower level become members of the next higher level. These new members in turn organize themselves into clusters, and so on. At the lowest level of clustering, each node monitors the link state of each neighbor and broadcasts it to the cluster. The cluster-head summarizes the link state information within the cluster and propagates it to the neighboring cluster-heads. The knowledge of connectivity between neighbor cluster-heads leads to the formation of the next level clusters. Each cluster has three types of nodes: a cluster-head node which acts as a local coordinator for each node, gateway nodes which are nodes that lie in two different clusters and internal nodes. With this organization, hierarchical routing can be achieved using a hierarchical addressing scheme. The advantages include the ability to create logical/hierarchical addresses for routing and reducing routing table size for most nodes. The drawbacks include the overhead caused by the exchange of routing packets, and the higher average number of hops that the packets take.

The Zone Routing Protocol [55] (ZRP) is a hybrid protocol that uses a proactive routing approach for local nodes and a reactive routing approach for distant nodes. Each node has a pre-defined zone centered at itself, defined by a radius of a number of hops. For intra-zone routing, a conventional link state route approach is used. For inter-zone routing, ZRP uses a route discovery method. A RREQ is broadcast via nodes on the border of a zone, using the "bordercast resolution protocol" (BRP). The route queries are broadcast from one node's border nodes to another node's border nodes until one node knows the exact path to the destination node.

This hybrid approach limits the proactive overhead to only the size of the zone and the reactive search overhead to only selected border nodes.

2.2.4 Geographic Routing Protocols

The use of geographic data for scalable routing in large scale computer networks was discussed by Gregory Finn in 1987 [56] as a mechanism for dealing with Metropolitan networks by using latitude and longitude as components of a hosts network address. Geographic routing is appropriate if the physical distance dominates end-to-end packet delay. In contrast to topology-based routing systems, geographic routing schemes forward packets by only using the position of nodes and their relationship to the location of the destination node [57]. Packets are forwarded greedily, that is to a neighbor nearer to the destination than the current node, until no such node exists or the packet has reached its destination. If no closer node exists, then it is assumed that there is an obstacle and the current node floods the network the packet to all of its neighbors until a node with a neighbor closer to the destination is reached and greedy forwarding can be resumed. In another example, Geographic Perimeter Stateless Routing for Wireless Networks [58] (GPSR) forwards packets in "greedy mode" by default, selecting the next hop based on a locally optimal greedy choice of the neighbor geographically closest to the destination. When there is not a locally optimal next hop based on location, GPSR enters "perimeter mode" using the "right hand rule" where the next edge traversed is the next one sequentially counterclockwise about the line between the current node and the destination, enabling the protocol to route around "holes" in the topology.

Other geographic routing protocols that have received attention recently are Distance Routing Effect Algorithm for Mobility [59] (DREAM) and Location-Aided Routing [60] (LAR). In DREAM, a source node uses a location table to construct a circle centered on the last known location of the destination. The radius of the circle is determined by the velocity of the destination node. The source defines a "forwarding zone" as the area enclosed by the angle whose vertex is the source node and whose sides are tangent to the circle containing the last known location of the destination. The source node then sends its packets to all neighbors in the forwarding zone. Each of these neighbors computes a new forwarding zone based on their tables and then forwards the packets similarly. Thus, packets are flooded in a "forwarding zone" rather than the entire network. LAR uses the same concept of a circle around the last known location of the destination node and floods a destination zone with route request packets. Instead of using a triangular region to forward, LAR's zone is a rectangle. Moreover, in LAR if a node that receives a route request packet is closer to the destination than the neighbor that forwarded the route request packet it will forward the packet. Otherwise, the node drops the route request packet.

In chapter 3 we describe a radial geometric approach to routing in LSSON which uses some principles of geographic routing.

2.3 Routing Protocols for Hybrid Networks and Multi-hop Cellular

A Unified Cellular and Ad-hoc Network Architecture [26] (UCAN) uses conventional cellular routing except when there is a low data rate between a base station and mobile host.

When a mobile client that is actively receiving packets from the base station experiences low downlink channel rate, it sends out a route request message using the IEEE 802.11b interface. This route request is propagated through several intermediate clients until it reaches a mobile client with high downlink channel rate. The proxy client then sends a proxy application message to the base station and serves as a proxy to forward packets to and from the destination. The PARCelS [27] system also only uses multi-hop forwarding when there is congestion occurring within a cell. At this, a mobile host starts to look for relay routes to other cells that are less congested by broadcasting route discovery messages (RDM). When a RDM reaches a mobile host in a cell with more free channels, a route trace-back message (RTM) is generated and sent back to the searching host.

In Cellular Aided Mobile Ad Hoc Network (CAMA) [28], there are "CAMA agents" which may work as centralized positioning information servers. When a mobile needs to send data to a destination, it will send a routing request to the CAMA agent. The CAMA agent responds with a complete route including every intermediate mobile. The algorithm the CAMA agent uses to generate this route is a heuristic called multi-selection greedy positioning routing (MSGPR). The CAMA agent keeps a position information table. This table includes the position of mobile with the route request, the IDs of neighboring mobiles within the sender's radio coverage and their positions, the distance from the mobile to its neighboring mobiles and a delta, which is the change of distance between the mobile and its neighboring mobiles based on the last two position updates. For a mobile, *n* of its neighboring mobiles which are closest to the destination are chosen as next possible hops. These *n* mobiles form an original searching set.

For each of these, n of its neighbors which are closest to the destination are found. From these up to $n \times n$ possibilities, the best n are chosen and, again, n of their neighbors are chosen which are closest to the destination. The procedure is repeated until the destination is included and a maximum of n different routes can be found. The final selection of the route is based on end-to-end packet delay. The results of this kind of routing approach showed large improvements over AODV and DSR according to [28].

The Hierarchical Cellular Multi-hop Network [29], [61] proposed that routing in the multi-hop network supported by an area-wide cellular overlay network be done by Cellular Based Multi-hop (CBM) routing. Here requests are sent to the base station of the overlaying cellular network. This central entity determines the route and responds with a packet comprising a series of mobile nodes willing to relay the data traffic between the source and destination. Service and route discovery is performed in the overlay cellular network and packet data transmissions in the multi-hop network. This exploits both the ability of the macro network to communicate with all of the nodes and the throughput of multi-hop transmission. Results in [29] have shown that CBM leads to low packet drops that might happen due to wrong route information and adds little overhead to the traffic network. Moreover it allows fast packet delivery because of quick route establishment, and the routing overhead increases only linearly with the number of nodes, which indicates that CBM scales well with network size. If Dijkstra's algorithm is used, it takes at least O(n) time, where n is the mean number of nodes in a cell. Since the base station has to service O(n) requests per unit time, the computational burden at the base station is approximately O(n^2) [62]. Routing using multi-hop cellular networks exploits highly reliable base stations and scales much better to many nodes than ordinary ad hoc routing protocols.

In [32] the authors compared decentralized routing algorithms in ad hoc networks with routing in centralized multi-hop cellular networks. AODV and DSR are compared to routing schemes that are supported by a cellular network. The cellular base station gets updated topology information from mobiles when they send their neighbor information to the base station. The centralized routing algorithm used is one in which route discovery packets are sent to a base station and, if the destination is available, the base station sends a route reply packet containing the route information. They found that the overhead related to AODV and DSR is significantly higher than the overhead for the centralized routing scheme in a cellular network, particularly for larger networks. In AODV and DSR the number of broadcast packets increases as the square of path length, indicting that the network is vulnerable to broadcast storms when the network gets too large. In contrast, the centralized cellular routing method has an overhead proportional to path length, indicting a superior scalability for multi-hop cellular networks.

In a multi-hop cellular system with fixed routing nodes, several routing strategies have been considered as in [41]. As an example, when a mobile wants to establish a new connection, it sends a request to a nearby node (fixed node or base station) if it is covered by one. If a node is found, a possibly multi-hop route is formed to the base station via the fixed routing nodes based on some shortest path algorithm. While this provides a path to the base station, it does not enable load balancing. Three other approaches that support load balancing approaches were compared to this non-load balancing algorithm. First, when a mobile wants to make a connection, it attaches to a base station if it is not congested. If it is congested, either the mobile is put into a queue or an alternative route to another base station is found through the fixed routing nodes. If the mobile is not covered by any base stations but is covered by a fixed routing node, a route to a base station is formed via the fixed routing nodes. The choice of base station depends on the load at the base station and all packets follow along the same route to a given base station. In a second approach, when a mobile requests service, it connects to a base station directly or via the fixed routing nodes, depending on its location. The destination base station is decided upon while initiating the connection and does not change for the duration of the connection. However, packets may travel through different intermediate fixed routing nodes depending on the load, attempting to perform load balancing among the fixed routing nodes. In a third approach, the least loaded destination base station is chosen initially but can be changed during the connection. Once a packet reaches a fixed routing node, the next hop is decided by taking into account the load of the neighboring fixed routing nodes and the number of hops between the fixed routing nodes and all of the base stations in the geographical area. Based on simulation it was found that the third routing strategy produced the fewest average number of hops per transmission. The ability to change the destination base station avoided a situation where alternative routes to a base station became very long.

In the Integrated Multi-hop Cellular Data Network [63] (IMCDN), rooted spanning trees are used for the routing topology. The fixed routing nodes maintain a spanning tree for every base station in the system. The spanning tree is essentially the routing topology within a layer and is constructed and maintained based on an algorithm developed in [65]. Each fixed relay node (called "routing nodes") maintains a simple routing table for each base station containing the node's parent id, its depth in hops from the base station and the load associated with the tree. Each node in the system is assigned a home base station based on the cell within which the node resides. It is typically the "closest" base station in terms of hop count or the base station from which it receives the strongest signal. Downstream messages are delivered via broadcast message from the home base station. When a mobile has a message to send, it is sent to a nearby routing node which chooses a tree according to a load-balancing or minimum delay criteria [64]. The routing node then relays the message to its parent node in the selected spanning tree until it arrives at the base station. Here, it is assumed that all base stations are connected to a wired infrastructure so that large scale routing can be accomplished by simply getting a message to any base station in the system.

The IMCDN scheme's algorithms enable a self-organizing network to offer a cellular system upstream multi-hop access while significantly deepening its coverage, lowering power consumption of mobiles, and increasing network capacity. It also offers very simple routing tables, enabling low-power, and low-complexity routing nodes. IMCDN works well when there are not a large number of base stations in the system and when all base stations are assumed to be "wired" to all other base stations through a high-speed back-haul network. However, when there are a very large number of base stations in the system, the number of spanning trees that each node must keep track of may become unmanageable. Moreover, if the base stations are not connected to a high bandwidth backhaul network, large scale routing is no longer a trivial matter of just forwarding a message to the nearest base station. In this work, we use the IMCDN as a basis and improve upon it by limiting the number of routing trees each wireless routing node maintains to the number of cells in a cell cluster, a familiar concept in cellular networks.

Chapter 3

Large Scale Self-Organizing Network (LSSON)

3.1 Overview

In this chapter, we present a network architecture and routing methodology for a dense multi-hop cellular network that scales well to a very large number of nodes. A hierarchical proactive routing scheme is employed that involves two-tiers: an upper tier for large scale cell-level paths and a lower tier for local tree-based routing. The upper-tier routing is accomplished by tracking a next hop cell for L representative routers or "virtual routers" that represent sectors emanating from a source root thereby forwarding the message to the parent node associated with the next cell's routing tree. Lower-tier routing involves simply routing to the parent node of the tree with respect to a destination base station. Once there, the message is broadcast to the destination user.

Routing complexity is reduced from $O(n^3)$ to O(1) for this routing scheme at a small cost in efficiency. Simulations are performed on different sized networks ranging from 70,000 nodes to nearly 700,000 nodes to demonstrate successful performance of the large-scale routing approach.

3.1.1 Introduction

We present the design and analysis of a novel hierarchical large scale wireless architecture and associated protocols for networks that may scale up to hundreds of thousands of nodes. The architecture is self-healing, self-organizing, and enables load balancing. Complexity for large scale routing is reduced dramatically, enabling routing nodes in the network to be simple, inexpensive and easy to deploy.

Mesh network routing models that have been proposed and analyzed in recent years such as those in [66] perform well under many conditions but can become problematic when network sizes become very large. Structured peer-to-peer overlay networks for large scale routing are highly scalable systems that utilize distributed hash tables (DHT) enable nodes to maintain small routing tables [67]. While producing reasonable bandwidth usage and reasonable accuracy, they add latencies in the usage of overlay routing, which differs from the underlying network routing. Ant Algorithms [68] have the capability to find a shortest path between source and destination but with many thousands of nodes, the number of ant agents needed to produce accurate results becomes prohibitively large. Numerous sensor network algorithms, such Rumor Routing [69], have been proposed. They typically save energy through caching and use of high-level descriptors or *meta-data*, and often don't need to maintain the whole network topology. However, there may be significant inefficiencies if the number of events in the network is large, there may be difficulties in tuning time-to-live for queries and agents to prevent overhead, and on-demand routing produces additional overhead.

In our Large Scale Self Organizing Network (LSSON), we use proactive routing where next hop information is kept for representative routers in directional sectors. We require geographic positioning of key nodes called "roots," which will also be referred to as base stations in this thesis. Our solution incorporates a hybrid network model that integrates the benefits of both peer-to-peer multi-hop and cellular networks. Substantial work including those mentioned in [70,71] demonstrate that peer to peer routing in hybrid networks extends the reach of network, lowers per-node power requirements, and increases the raw capacity of a network. At the same time, base stations better support large scale routing without involving mobile users in the routing process. Fixed routing nodes in these systems are placed at strategic locations, relieving mobile nodes from complex routing functions and enabling load balancing by diverting traffic from a more congested cell to another less congested cell [72]. They can also improve performance in high traffic locations as well as "shadowed" areas of the network and better facilitate real-time multi-media services.

3.2 LSSON Architecture

Our physical topology is a Multi-hop Cellular Network model with fixed relay nodes. Mobiles communicate upstream through the fixed relay nodes to a base station. Mobiles register with their local base station as is typical with cellular systems. The mobility management aspect is not addressed in this thesis but we assume that the system is so equipped such that mobiles can find their ever-changing location-dependent network address from base station broadcasts and can acquire the address of others they need to communicate with through a proper signaling protocol. We do not assume that roots in the system are "wired" through a high-speed backhaul network. Therefore, large scale routing is not a matter of simply forwarding a message to the nearest base station and as such routing complexity becomes a major challenge.

Messages delivered downstream to mobiles are achieved by one-hop direct broadcast from high power base stations, or optionally in multiple hops downstream to nodes that are out of broadcast range via source routing. The fixed relay nodes are self-organized into a multiple-tree hierarchical routing network similar to the Integrated Multi-hop Cellular Data Network presented in [9]. Dynamic routing trees are maintained and rooted at each base station and nodes maintain their parents for each tree. The trees are limited to cell clusters as will be described in Section 3.3. Data messages are sent along a sequence of routing trees as selected by the upper-tier routing algorithm until they finally arrive at the destination's base station.

A cell is defined as the area where signal power from a base station is strong enough to deliver data messages to mobile users. The cell of a base station is overlapped with those of its neighbors. Thus a node may simultaneously be in multiple cells depending on its location and propagation characteristics of base stations. As in cellular systems, cells are grouped into clusters of adjacent cells, taking advantage of the frequency re-use concept. A group of *N* cells that make up a cell cluster have coverage ranges that overlap so that a cluster is formed from a union of a cell and its adjacent cells. In our scheme, a routing tree is rooted at a base station and limited to the cell cluster associated with this base station.

3.3 Overview of Routing

Intra-cluster, or "local" routing is a simple process of a message proceeding to the next hop along a routing tree to a root whose cell includes the destination. This is shown in Figure 3.1. Intra-cluster routing occurs when the source and destination are within the same cell cluster.

We define large scale routing as routing that involves communication between two nodes that are not in the same cell cluster. When inter-cluster routing is needed, packets are put on the routing tree for the next hop cell towards the destination. Figure 3.2 demonstrates how this is accomplished by an example.

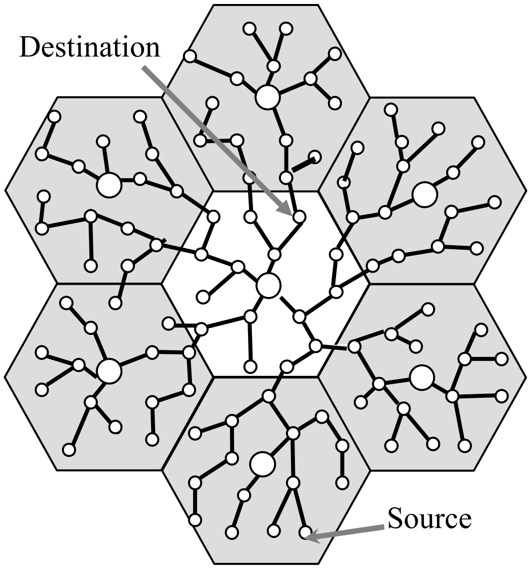


Figure 3.1: Intra-cluster routing example

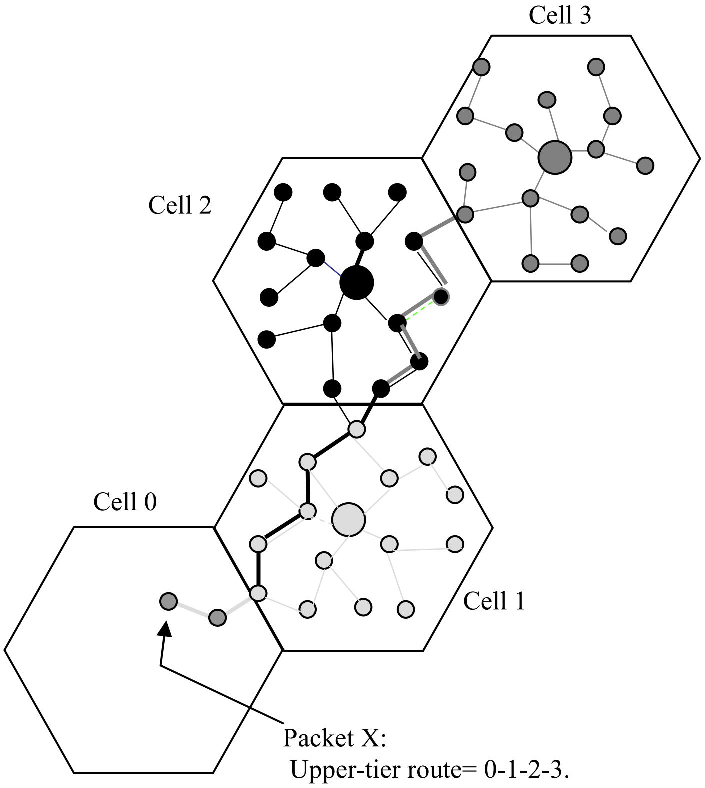


Figure 3.2: Inter-cluster routing example

As shown in Figure 3.2, a packet in Cell 0 needs to be delivered to a node in Cell 3. According to the root of Cell 0, the next hop cell to Cell 3 is Cell 1. This information is known to all the nodes in cell 0 through broadcast by the root of the cell. The packet is part of Cell 1's routing tree while in Cell 0 and then when it crosses into Cell 1 becomes part of Cell 2's tree, which is the next hop cell with respect to the destination cell. When the packet crosses into Cell 2, it becomes part of Cell 3's tree. From this point only intra-cluster routing is involved. The packet is forwarded up the routing tree until it arrives at the root of Cell 3. From there, it is broadcast to the destination node in Cell 3.

3.3.1 Initial set up of cluster routing trees

As mentioned previously, a cell is defined by the coverage area of a base station. Moreover, in cellular systems, clusters are assigned according a frequency re-use scheme. To begin the creation of a cluster-based spanning tree, a base station sends out a beacon signal on a control frequency, identifying itself to nodes within its coverage range along with other information. This message is of the same format as the "X" message specified by Hester and Lee [8,9] and shown in Figure 3.4a. Since each base station is the root of a routing tree for a cell cluster made up of its own cell and neighboring cells, each base station keeps the TREE_IDs of the neighboring cells that are part of its cluster. The root broadcasts in its beacon signal this set of TREE_IDs that make up its cluster, specifying the routing trees that all nodes within its range must join. Accompanying these TREE_IDs are values for the depth of the TREE_ID. Specifically, the root broadcasts the value of its own TREE_ID and indicates its tree depth as 0 (the root of the tree). All nodes that receive this message from the base station enter the TREE_ID of the base station they are within range of in their routing table. Nodes also respond to the root broadcast by identifying themselves. A subset of all nodes in the cell have their packets received by the root. The nodes that are not close enough to the base station to have their message received or do not have sufficient signal power to send messages to it will not receive a response from the base station and the tree creation process times out after a short period of time. For nodes that can contact the base station directly, the base station then sends an acknowledgement to each responding node, verifying upstream connectivity. Upon receipt of this message, these nodes include the base station node as a TREE_ID and set their depth to 1, indicating that the base station is the parent node. Then these nodes advertise their presence indicating that they are 1-hop depth from a particular base station, or TREE_ID. When another node receives such a message, it compares its routing table TREE_ID entries to the TREE_ID being advertised. If the depth in the node's table is less than or equal to the advertised depth (in this case 1), then the message is ignored. Otherwise, the nodes respond, identifying themselves. Responses received by the 1-hop depth nodes are acknowledged and these nodes set their depth to 2 for that particular TREE_ID and enter the advertising node as their parent node. This process continues successively until all nodes within a cell have joined the routing tree associated with the root of the cell as specified in the pseudo code of Figure 3.3.

```
BS broadcasts beacon signal identifying itself with TREE_ID and depth=0.
Nodes within range receive signal, enter TREE_ID in routing table with depth = ∞
Do until all nodes TREE_ID have <>  ∞
    If depth of node with TREE_ID is > broadcast depth +1
    Then
        After random delay node responds with node_id
        Broadcast node ACKs
        If ACK received
            Depth = Broadcast depth + 1
            Node broadcasts node_id, TREE_ID and depth
        Else
            No Update
    Else do not respond to broadcast
Enddo
```

Figure 3.3: Pseudo code for initially setting up cluster-based spanning trees

3.3.2 Identification of Cell Clusters

A cell cluster is a set of cells assigned by a BS. It contains cells that are adjacent to and have overlapping range with that BS. The cluster associated with a BS is determined and maintained by the BS itself. Each node within range of a BS periodically sends a list of BSs they are within range of to each root along the appropriate tree if it exists. Based on this information from all of the nodes within a cell, the BS determines the neighboring BSs that have overlap signal range with itself. The BS then broadcasts this list of BSs that make up its cell cluster. The nodes within the cell then only join routing trees related to those BSs. This limits the size of the routing trees to the size of clusters only.

Each node in the system has one unique home base station. Although a node may actually be in signal range of multiple base stations, in order to simplify routing, each node selects only one home base station. This selection may be done on a criterion such as the base station from which the node has the smallest number of hops.

Periodically each base station in the system broadcasts a control signal on a control frequency, identifying itself and the other cells that belong to its cluster. The routing tree for each base station is constructed and continually updated as new nodes are added and/or existing nodes are dropped. A pre-existing routing tree of *n* nodes expands to become a routing tree of *n* + 1 nodes, whenever a new node is added to it. When a new node is added, it is placed within the range of at least one other active pre-existing node in the network. Thus, the range of any network node must be non-empty. A new node must be within range of at least one active node that belongs to one or more existing routing trees.

3.4 Building and Maintaining Routing Trees

Initially a new node will communicate on a control channel. In this network, all nodes are capable of receiving and transmitting on two channels, a control channel and a data channel. A node joins a physical network through an "X-Y-Z" 3-way handshake process which takes place over the control channel. When a node enters the space of an existing network with *n* nodes, the new node N_{n+1} broadcasts an "X Message" to all nodes in its range signaling the network of its arrival. The basic frame structure is the same as that as specified by Hester and Lee [8,9] and shown below except that the field named of the "LAYER_ID" in [8,9] is changed to TREE_ID:

X Message

Msg Type	SRC_ID	TREE_ID	Depth	Parent ID	LOAD PARAMETER	TTL

Y Message

Msg Type	SRC_ID	TREE_ID	Depth	Parent ID	LOAD PARAMETER	TTL

Z Message

Msg Type	SRC_ID	TREE_ID	Depth	LOAD PARAMETER

Figure 3.4a,b,c: Message formats for "X," "Y," and "Z" messages

All of these frames have a multiple row structure with entries for TREE_ID in the "X-Y-Z" messages, enabling the tracking of topological information associated with each routing tree and allowing for a row corresponding to each base station that is in the cluster of a node's home base station. Information that a node keeps is maintained in three tables: a range table which identifies the other nodes within its range and a routing table that keeps information about routing trees that the node is a part of it, and an address table (to be discussed in Section 3.6.1.4) which is used for large scale routing.

When a new node enters the network in initialization mode, it goes through the X-Y-Z control message exchange process and establishes its range $R_{i,}$ the set of nodes with which it can directly communicate. In a multiple routing tree network, depth-1 nodes within a given Tree *k*, are nodes within a one hop distance of the root of Tree *k*. The depth-2 nodes within Tree *k* are nodes within the range of the depth-1 nodes of Tree *k*, but outside the two-way communication range of the root of Tree *k*. In general, depth-m nodes within a given tree *k* are nodes within range of depth-(m - 1) nodes of tree *k*, but outside the two-way communication range of the root, the depth-1 nodes, the depth-2 nodes, …, and the depth-(m - 2) nodes within Tree *k*. Based on the information contained in the Y messages received from the members of R_i, Node *i* determines the neighboring node with the smallest depth, $m_i^{(k)}$ for tree *k*, within R_i and selects it as its parent within Tree *k*, i.e. $D_i^{(k)}$ (Parent node in tree *k*) = $m_i^{(k)}$. Then the node *i* sets its depth within the Tree *k* to be one plus the depth of its parent $P_i^{(k)}$ within the tree, $D_i^{(k)}$ (Node *i* in Tree *k*) = $m_i^{(k)} + 1$. The node only goes through this algorithm for the trees specified by the BS(s) as being part of the cell cluster that the node belongs to. Finally it broadcasts on the control channel its first W Message, containing its depth information and the load information of each available tree. Since each Y and W message contains the depth and load parameters of the sender in every tree, every node will be able to maintain its range table and routing table upon the receipt of such messages.

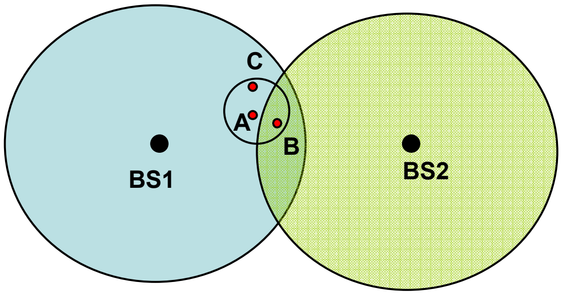


Figure 3.5: A new node entering the network within range of BS1

For example, in the scenario depicted in Figure 3.5 above, a new node, *A*, is able to receive control signals from the base station, BS1. After receiving a control signal from BS1, *A* begins to populate its "X" message with the TREE_IDs specified by BS1 as the roots that belong to its cluster. Two nodes within range of Node *A* are Nodes *B* and *C*. Node *C* is also only within the range of BS1 while Node *B* is within the range of two base stations, BS1 and BS2. Nodes *B* and *C* each respond with a "Y" message that has the same format of the "X" message that Node *A* sent. This enables Nodes *B* and *C* to thereby inform *A* that that they are within its range. If Node *A* receives the "Y" messages from Nodes *B* and *C*, it will add them to its range set while replying with a "Z" message that confirms the fact that they are "mutually" within range. Upon receipt of the "Z" message, *B* and *C* add *A* to their range set and perform certain maintenance tasks to be described later. Since Node *C* clearly resides in the same cell as Node *A* (within range of BS1) it will send information to *A* about the exact same trees that Node *A* keeps in its routing table. Since Node *C* is within receiving range of both BS1 and BS2, it may choose BS2 rather than BS1 as its home root. This selection may be done by a criterion such as signal power from the root, but the rule remains that any node can only belong to one home root. If node *B* "belongs" to the root BS2, it will track some information about trees different from the trees that *A* tracks, and thus contain some different TREE_IDs in its Y message. Upon receipt of the Y message from *C*, Node *A* will ignore TREE_IDs not specified by BS1.

Base stations maintain cluster information using periodic node-to-base station control messages. Since a cell cluster is made up of cells that overlap with each other, and in order to minimize control messages in the network, only those nodes that are within range of more than one BS send control messages to their home roots. These control messages contain the IDs of the BSs that they are within range of. These are sent along the appropriate routing trees to the respective roots. The receiving base station collects these messages and accordingly defines a cluster as the collection of BS IDs received in the control messages. The BS then broadcasts this information to all the nodes within its range. For example, in Figure 3.5, Node *B* will receive two different sets of cluster information, one from BS1 and another from BS2. Node *B* will track in its routing table only entries for root its home root.

If a node is not within signal range of a base station it can still be part of the network. In this case, it does not receive a beacon signal of a base station which limits the trees that the node can join. Then the node is free to join all trees of the neighboring nodes in the network. This enables the network to extend its reach into areas that may not be penetrated by a base station signal.

3.5 Routing Table Information

Each node maintains a routing table based on information it receives from its neighbors and from the BS(s). The format of the routing table is shown in Table 3.1.

k	The Tree ID: only those specified by home BS(s)
$D_\mathrm{i}^{(\mathrm{k})}$	The depth of node *i* in tree *k* (i.e., tree rooted at BS k)
$P_\mathrm{i}^{(\mathrm{k})}$	The parent node in layer *k* within R_i whose depth achieves min $D_\mathrm{j}^{(\mathrm{k})}$ in tree *k*
$\rho_\mathrm{i}^{(\mathrm{k})}$	Load information of the nodes in the path from node i to the root of tree *k*

Table 3.1: Routing table of node *i*.

A routing table entry contains information for every routing tree to which the node should belong, typically specified by the home BS. For each tree id (i.e., BS id), its parent id, the depth in the tree (in number of hops), and its load within that tree is tracked. The space complexity of this routing table is O(*T*), with *T* being the total number of routing trees. The complexity does not grow with the network size in terms of number of nodes. Rather, it grows linearly with the number of trees which is limited by the cluster size of its home base station. The tree id specified in the message depends on the method of inter-cluster routing.

3.6 Reducing the Complexity of Large Scale Routing

A large wireless self-organizing network may contain a huge number of roots, i.e., a huge number of cells. A distance vector routing algorithm used at the root level among all of the roots may then be too complex to be practical. As a solution to this problem, we have examined a hierarchical destination address aggregation scheme where the degree of aggregation is proportional to the geographical (Euclidean or L1) distance between the current location and the destination. As a further improvement, we explored a radial geometric approach. The objective is to drastically reduce the size of root-level (upper-tier) routing table and the associated communication and computational complexities involved such that they are independent of the density and the dimension of the network. This entails a hierarchical network partitioning and addressing scheme and a novel multiple-tier routing strategy.

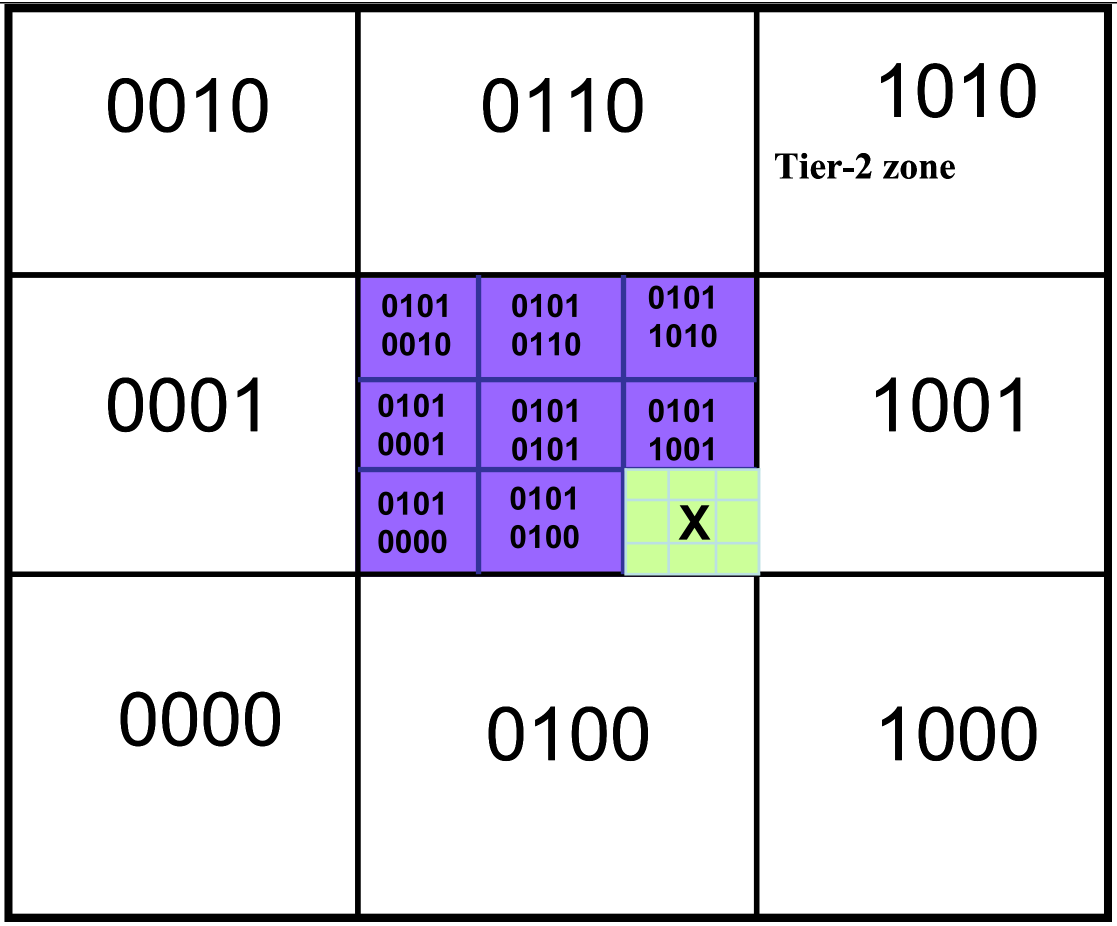
3.6.1 Zone Routing: Hierarchical Partitioning and Addressing

To reduce inter-cluster routing complexity, a destination address aggregation approach can be used whereby the network is partitioned into M tiers of K partitions each. The basic idea is to aggregate remote destinations by geographically partitioning the network into K equal-size tier-M zones each of which is similarly sub-partitioned into K equal-size Tier-(M - 1) zones, …, and finally each Tier-1 zone is sub-partitioned into K Tier-0 zones. Figure 3.6 depicts such a hierarchical partition with K=9, where a Tier-0 zone is smaller than the size of a cell - the broadcast range of a root - such that a Tier-0 zone will never contain more than one root.

As a packet traverses the network, it progresses from one cell cluster to the next. Before it reaches the cell cluster in which its destination is situated, it follows a dynamic “inter-cluster route" where its destination is absorbed by a "super node" - a root representing a high-tier zone whose address matches the prefix of the destination address. As soon as the packet reaches the cell cluster of its destination, it switches to "intra-cluster route" thus following the intra-cluster routing table en route to its destination. All nodes in a cell use a common inter-cluster routing table; one example is shown in Figure 3.6 where the degree of destination address aggregation is reflected in the address prefix length. We note that the "next root" indicated in the table is readily translated to "next hop" using the intra-cluster routing table because the "next root" is within the cell cluster of the node with which this inter-cluster routing table is associated. The intra-cluster routing table contains the parent node along the upstream routing tree path to the root whose range covers the destination, and then once the packet gets to the root of the destination cell, it is broadcast directly to the destination node. Therefore, while all nodes in a cell share the same inter-cluster routing table and thus the same next root for a given remote destination, they may have different next hop nodes to that root. As a result, routing complexity is drastically reduced due to the possible destinations that a single node's routing table has to deal with.

It is important to point out that the hierarchical zone structure is fixed and is independent of the varying wireless cell and cell cluster boundaries. The inter-cluster entries essentially capture root-level shortest distances and rely on aggregated addresses. Therefore, they are sub-optimal. They indicate the "next root" from the present location to the next destination zone. Each zone listed in the routing table must contain a representative root.

Figure 3.6: A geographically based hierarchical network partitioning and addressing scheme. A 3-tiered zone hierarchy is shown. The addresses of the tier-0 zones in the 0101.1000 tier-2 zone are 0101.1000.0000, 0101.1000.0001, 1010.1000.0010, 0101.1000.0100, 1010.1000.0101, 1010.1000.0110, 1010.1000.1000, 1010.1000,1001, and 1010.1000.1010.



3.6.1.1 Partitioning Zones

There are several possible approaches to graph partitioning. Developing an algorithm to partition a graph into "balanced" partitions, where there are even number of nodes in each partition, is known to be an NP-hard problem, and the only way to cope with it is to use heuristics to come as close as possible to the optimal solution [73]. In our partitioning approach, we make no effort to balance the partitions, but utilize very straightforward tools to accomplish our purpose of root aggregation. Quad-trees are a simple partitioning structure, and part of a family of hierarchical spatial data structures. One example of this structure is partitioning a two dimensional space by recursively subdividing it into four quadrants [74]. A more general definition of such shapes is "region quad-trees" which are refinements of a shape into smaller and smaller instances of that shape [75]. The degree of a quad-tree is the number of edges formed from one edge when a shape is subdivided into smaller elements. A square that is partitioned into a three by three arrangement of squares is a degree three square quad-tree. The level of the root of a quad-tree is zero and the level of a child is one more than its parent.

We have used quad-trees in a partitioning heuristic, where $\sqrt{K}$ is the degree of the quad-tree and the number of levels is equivalent to the number M tiers. In our terminology, we define the tier number in a way different from standard quad-tree terminology: instead of tier 0 being the largest sized element, we define tier 0 to be at the smallest sized element. Thus, tier 0 partitions are those of smallest size, and tier M partitions are those of largest size. Figure 3.6 shows a degree 3 quad-tree with 3 levels. We define this as a system of 3 tiers with 3 x 3 = 9 partitions each.

3.6.1.2 Addressing Zones

Addressing of two-dimensional zones in a tiered hierarchical structure is similar to internet protocol (IP) addressing where the address hierarchy is reflected in the grouping of bits. In an *M*-tiered hierarchy with *K* partitions per tier, there will be *M* segments of bits. The number of bits, *k*, in each segment must satisfy the inequality $2^k \geq K$. There are two components to each tiered address segment: the x coordinate and the y-coordinate. Suppose *k* is an even number. Then the first group of *k*/2 bits is the x-coordinate and the second group of *k*/2 bits is the y-coordinate. As an example, in Figure 3.6 there are K = 9 partitions per zone. The amount of bits necessary to represent the tier address in each segment is 5. In the figure, the partition in the upper right hand corner is a tier-2 zone. It has an address of 1010 in binary, which corresponds to the zone 2,2. The high level address of the center tier-2 zone is 0101 in binary, which corresponds to zone 1,1. This and every other tier-2 zone is made up of *K* tier-1 zones. In this case *K*=9. These zones have their own tier-1 address within tier-2. So the address of the tier-1 zone in the upper right hand corner of the middle tier-2 zone is 0101.1010. In the same way within each tier-1 zone there are *K* (in this case *K*=9) tier-0 zones. These are shown in the lower right corner of the figure. These tier-0 zones have three components to their address. The tier-0 zone with an "X" in the Figure 3.6 corresponds to the tier-0 address 0101.1000.1010.

3.6.1.3 Selecting Representative Roots

The partitioning of the network into zones creates a group of roots in each partition. Within each partition, a representative root must be selected that can serve as a routing table entry for routing nodes in distant zones for that zone address. There are various approaches to "electing" a representative root, such as a "cluster head," that are potentially useful. The basic idea is to find a center node for each zone that minimizes the maximum of the weighted distances between that point and the other nodes [76]. This issue was first presented and solved by Hakimi in 1964 in his paper about the optimal location of switching centers [77]. More generalized approaches are the centroid, and the Fermat-Weber point. The centroid is found by minimizing the mean or the sum of the squared distances between a vertex and the other vertexes. The Fermat-Weber Point is the point that minimizes the average cost from a center point to other points. A similar problem is the "1-center problem" where the goal is to find a center location such that the maximum distance from the center location is minimized. One can define an L_p estimate as one that minimizes the sum of the p^{th} powers of distances of observations from the estimate. The centroid is the L_2 estimate; the next most commonly used estimate is the L_1 estimate, also known as the Fermat-Weber as defined point above. For the location of our representative roots in each zone we shall consider the L1 estimate.

Using the L_1 distance methodology outlined above, we are able to select a representative root for each tiered zone. As an example, in Figure 3.7a there are six roots within one zone. Since Nodes *A* and *B* are both near the center of the zone it is reasonable to assume that one of them should be considered as the representative of the zone. Using the Fermat-Weber method, Node *A* is considered the center of the zone. If we use L_2 distance, however, our choice would be Node *B* since the mean distance between *B* and each of the other roots is approximately 38 (Figure 3.7b) while the mean distance from *A* to the other 5 nodes is approximately 39 (Figure 3.7c). For the purposes of our large scale routing simulation we chose the L_1 center since it was easier to calculate. In a grid topology, the zone representative is very frequently the root located closest to the geographical center of the zone.

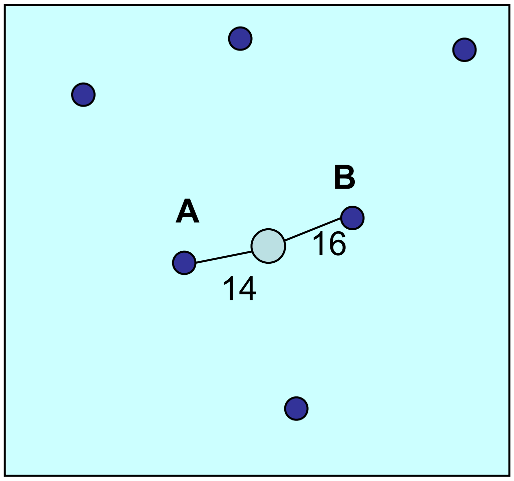


Figure 3.7a: Six roots in a zone partition. Node A is closest to the center of the zone, so it is the Fermat-Weber point or 1-center of the zone.

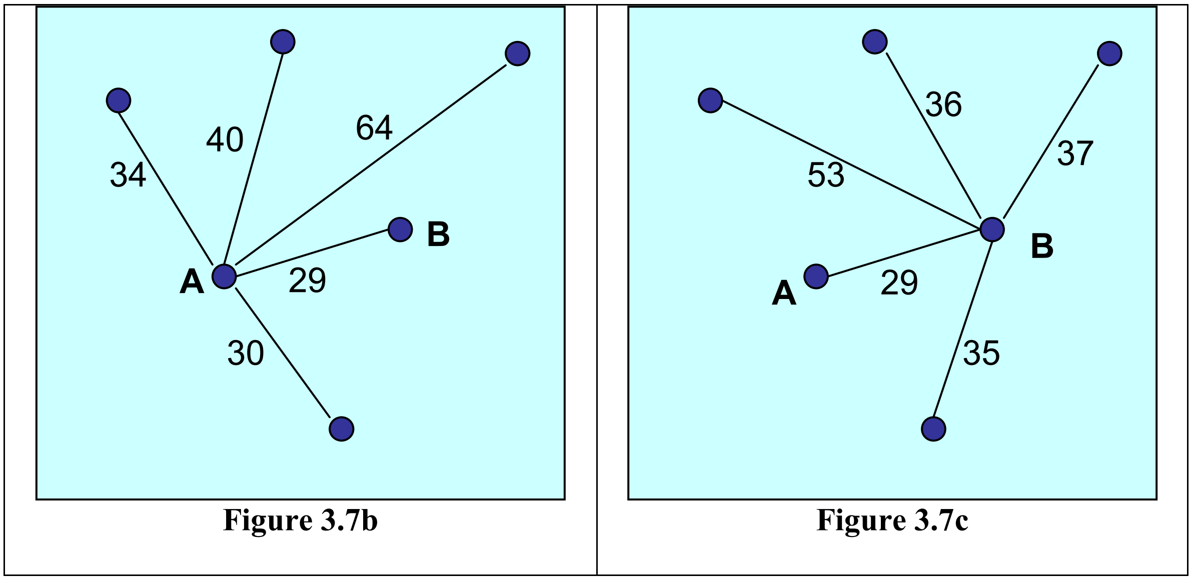


Figure 3.7b

Figure 3.7c

Figure 3.7b,c: The mean distance the 5 other roots are away from B is 38, while the mean distance the other roots are away from A is 39. Therefore B is the centroid for this zone.

Within each tier, a representative root is chosen. Each tier-*M* zone will select its L_1 center as its tier-*M* representative. In the same way, each tier-(*M* – 1) zone will select its L_1 center, each *M* - 2 zone will select its L_1 center and so on.

3.6.1.4 The Address Table

As mentioned previously, intra-cluster routing is done using a node's routing table associated with the tree structure set up within each cell cluster. Large scale routing utilizes a node's address table in conjunction with its routing table. The address table keeps track of distant representative roots for large scale routing as in Table 3.2:

Address	The high level address of a distant node
Tree ID	The ID of the next hop cell (will also be a tree id within cluster)

Table 3.2: Address table

The address table resolves a higher level network address to a next hop cell. This identifies a tree ID for the message to be sent on. The address table contains entries for every highest tier address which is the set of highest order bits in the zone address. In addition to entries for every tier-*M* zone it also contains entries for less distant tier-(*M* - 1) zones: zones that have multiple entries for some tier-*M* zones, since there are *K* tier-(*M* - 1) zones in each tier-*M* zone. These are identified with the same highest order set of bits but unique order *M* - 1 sets of bits. Entries for less distant addresses contain more lower order sets of bits. The distance to each of these zones is the distance to the representative "super-root" in that zone. The address table contains the address or sub-address of the distant zone and a next hop root. After the address look-up occurs for distant nodes, the next hop root is then referenced in the routing table to determine which tree to use in sending the packet on its way. As described in the previous chapter, the routing table contains an entry showing the next hop address for each root in the particular root's cell cluster.

An example of use of the address table and routing table is shown in Table 3.3. Addresses are in hexadecimal with an M=4 and P=16 zoning structure. The address of the root containing this table is 3.5.C.E and it contains entries for *P* - 1 tier-*M* zones, *P* - 1 tier *M* - 1 zones, *P* - 1 tier-1 zones, and *P* - 1 tier 0 zones. If it needs to send a message to a very distant node whose address is A.1.2.3, then it refers to its address table. According to the address table, the entry for the tier-3 Zone *A* has a next hop of root *d*. This value is then used to determine the next hop node. In Table 3.4, the routing table entry for root *d* indicates that the next hop node is 10203.

Tier-M		**Tier M-1**		**Tier-1**		**Tier-0**	
Destination	Next Root	Destination	Next Root	Destination	Next Root	Destination	Next Root
0.x.x.x	a	3.0.x.x	c	3.5.0.x	a	3.5.C.0	c
1.x.x.x	a	3.1.x.x	c	3.5.1.x	c	3.5.C.1	f
2.x.x.x	a	3.2.x.x	d	3.5.2.x	d	3.5.C.2	a
5.x.x.x	b	3.3.x.x	d	3.5.3.x	e	3.5.C.3	e
5.x.x.x	b	3.4.x.x	e	3.5.4.x	d	3.5.C.4	b
6.x.x.x	c	3.6.x.x	f	3.5.5.x	e	3.5.C.5	c
7.x.x.x	c	3.7.x.x	f	3.5.6.x	b	3.5.C.6	a
8.x.x.x	c	3.8.x.x	a	3.5.7.x	b	3.5.C.7	a
9.x.x.x	d	3.9.x.x	a	3.5.8.x	a	3.5.C.8	d
A.x.x.x	d	3.A.x.x	b	3.5.9.x	f	3.5.C.9	f
B.x.x.x	e	3.B.x.x	b	3.5.A.x	f	3.5.C.A	e
C.x.x.x	e	3.C.x.x	c	3.5.B.x	d	3.5.C.B	e
D.x.x.x	e	3.D.x.x	c	3.5.D.x	e	3.5.C.C	d
E.x.x.x	f	3.E.x.x	f	3.5.E.x	a	3.5.C.D	d
F.x.x.x	f	3.4.F.x	d	3.5.F.x	b	3.5.C.F	b

Table 3.3: Address Table for M = 4, P =16.

Tree	Hops	Parent	Load to root
a	5	10230	12
b	4	10299	14
c	5	10302	25
d	6	10203	12
e	3	10576	25
f	5	10205	17

Table 3.4: Routing table for example node.

Every tier-0 root exchanges control packets with every other tier-0 root within a certain range in order to record the best next hop to that root. The range we define is max(dx,dy) ≤ D_0, where dx is the x-coordinate difference between the root and the outer edge of the range and dy is the y-coordinate difference between the root and the outer edge of the range. This essentially creates a square centered at the root that has equal sides of $2D_0$. In a similar way, every tier-1 root exchanges control packets with every other root within a range where max(dx,dy) ≤ D_1, every tier-2 root exchanges control packets within a range max(dx,dy) ≤ D_2, every tier-3 root exchanges control packets within a range max(dx,dy) ≤ D_3, and so on. The values of D_j, j = 0, 1, 2, … are flexible parameters with the constraint that $D_{\text{j-1}} < D_\text{j}$.

3.6.1.5 Routing Complexity

Distance vector routing algorithms used by each routing node maintains distance values representing the best distance from that node to each destination. These values are updated by neighboring nodes exchanging their distance information with each other. A flavor of distance vector routing is the Bellman-Ford routing algorithm. For a system of *n* nodes and *E* edges, the complexity of the Bellman-Ford algorithm is O(n^2E) [78], which is essentially O(n^3). For a very large number of nodes, this is impractical for large scale routing. If there are N roots in the system and we use the cluster trees for intra-cluster routing, complexity can be reduced by limiting distance-vector routing to inter-root or inter-cell routing. If distance vector routing is utilized among all of the roots, then the complexity is O(N^3) where typically, $N << n$ and thus a significant improvement in routing complexity is achieved. However, as the network gets bigger, this may still pose problems since N grows large. The tiered zone model we have described scales much more gracefully. Utilizing this hierarchical model, the number of entries that need to be tracked is $P^2 + (M - 1)P$. This value is approximately MP when M is large. For a very large network this results in significantly fewer control messages, a large reduction in routing table size, and thus a proportional reduction in storage and memory space required of nodes within the system. As a result, the routing complexity is reduced to O($(MP)^3$). With $MP << N << n$, this presents a sizable reduction in complexity.

For a very large scale routing network, the complexity reduction is tremendous. For example, Table 3.5 shows the degree of improvement in a very large scale routing environment with 770,000 nodes and 2,500 base stations and a 4-tiered partitioning hierarchy. Such a dense multi-hop networking environment will be used in our simulation study to be presented later in this chapter. In a less dense multi-hop networking environment such as IEEE802.16MMR, the complexity improvement is still quite significant. For example, with the 2,500 base stations embedded in 77,000 relay stations (routing nodes) Table 3.6 shows the complexity reduction achieved by the tiered zone partitioning and the associated address aggregation presented. The tiered zone approach reduces the complexity by twelve orders of magnitude for the dense network and nine orders of magnitude for the less dense network.

	Complexity	**Steps in Algorithm**	**Efficiency**
Distance Vector among relay nodes (n = 770,000)	n^3	4.56×10^{17}	1
Distance Vector among roots (N =2,500)	N^3	1.56×10^{10}	29,218,112
M Tiers, P partitions (M=4, P= 9)	$(MP)^3$	46,656	9.78×10^{12}

Table 3.5: Complexity reduction for dense relay nodes (simulation is in next section)

	Complexity	**Steps in Algorithm**	**Efficiency**
Distance Vector among relay nodes (n = 77,000)	n^3	4.56×10^{14}	1
Distance Vector among roots (N =2,500)	N^3	1.56×10^{10}	29,218
M Tiers, P partitions (M=4, P= 9)	$(MP)^3$	46,656	9.78×10^{9}

Table 3.6: Complexity reduction for lower density of relay nodes (such as in IEEE 802.16MMR)

3.6.2 Radial Geometric Routing

A radial geometric approach is a further reduction of complexity for large scale routing. By this method, the optimal next hop cell for each destination is computed according to the angle between the source and destination roots based on their polar coordinates. Using a sector table, the system routes a packet towards the destination through the next hop cell. The distant location is translated to a sector and then resolved to a next hop cell ID. That next hop cell ID is referred to in the TREE_ID of the routing table to find that next hop node for the message which is the parent node for that routing tree.

3.6.2.1 The Sector Table

The sector table is the key to radial geometric inter-cluster routing, and every routing node in the network has one. The sector table identifies a next hop cell based on the angle the destination root is from the current root, utilizing several reference points on the fringe of the network. The format of the sector table is shown in Table 3.7. When a message is received by a routing node and the destination is not within the cell cluster, the sector table is referenced. The geographic location of the destination is assumed to be known. By the algorithm, the optimal next hop cell for each destination is computed according to the angle between the source and destination roots. For a representative root at every φ radians, a sector table has 2π / φ entries, and using those entries the system is able to route a packet towards the destination through the next hop cell. The distant location is translated to a sector and then resolved to a next hop cell ID. Then that next hop cell ID is referred to in the TREE_ID of the routing table to find that next hop node for the message which is the parent node for that routing tree.

Phi (φ)	The angle sector that packet needs to be routed towards
Tree ID	The ID of the next hop cell (will be a tree id within cluster)

Table 3.7: The sector table

3.6.2.2 Construction of the Sector Table

A destination aggregation scheme is employed where far away nodes are represented according to the radial scheme in Figure 3.8 below. Conceptually there are M representative virtual roots, also called "positioning routers," on the circumference of a circle that encloses the network. Roots maintain next hop information with respect to these representatives on the periphery. With sectors bounded at each $k \times \Theta$ radian increment (where 2π is a multiple of Θ, $k = 1,2,\ldots 2\pi/\Theta$) from a reference point in the center of the network, every root in the network has $M = 2\pi/\Theta$ entries to choose from to create its sector table. The sector table utilizes this information to select entries for larger sectors with respect to a base station, giving a total of $L = 2\pi / \varphi$ entries.

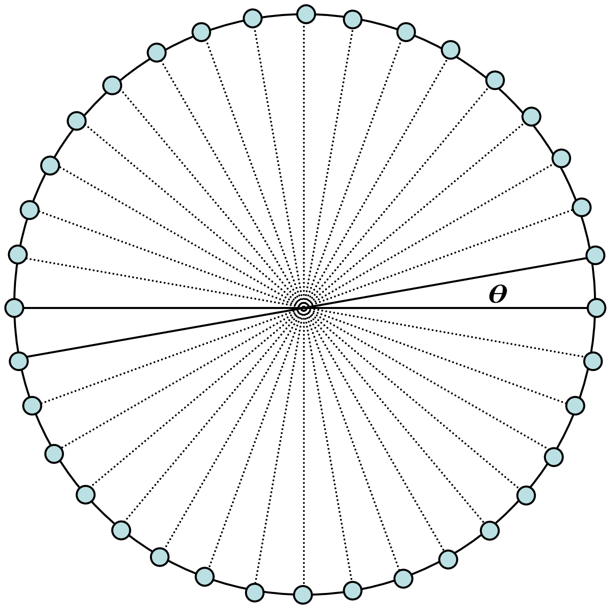


Figure 3.8: 2π / Θ representative (virtual) roots in network used for construction of sector table.

In this scheme there are 2π / $\boldsymbol{\Theta}$ representative roots on the periphery of the network that exchange distance vector information with all of the other roots in the system. These roots are taken as representatives at angle increments from some reference in the center of the network. As a result, every root in the network has a maximum of $2\pi/\boldsymbol{\Theta}$ entries in its large scale routing table, or sector table. All of the roots utilize this information to create a sector table of larger angle increments with respect to themselves. This is shown in the Figure 3.9.

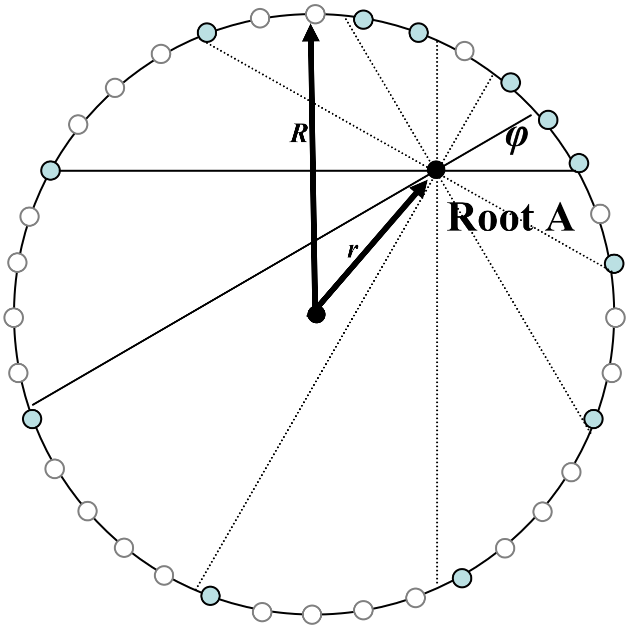


Figure 3.9: Sector table entries for root A are shaded representative periphery roots.

In the above figure, Root *A* is located a certain distance, *r*, from the center of the network. It has sector table entries at every increment angle φ with respect to itself, where φ > θ. These sector table entries are selected from the 2π / θ positioning routers on the fringe of the network. The positioning routers that are closest to each of the increments of the angle φ, as shown by the periphery nodes that are shaded, are included in Root A's routing table while the others are ignored. Notice that roots that are far from the center point of the network will have a relatively few number of positioning routers to choose from in the arc of the circle that is closest while they will have many to choose from in the distant arc. According to the theorem of intersecting chords, the measurement of the vertical angles that are created from two intersecting chords is half of the sum of the corresponding arc lengths on the circle. So that if there are two chords that intersect at 30 degrees, the sum of the two arcs that they create on the circle is 60 degrees.

3.6.2.3 A Condition on Number of Sector Table Entries

In general if Root A is a distance r from the center of a circle with radius R, then the smallest arc that would be created on the circle is $\varphi[1 - (r / R)]$. The proof of this is given below:

Lemma: If Root A is at a distance r from the center of a circle with radius R, then the smallest arc that would be created on the circle is $\varphi\, [1 - (r / R)]$. The proof of this is given using Figure 3.10:.

We need to prove: $\text{arcAC} \geq \varphi\, (1 - r / R)$:

Given: $PQ = R + r$, $PR = R - r$

If $\text{arcAC} \geq (1 - r / R)\varphi$

then $\text{arcBD} < (1 - r / R)\varphi$

then $AC/BD \geq (1 - r / R) / (1 + r / R)$

$\geq (R- r)/ (R + \mathrm{r})$

Using the fact that triangles ACP and DBP are similar triangles,

$$AC/BD = AP/PD = CP/PB \qquad (1)$$

Since the intersecting chords theorem states

$AP \times PB = CP \times PD$

Then,

$(R + r) \times (R - r) = AP \times PB = CP \times PD$

and therefore,

PD = $(R + r) \times (R - r)$ / CP. (2)

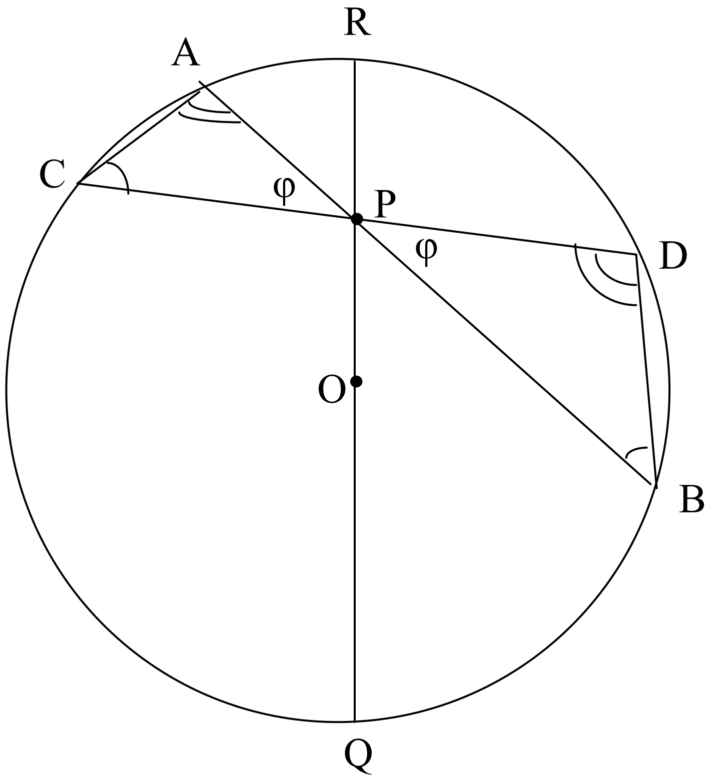


Figure 3.10: Proof of smallest arc of intersecting chords

Substituting (2) into (1) yields

AC / BD = AP / $[(R + r)(R - r)$ / CP]

and

AC / BD = (AP x CP) / $[(R + r)(R - r)]$ (3)

Since the distance from a point inside a circle through the center of the circle to the circle's edge is longer than distance from that point to any other point on the circle's edge we have:

PQ > PB and PQ > PD.

Since,

PQ x PR = PB x AP = PC x PD

it follows that

AP > PR = $R - r$

and

CP > PR = $R - r$ (4)

Applying (4) in (3) results in:

AC/BD = (AP x CP) / $[(R + r)\ (R - r)]$

and

AC/BD $\geq [(R - r)\ (R - r)]/[(R + r)(R - r)]$

AC/BD $\geq (R - r)/(R + r)$ Q.E.D.

Hence, arcAC $\geq \varphi\ (1 - r / R)$.

As this applies to our problem, Root A will have an entry in its sector table for every φ if φ * [1 - (*r*/*R*)] > Θ. The resulting sector table will contain $L = 2\pi/\varphi$ entries for each node. Nodes sharing the same home root will have the same sector table entries. It is thus important that Θ be small enough and φ be large enough so that roots distant from the center have fully populated sector tables. We may assume that roots that do not satisfy φ[1 - (*r*/*R*)] > Θ are on the outer fringe of the network and within the same cell cluster as their destination. In this case, intra-cluster routing is used. The resulting sector table will be of the form shown in Table 3.8 below:

φ	**Next root**
0	B
π/6	C
π/3	D
π/2	E
2π/3	E
5π/6	F
…	…

Table 3.8: Sector Table example

Large scale routing tables, or sector tables, will contain 2π / *φ* entries in them for each node. Nodes belonging to the same root will have the same large scale routing table.

3.6.2.4 Complexity

Over the years, there has only been marginal improvement in the $O(n^3)$ complexity of optimal algorithms that calculate shortest paths to and from every resource using distance vectors in a network of *n* nodes [9]. Since this is so costly in network overhead and routing table size, heuristic routing approaches that greatly reduce complexity while maintaining relatively good accuracy are much more appealing in practice. As an example, Table 3.9 compares the routing complexity of four methods: first an optimal algorithm, secondly a method whereby only all of the roots in the system exchange distance vector information, and thirdly the hierarchical partitioning method discussed in 3.6.1 and lastly the radial geometric method with LSSON we have outlined here. With n = the total number of nodes in the system, and N = the total number of roots in the system this demonstrates a drastic reduction in routing complexity. The complexity of the radial geometric approach is O(1) with respect to the total number of nodes in the system. Thus, as nodes are added no additional entries or overhead messaging is required.

Method	Complexity
Optimal Routing	$O(n^3)$
Root-level Distance Vector	$O(N^3)$
M Tiers, P partitions	$(MP)^3$
Radial Geometric Routing	O(1)

Table 3.9: Routing table complexity comparison

3.7 Simulation Scenarios

The large-scale routing methods presented in the previous sections drastically reduce routing complexity. To empirically analyze the performance of our large scale routing system, we simulated a very large scale routing environment. There are numerous simulators available that are designed for networks with relatively small numbers of nodes, from tens to possibly hundreds of nodes. These simulators tend to focus on the detailed interactions between one node or protocol and another, with the remaining nodes acting as routers or generating noise. With very large networks and a new routing protocol such as ours, we are interested in first examining the feasibility and promise it holds using very basic macro metrics. Thus, we have designed our simulator to focus on routing characteristics, comparing our protocol's performance to the optimal case.

Future multiple-hop networking environments will utilize dense communication infrastructures with thousands of nodes. To reflect this trend, our simulation included 693,000 routing nodes and 2,500 roots (or base stations). The network was laid out in a simple grid manner with each of the routing nodes and roots being stationary and able to handle routing packets between mobile nodes located anywhere in the network. One thousand random source-destination pairs were chosen and four routing schemes were simulated: a breath first search that yields the optimum hop count from the source to the destination in the network, a hierarchical routing model that uses tree routing within cell clusters and a distance vector routing protocol among all of the 2,500 roots in the system, our model that utilizes tree routing with cell clusters and root aggregation among tiered zones as described in Section 3.6.1, and radial geometric routing as described in Section 3.6.2. The performance and overhead of each was compared.

In this simulation all nodes in the system have the same signal range and the same number of neighbors. In this dense grid network, each node has 12 neighbors within its range and packets can hop to any one of the neighbors when routed. Based on the X-Y-Z messaging described earlier, routing trees rooted at each cell root were created but were limited to the cell cluster as shown in Figure 3.11a. This tree creation was also simulated in the random absence of nodes (in case of node failures) and also shown in Figure 3.11b. The simulation used a 10,000 x 10,000 unit grid. Every node in the grid was placed at multiples of 20 units in this grid. Roots are between 100 and 150 units apart, forming groups of 7 cell clusters.

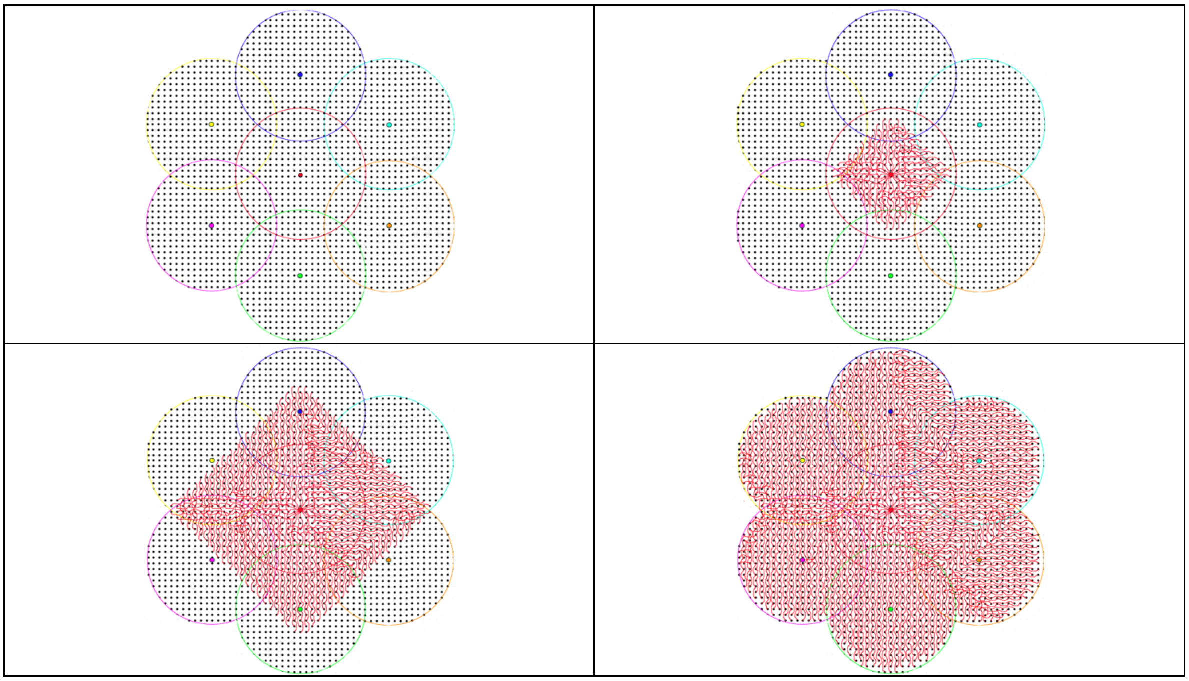


Figure 3.11a: Routing Tree creation from cell root outward

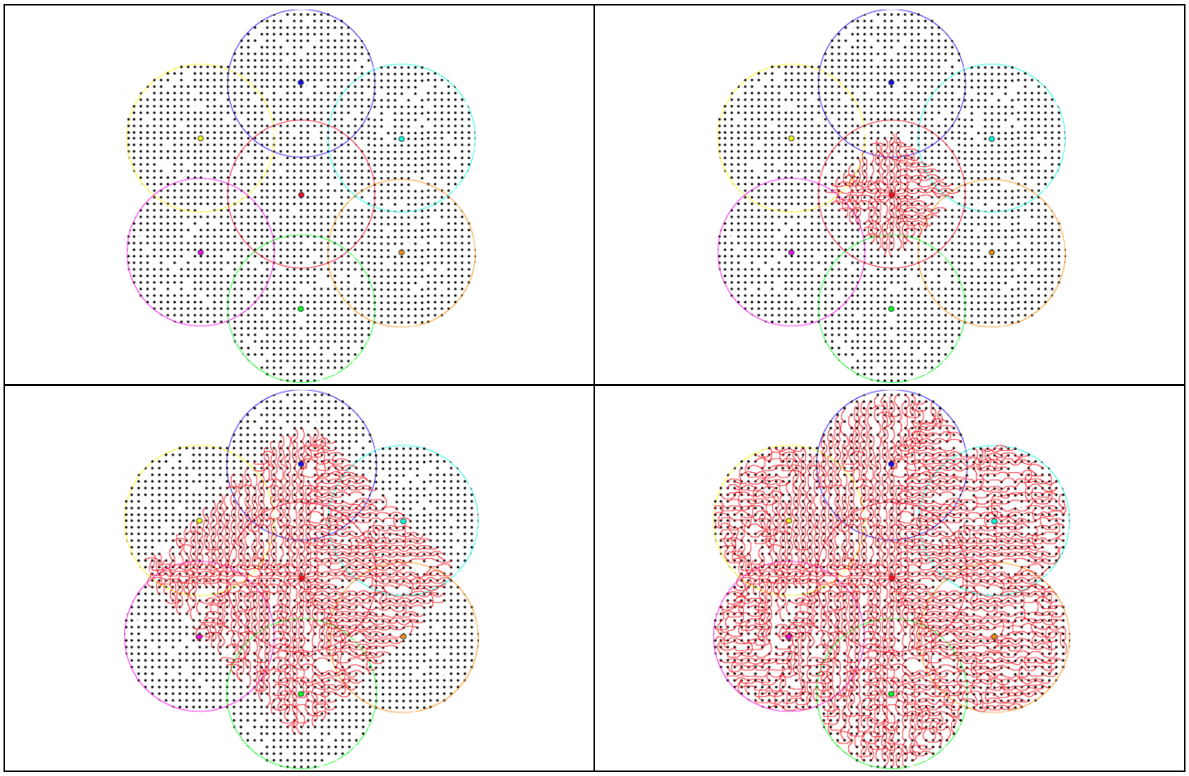


Figure 3.11b: Routing Tree creation from cell root with 10% random node failures

Once all of the trees are created for each cell, every node has a routing table entry for its home root as well as each neighboring root that is part of its cell cluster as shown in Figure 3.12. In this Figure, all nodes in C_1 are part of 7 routing trees, one for its home cell (C_1) and one for each of the adjacent cell roots in the cell cluster ($C_2 – C_7$). Figure 3.13 displays an example of a packet being routed from one node to a node in a remote cell. This involves inter-cluster routing.

With 1,057 simulated sample paths in the 693,000 nodes in network the average optimal hop count was 253.72. The standard error of a statistic is the standard deviation of the sampling distribution of that statistic. The estimated standard error of this point estimate is $S/\sqrt{n}$ with S being the standard deviation of the sample. With S = 136.25 and n = 1,000, this gives $136.25/\sqrt{1000}$ = 4.31. The standard error is about 1.67 percent of the sample mean, implying that we have obtained a relatively precise point estimate of the average hop count in the system. If we can assume that the hop count is normally distributed, then two times the standard error is 8.62, and we are highly confident that the true mean hop count is within the interval 253.72 ± 4.31 or between 249.41 and 258.03. The distribution of the hop counts is shown in Table 3.10.

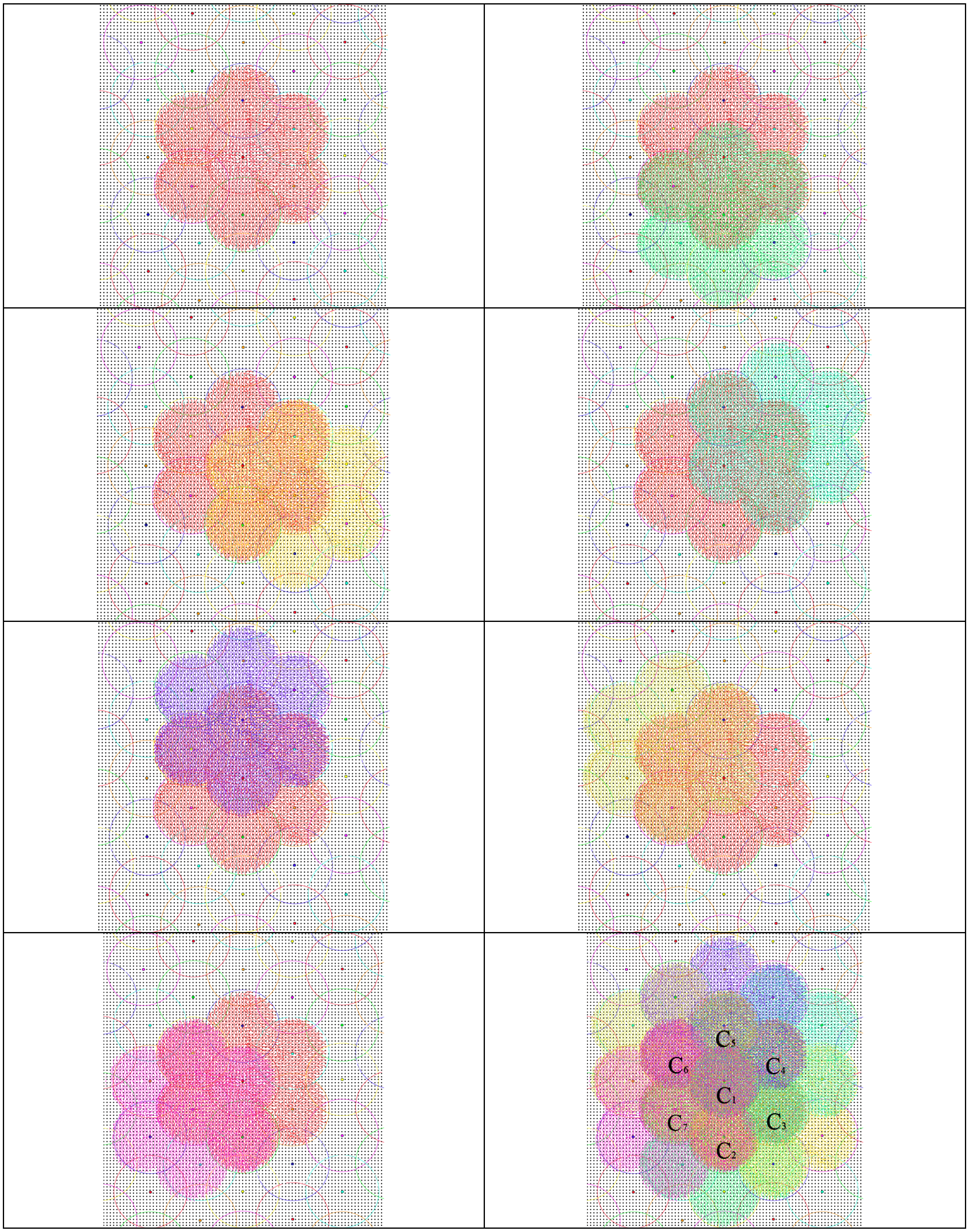
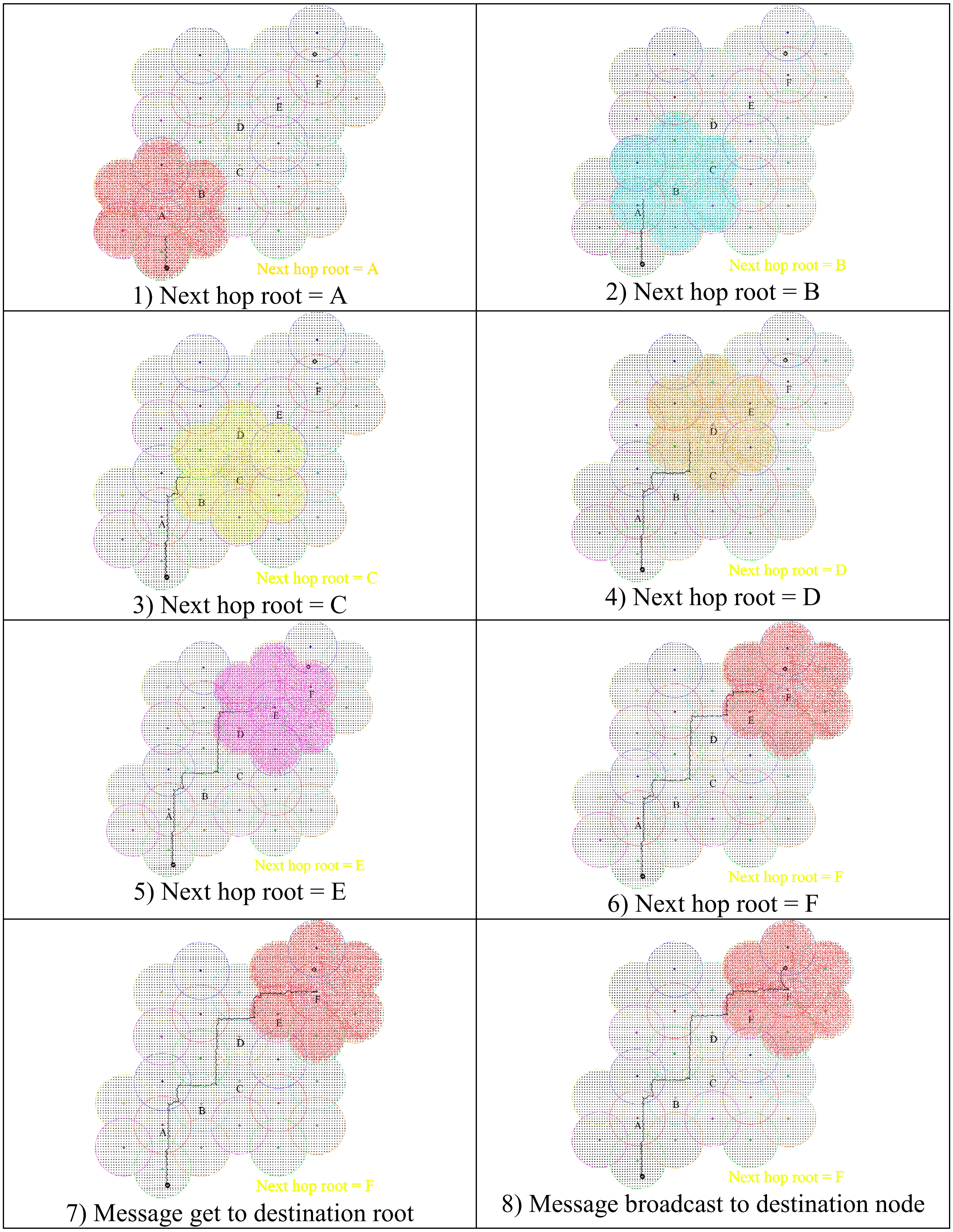


Figure 3.12: A Cell Cluster: Seven trees that make up a cell cluster centered at C_1



1) Next hop root = A

2) Next hop root = B

3) Next hop root = C

4) Next hop root = D

5) Next hop root = E

6) Next hop root = F

7) Message get to destination root

8) Message broadcast to destination node

Figure 3.13: Inter-Cluster Routing Example.

# hops	Frequency	%
1-100	139	13.28%
101-200	246	23.50%
201-300	288	27.51%
301-400	217	20.73%
401-500	101	9.65%
501-600	43	4.11%
601-700	12	1.15%
701-800	1	0.10%

Table 3.10: Distribution of hop counts in simulation

There were 6 scenarios that were simulated:

1. Optimal. The optimal hop count using a breadth first search between the source and destination node. Each node maintains routing entries for all 693,000 nodes.
2. Root Distance Vector. The hop count using a cluster based tree for local routing and next hop to the next root and distance vector routing among the 2,500 roots. Each root had to maintain routing entries for all 2,499 roots.
3. Hierarchical L1 low. This used routing table entries hierarchical zones: all t-0 zones within an L1 distance equal to 1250 units, t-1 zones within an L1 distance equal to 5,000, and t-2 zones distances outside of 5,000 units. Each root had on average 213 routing table entries for other roots (out of a total of 2,500), more than an order of magnitude reduction in routing table size.
4. Hierarchical L1 high. More table entries were included in this simulation. For the grid layout described above, t-0 zones within an L1 distance equal to 2500 units, t-1 zones within an L1 distance equal to 7,000 and t-2 zones outside an L1 distance of 7,000 were

included. Each root had on average approximately 536 table entries, a significant but not huge reduction in routing table size.

5. Hierarchical max(dx,dy) low. Max(dx,dy) was used instead of L1 distance. For max(dx,dy) from the root less than or equal to 1250 all t-0 zones were included in routing tables. For max(dx,dy) < 5000 all t-1 zones were included in routing tables. For max(dx,dy) > 5000 all t-2 zones were included. Each root had an average of 361 routing table entries.
6. Hierarchical max(dx,dy) high. More routing table entries using max(dx,dy) distance from each root. Here, all t-0 were included within max(dx,dy) < 1250 as above. The only changed compared to #5 was that the distance for t-1 zones was extended to max(dx,dy) = 7000. Outside max(dx,dy)=7000 all t-2 zones were included. With this each root had approximately 409 routing table entries.
7. Radial Geometric: Directional routers, placed at $\boldsymbol{\Theta}$ = 8 degrees were chosen among roots most distant from the center of the network and within the $\boldsymbol{\Theta}$ increment. For each node, a directional router for each $\boldsymbol{\varphi}$ = 30 degree increment was chosen. This resulted in each root having 12 entries in its sector table, one for each 30 degree directional increment.

Once the trees were set up for each root, then the nodes' routing tables were set up according to each particular methodology. Six different network sizes were simulated, ranging from 70,000 nodes and 268 roots to more than 660,000 nodes and 2500 roots. One hundred simulations of each network were done with a random source and random destination. For each source and destination combination, hop counts for each of the 7 methodologies were generated.

3.7.1 Simulation Results

As shown in Table 3.11, performance of each routing scheme depends on the number routing table entries. Not surprisingly, the greater the number of routing table entries, the closer that scheme came to the optimal hop count. The root level distance vector scheme performs closest to optimal. It averages less than a 2% hop difference from the optimal path which requires a table entry for every root in the system. The hierarchical zone routing schemes drastically reduce the routing table size and control packet overhead while still maintaining reasonable path length values. The cases where hierarchical zone routing most deviate from the optimal path in hop count are those in which the source and destination are quite far apart and the higher tier aggregation causes some imprecision. The radial geometric approach produces acceptable hop counts, within 10% of optimal, while reducing the number of routing table entries by an order of magnitude compared to the hierarchical zone routing methods.

	Routing Table Entries per node	Total Nodes to Number of Table Entries Ratio	Avg Hop Count	Hop Count to Optimal Ratio
optimal	663,943	1	253.72	1.00
root dv	2,500	266	258.5	1.02
hier1 high	536	1,239	263.01	1.04
hiermax high	409	1,623	264.96	1.04
hiermax low	361	1,839	267.3	1.05
hier1 low	213	3,117	267.8	1.06
Radial Geometric	12	55,329	279.14	1.10

Table 3.11 Large Scale Routing Simulation Scenarios

3.7.2 Advantages of Radial Geometric Routing

The radial geometric approach produced an impressive combination of routing table complexity reduction and routing efficiency. Performance also scaled well with network size, showing equally good performance in a relatively small network compared to a much larger one as shown in Figure 3.14 and in Table 3.12. Radial geometric routing performed equally well for long routes and shorter routes within a given network, as shown in Figure 3.15 and in Table 3.13. Since inter-cluster routing uses distant reference routers, it was plausible that the closest source-destination pairs might be the worst performing in this scheme. However, when we examine the distribution of number of hops in the large network of over 660,000 nodes and compare that to how far off from optimal the route was, we found that radial geometric routing is quite robust with respect to the distance between source and destination. The ratio of the radial geometric routing hop count to the optimal hop count for shorter routes (those 100 optimal hops and less) was 1.09 while the same ratio for the longest routes (those with more than 400 hops) was 1.10.

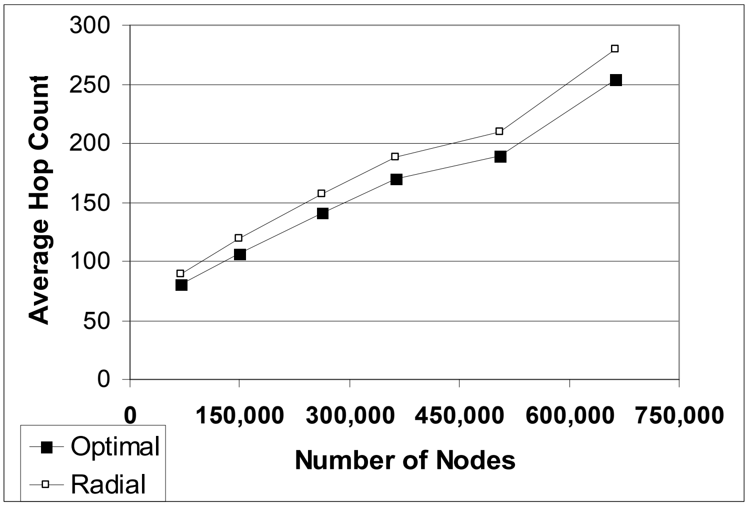


Figure 3.14: Performance of the radial geometric approach with varying network sizes.

Roots	**Nodes**	**Radial:Optimal**
268	70,304	1.11
548	150,038	1.12
1,014	262,855	1.11
1,402	363,027	1.11
1,932	504,961	1.11
2,500	663,943	1.10

Table 3.12: Ratio of radial geometric hop count to optimal for different network sizes

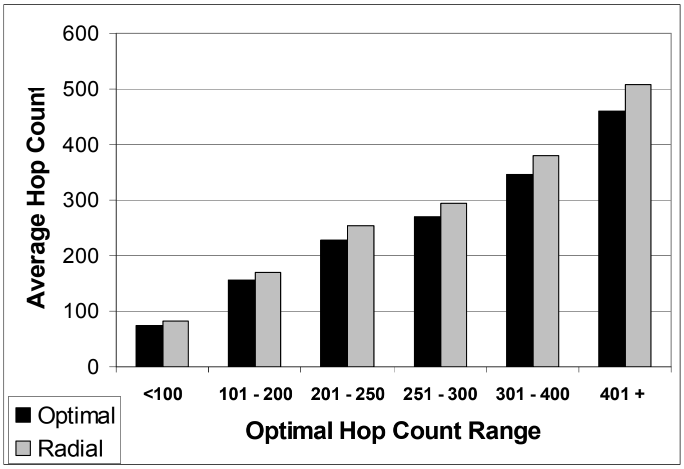


Figure 3.15: Radial geometric routing performance versus for varying hop count scenarios

Hops	Optimal Avg	Radial Avg	Radial:Optimal
<100	74.9	81.6	1.09
101 - 200	155.6	169.8	1.09
201 - 250	228.7	253.4	1.11
251 - 300	269.1	294.2	1.09
301 - 400	345.8	380.2	1.10
401 +	460.5	508.1	1.10

Table 3.13: Ratio of radial geometric hop count to optimal hop count for different path sizes in a very large network.

3.7.3 LSSON Performance

Based on the analysis and data generated, LSSON routing is a decent heuristic for large scale routing. It provides for drastically simplified routing tables and reasonably efficient delivery of messages and scales very well to very large networks containing hundreds of thousands of nodes. Moreover, from the simulations it treats relatively long routes as fairly as relatively short routes in the network.

LSSON large scale routing differs from geographic routing in that it collects actual cost metrics from representative distant directional routers instead of simply forwarding a packet in the direction of the destination. Moreover, the tree structure overlay assures reliable, self-configuring and self-healing local routing.

Chapter 4

Virtual Circuits and Load Balancing in LSSON

4.1 Introduction

The expansion of wireless networks has led to a greater demand of services to customers over and above traditional voice service. Devices such as the iPhone [79], and services such as PacketVideo Networks [80] are bringing applications to the mainstream that increase the need for Quality of Service (QoS) guarantees through the reservation of network resources. Therefore, in developing ad hoc and multi-hop cellular networks there is utility in allocating resources by the use of virtual circuits, a reservation of bandwidth along a fixed path between the endpoints of the connection. In order to achieve this goal within the framework of our wireless multi-hop network design and topologies, we have developed an effective means to accommodate virtual circuits and quantify its feasibility.

In this chapter, we introduce a means by which virtual circuits can be established in a self-organizing multi-hop cellular network and demonstrate ways available system capacity can be used more efficiently. The system utilizes stationary low-cost, low-power routing nodes between mobile data terminals and base stations that self-organize into routing trees. Virtual circuits are established using the routing trees and system capacity is shown to be utilized more efficiently by a multiple tree load balancing approach that enables virtual circuit connections from a user node to the land-based core network (e.g., the internet or telephone network) through one of many base stations including the user node’s home base station. By analysis and simulations, we compare the Erlang B capacity of the network in handling virtual circuit traffic using several routing scenarios. Furthermore, a dynamic tree approach is used to even more effectively balance the load and increase available capacity. Finally we comment on parameters in routing tree creation that affect system performance.

4.1.1 Static Assignment of Virtual Circuits

The simplest method to set up virtual circuits is one with static routing trees whose routing remains fixed, independent of the state of the network. Static routing requires little or no computational resources and mainly requires effective real-time traffic forwarding. Early telephone networks used this kind of routing, within Bell’s fixed hierarchical network [81]. Since no real-time routing computation was necessary, the bulk of the engineering work was related to designing the network to exceed peak-traffic predictions and making sure redundant links were available in the case of failures in the network.

Static routing uses table mappings that do not change unless the network administrator alters them. Algorithms that use static routes are simple to design and work well in environments where network traffic is relatively predictable and where network design is relatively simple.

4.1.2 Dynamic Assignment of Virtual Circuits

Because static routing systems cannot react to network changes effectively, they generally are considered unsuitable for large, constantly changing networks. With advent of smart digital switches, evolution to dynamic routing was made possible [82]. Dynamic routing enables the selection of routes based on the state of the network. Here, routing is changed depending on the state information at the time of the routing request. The majority of the dominant routing schemes today are dynamic routing algorithms, which adjust to the changing network state. If network information indicates that a change has occurred, the routes are recalculated and routing tables updated. With virtual circuits, a newly arriving call at a source node is routed along the minimum length path to its destination node. All packets belonging to this call follow the same path through the network. If that route then becomes overly congested, new routes for virtual circuits can be dynamically assigned to better use network capacity.

4.1.3 Multiple Virtual Circuits

Some more sophisticated protocols support multiple paths to the same destination. Unlike single-path algorithms, multi-path algorithms permit traffic multiplexing over multiple lines. The advantage of multi-path algorithms is that they can provide substantially better throughput and reliability through load sharing. Multiple paths can be offered either with static virtual paths or dynamic virtual paths. In either case, they offer improved load balancing and reliability [83].

4.2 Notation Used in this Chapter

For clarity we shall adhere to the following notation in this chapter:

TLN $\equiv$ Top Level Node or a node one hop from the base station.

$\rho_i \equiv$ load of incoming calls to node *i*.

$\rho \equiv$ load vector of incoming calls to every node within a routing tree.

$L \equiv$ Traffic load on a cell.

$L_i \equiv$ Traffic load on a node *i*.

$T^k \equiv$ The routing tree rooted at BS *k*.

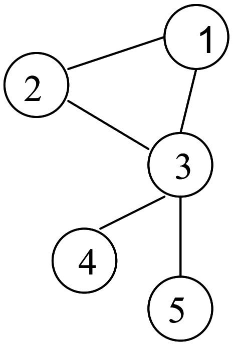
$T_i^k \equiv$ the sub-tree of T^k rooted at node i.

4.2.1 Physical Topology Graph

A physical topology graph of a network can be conveniently represented by an *n* x *n* adjacency matrix , **A** [9], where the number of nodes in the network is n. Each entry in the adjacency matrix **A** is either 0 or 1 depending on whether there is a two-way communications link between Node *i* and Node *j*:

$$[\mathrm{A(i,j)}] = \begin{cases} 1 \; if\ edge(i,j) \in E(t) \\ \quad 0 \;\; Otherwise \end{cases}$$

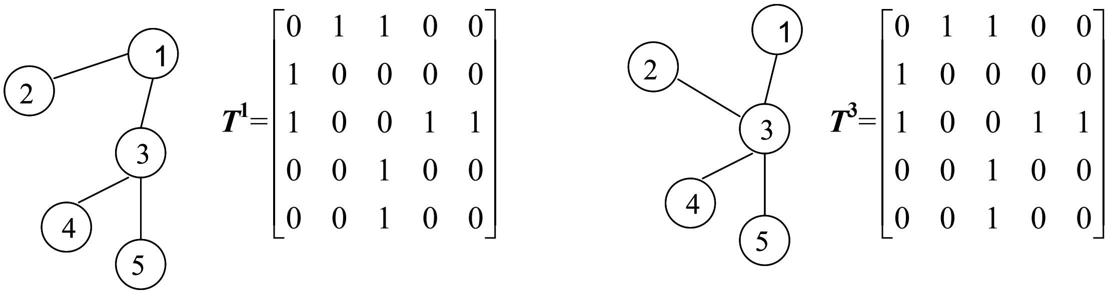
Figure 4.1 shows an example of a physical topology graph and an adjacency matrix that represents the physical topology.



$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Figure 4.1: A physical topology graph and its corresponding adjacency matrix

In our work, we utilize rooted trees as the routing topology for the self-organizing wireless network. Based on the physical topology of the network, multiple tree routing topologies are constructed and maintained. A tree routing topology at time t is an acyclic sub-graph of the physical topology where T^k is the routing tree rooted at node *k* and includes the set of nodes belonging to that tree. Figure 4.2 shows an example of a physical topology and two possible routing topologies - one rooted at Node 1 and another rooted at Node 3. We also show the corresponding routing topology matrices for T^1 and T^3 respectively.



$$\boldsymbol{T}^{\mathbf{1}}=\begin{bmatrix}0&1&1&0&0\\1&0&0&0&0\\1&0&0&1&1\\0&0&1&0&0\\0&0&1&0&0\end{bmatrix}\qquad \boldsymbol{T}^{\mathbf{3}}=\begin{bmatrix}0&1&1&0&0\\1&0&0&0&0\\1&0&0&1&1\\0&0&1&0&0\\0&0&1&0&0\end{bmatrix}$$

Figure 4.2: Routing topologies for trees rooted at nodes 1 and 3 and their associated routing topology matrices.

4.3 Virtual Circuits in LSSON

The basis of the physical topology we use for virtual paths is a multi-hop cellular network model with fixed relay nodes. Mobiles communicate upstream through fixed relay nodes to a base station which has a sufficiently large bandwidth connection to a land-based core network such that the backhaul will never be a bottleneck. Mobiles register with their local base station as is typical with cellular systems. Messages delivered downstream to mobiles are delivered via one-hop direct broadcast from high power base stations, or optionally in multiple hops downstream to nodes that are out of broadcast range via source routing. The fixed relay nodes are self-organized into a multiple-tree hierarchical routing network according to the Integrated Multi-hop Cellular Data Network presented in [63]. Routing trees for upstream packet forwarding are maintained and rooted at each base station and each routing tree spans all nodes that have multi-hop access to that particular base station. Nodes in the network have one parent node per routing tree and forward data messages destined for a particular base station to the parent node in the routing tree. Data messages are sent along the routing tree until they finally arrive at a base station and can then be routed through a high bandwidth infrastructure. Then the message is broadcast downstream to the destination node through that node's home base station. In Figures 4.3a and 4.3b below two base stations have established routing trees through the network, thus each node has two alternative trees for routing their packets. In Figure 4.3c, Node 3 can either forward its packets to its parent node in BS1's routing tree, Node 2, or it can forward packets to Node 4, which is Node 3's parent node in the BS2 routing tree.

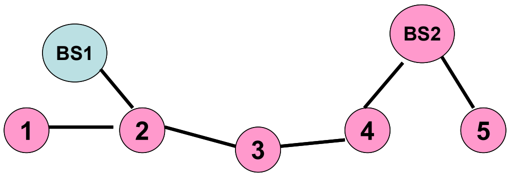


Figure 4.3a

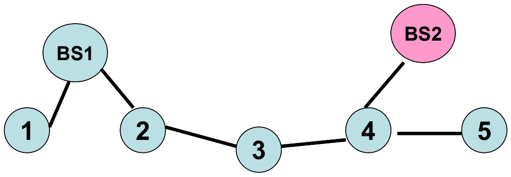


Figure 4.3b

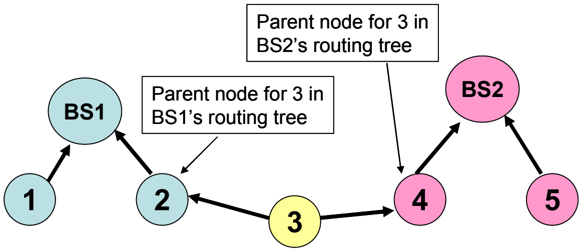

Parent node for 3 in BS2's routing tree
Parent node for 3 in BS1's routing tree
BS1
BS2
1
2
3
4
5

Figure 4.3c

Packet data using this routing topology has been studied in depth previously [9]. However, the use of virtual circuits and the concept of limiting the routing trees to cells and clusters has not yet been explored. Even so, the routing topology created by the system described above is a natural candidate for use with virtual circuits, as the routing trees can be used to set up virtual paths. If we allow for the virtual paths to adapt to the dynamically changing network traffic patterns, the self-configuring and adaptive aspects of [63] enable dynamic virtual paths as described in Section 4.1.2. Moreover, if we use a "multiple layer" or multi-tree system as described in [63], we may set up multiple virtual paths as described in section 4.1.3.

4.3.1 Traffic Patterns in a Single Tree Network

The traffic load that a node in a routing tree, *T*, experiences is equal to the summation of all the traffic of all of the children of that node plus that node's own traffic. Consider a system where calls arrive at each node distributed according to a Poisson process of mean λ and the duration of a call is exponentially distributed with mean duration 1/μ. Thus the call traffic load at each node is $\rho_i = \lambda_i / \mu_i$ for the nodes in the tree, for all $i \in T$. Therefore, the total load for the base station will be $L = \sum_{i \in T} \rho_i$.

For the single-tree example illustrated in Figure 4.4, if the load vector (in Erlangs) for the tree rooted at base station 1 is $\boldsymbol{\rho}$ = [4 2 1 3 1 5 3 1 1] then the total load for this tree is $L = \sum_{i \in T}^{n} \rho_i$ = 21 Erlangs. The total load for Node 6 will be $L_6 = \rho_6 + \rho_7 + \rho_8 + \rho_9 = 10$. Likewise, the total load for Node 4 will be L4 $= \rho_4 + \rho_5 =$ 4 and the total load for Node 1 will be L1 $= \rho_1 + \rho_2 + \rho_3 =$ 7. Note that the total traffic load of a non-BS node equals the sum of traffic loads of nodes on the sub-tree rooted at this node.

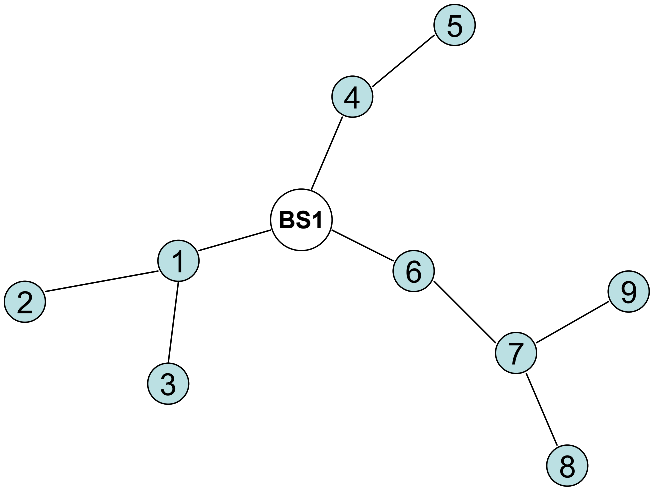


Figure 4.4: The routing tree for a base station, T^1

It is clear that the backhaul capacity of the BS should be based on *L*. Since we assume that all base stations are interconnected through a sufficiently large bandwidth backhaul network, the backhaul is assumed to have a higher capacity than *L*. Therefore, we extend the reasoning above to show that the capacity bottleneck in routing tree based uplink systems are the top level nodes. A *top level node* (TLN) is a node that is one hop from the base station and the node that forwards all of its calls and those of its children in its sub-tree to the base station. Consider a sub-tree T_f, rooted at a TLN *f* where T_f contains Node *f* and all of its descendant nodes within *T*.

Then the load on the TLN is $L_f = \sum_{i \in t_f} \rho_i$. This value may be used in sizing the upstream of the top level nodes.

4.3.2 Multiple-Tree Traffic

In a multiple cell network, there are multiple base stations, each of which is the root of a routing tree. Thus, the number of routing trees equals the number of cells. If there are *k* cells, then each node is part of *k* routing trees with each tree rooted at one of the *k* base stations. We shall enumerate the routing trees using a superscript. In particular, T^k is the routing tree rooted at BS *k*, T_f^k, the sub-tree of T^k rooted at a TLN *f*. For such a network, traffic at a particular node may be distributed over many trees. If $\rho_i = \lambda_i / \mu_i$ is the load of the traffic originating at Node *i* and p_i^m is the probability that Node *i* uses tree *m* for a call originating at that node, the load of traffic originating at node *i* for tree *m* will be $\rho_i p_i^m$. Thus the total load for the nodes rooted at a base station *k* will be $L^k = \sum_{i \in T^k} \rho_i p_i^k$.

In the same way for the sub-tree of T^k rooted at a last hop node *f*, T_f^k, the load will be $L_f^k = \sum_{j \in T_f^k} \rho_j p_j^k$. If the routing trees are static, then $L_f^k \geq L_e^k$ if $e \in T_f^k$. The upstream load on the top level nodes thus becomes the bottleneck for every subtree T_f^k. Therefore, when static routing trees are being used to establish virtual circuits to a base station and all routing nodes have the same upstream capacity, the limitation on total network capacity becomes a function of the number of TLNs. Specifically, if each node is capable of handling *m* virtual circuits, and there are *q* TLNs in a tree, then the maximum total of virtual circuits that can be established within a routing tree is mq. For no blocking to be experienced in a particular tree, we must have: $m \geq \sum L^{k}_{f}, f \in$ set of top level nodes in T^{k}.

Note that the probabilities of a node selecting a certain tree, p_{i}^{m} depends on the scheduling algorithm by which a node chooses a tree for a virtual circuit request.

4.4 Load Balancing

We assume that for successful virtual circuit establishment that once a path is found by the algorithm, the nodes that lie on the path from the source to the destination all have resources committed for the connection. This connection then takes away from subsequent requests for resources along this set of nodes for a particular tree. We assume calls can not be pre-empted after they are established. We consider two cases in our network to assess the advantages of using multiple trees:

1) The case where the VC path is confined to the local BS.

2) The case where the VC path can cross cell boundaries.

4.4.1 Blocking Rate for Home Cell Virtual Circuits

In this network, each fixed node is within the downstream broadcast range of at least one base station. Based on a criterion such as signal strength in a control channel, the node chooses one “home” base station. When the self-organizing network creates its routing tree, each node is assigned one tree to route to the base station. While this network has the same capacity as the others considered later, it does not provide any load balancing capability between cells. There is fixed routing from each node and only one virtual path available, and thus there is no alternative as to how a virtual circuit is routed. When the TLN of a routing sub-tree reaches its maximum usage capacity, it must block all incoming calls along that sub-tree. The routing in this case is similar to "shortest path routing," where the virtual circuit uses the shortest path to the nearest base station. Prior research has shown that this imposes limitations on the capacity of wireless networks since shortest path routing for virtual circuits creates traffic bottlenecks in the middle of the network, making it difficult to spread the load to less loaded areas of the network [84].

For a traditional cellular network, the Erlang B formula gives the probability of all channels in a cell being busy and is given by: $B(N, L) = (L_k^N/N!)/(\sum_{i=0}^{N} L_k^i / i!)$, where N is the number of channels in the cell and L_k is the traffic intensity in each cell. As shown in [85] for a multi-cell system, the average blocking probability for the entire system reaches a minimum value when the traffic is evenly distributed among all cells in the network. Thus in a multi-hop cellular environment, finding a way to distribute the load evenly among all cells is a key to a lower blocking rate. However, in a fixed single cell routing tree scenario, there is nothing that spreads the load from a cell with heavy traffic to one with less traffic. With no option provided for a node to route to other parts of the network, load balancing is not possible. At any period of time, there might be "hot spot" cells and the inability to share this traffic load with more less loaded cells may lead to substantial capacity waste within the system.

With respect to a multi-hop cellular network, there is at least one routing tree available to each node. As mentioned before, the primary limitation in a multi-hop cellular network with identical routing nodes throughout the system using routing trees for upstream communication to the root is the capacity of the TLNs, those that are one hop from the base station. Therefore, the probability of blocking in a cell is related to the number of TLNs there are and the resources available to them.

Consider a multi-hop cellular network, in which a cell as *q* TLNs and each TLN has *m* virtual circuits available. Then each node will be part of a sub-tree that has a capacity of *m* virtual circuits at the TLN to the home base station. Thus, in the Erlang B formula the number of circuits available is m for each sub-tree terminating at the root. In the best case, the traffic load, *L*, within the cell is evenly distributed among the TLNs. This scenario is depicted in Figure 4.5. Here, there are 6 TLNs and each of them is the root of a sub-tree. If the total load in the cell is L and we assume that this is evenly distributed among the 6 sectors served by each TLN, then the traffic load in each sector is L/6. Therefore, the lower bound on the blocking probability according to the Erlang B formula using the values of number of virtual circuits available and load as described above is B(*m*,*L*/6). In general for *q* TLNs in a cell, the lower bound on blocking will be B(*m*,*L*/q).

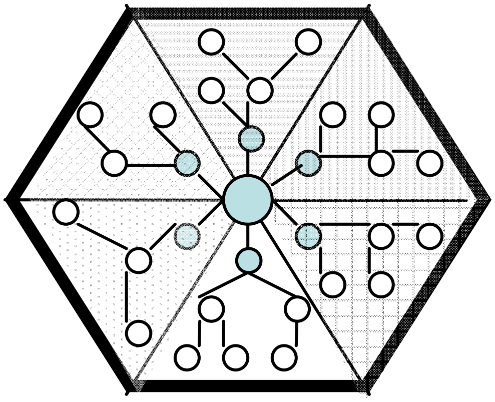


Figure 4.5: Six sectors in a cell each being served by a routing sub-tree.

It must be noted here that merely adding TLNs does not necessarily add capacity to a cell. In wireless systems with shared medium, there is a tradeoff between the number of TLNs and the amount of resources allocated to each one: a greater number of TLNs will result in a lower number of circuits available to each one, degrading system performance. Moreover, if feasible, adding sectors and TLNs that serve them adds additional expense and complexity to the system.

4.4.2 Multiple Trees

If we expand the reach of the routing trees from serving just one sector per tree to serving multiple sectors per tree, a decrease in blocking probability can be expected. Here, according to Figure 4.6 below, we have a set of cells, each with a TLN per sector. Each TLN serves an adjacent sector of the neighboring cell as well as its own sector in its home cell. In this case, nodes within Sector A and Sector B are part of two routing sub-trees, one rooted by the TLN in Sector A and the other rooted by the TLN of Sector B. If traffic is evenly distributed among each of the pairs of sectors, the traffic load served by one pair TLNs is $L/3$. The number of circuits available to nodes in those two sectors will be $2m$. Thus, the best case blocking probability in this scenario is $B(2m,L/3)$.

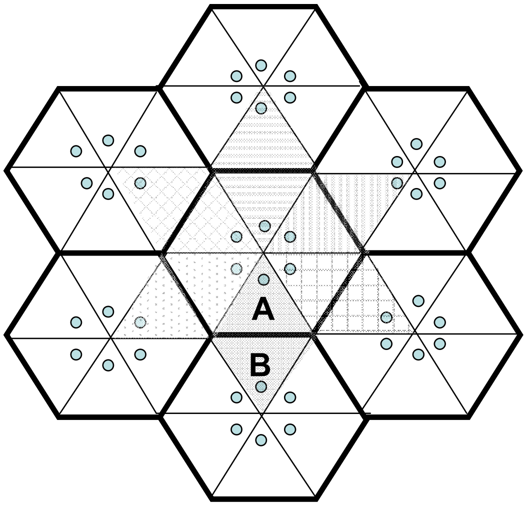


Figure 4.6: Adjacent sectors serve overlapping areas. Examples are sectors A and B and each similarly shaded area.

In general if there are *s* overlapping sectors, each served by a TLN and a load of *L* per cell with the load is evenly distributed among the sectors, the blocking probability per sector is B(*sm*,*sL*/6).

A multiple tree cellular network may be configured in such a manner that there is one routing tree per base station in a cell cluster and so that routing trees for each base station extend into neighboring cells and overlap with each other, as described in Section 1.3.2. Figure 4.7a shows a cluster of seven cells. The center cell, Cell 7, has a TLN for every sector. Each of those TLNs also serves a neighboring cell. Consider a scenario with every other cell outfitted similarly, with TLNs serving traffic in each sector as well as traffic in adjacent cells. In this case, TLNs will share some fraction of the cluster's traffic with other TLNs. Figure 4.7b shows a highlighted sector, Sector 4 of Cell 7. With the configuration of Figure 4.7a traffic in the highlighted sector will be serviced by the 7 TLNs shown. Since all 7 of those top level nodes may provide virtual circuits to traffic in that sector, it is labeled 7/7. Similarly, the other 5 sectors in that same cell utilize 6 of those 7 last hop nodes and, if traffic is distributed uniformly, then 6/7 of that traffic will be served by the last hop nodes that also serve the highlighted sector. With the same uniform traffic assumption, the fractions in each sector are the average proportion of traffic that have their virtual circuits provided by the set of TLNs that also serve the highlighted cell. When all of the fractions are added up, we find that the set of 7 TLNs serve $49s/7 = 7s$, or an aggregate of seven sectors. If the traffic in each sector is $L/6$, then the total traffic served by the 7 TLNs is $7L/6$. The uniform distribution of TLNs and of network traffic gives the best case scenario for call blocking: an Erlang B blocking rate of $B(7m,7L/6)$ for a 7 cell cluster.

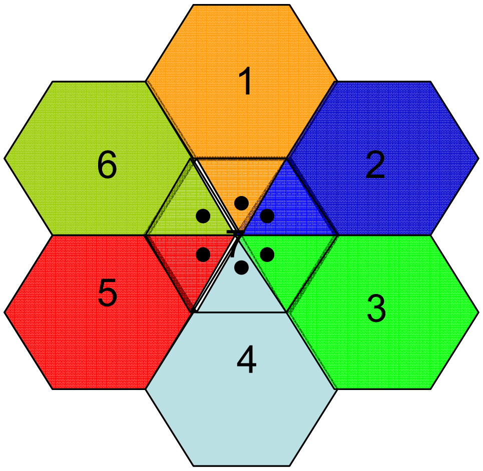


Figure 4.7a: TLNs serving adjacent cells

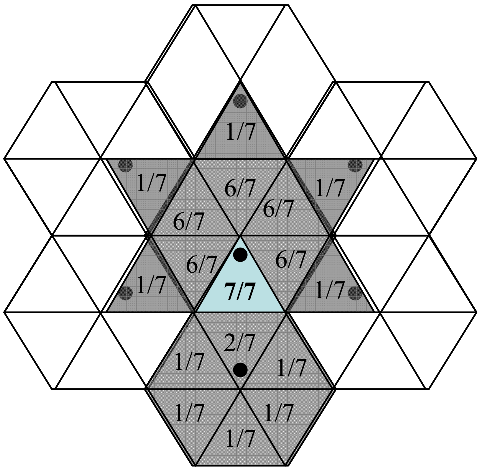


Figure 4.7b: The 7 TLNs serving highlighted sector with proportions of traffic in each sector they serve.

A cell cluster may be of any size. We would expect that the higher quantity of *s*, the better the load balancing qualities. A cellular cluster with 19 cells is shown below Figure 4.8. Here, an even distribution of routing trees requires that each TLN service its own sector plus 3 of the other cells in the cluster. Using the same analysis as with the 7-cell cluster, the highlighted sector uses 19 of the 19 TLNs shown. The fraction of traffic that the 19 TLNs serve in each sector is shown, with the assumption that traffic is uniformly distributed. The 19 TLNs thus serve an aggregate of 19 sectors worth of traffic or 19*L*/6. Since the number of available virtual circuits is 19*m*, we have an Erlang B blocking probability of B(19*m*,19*L*/6).

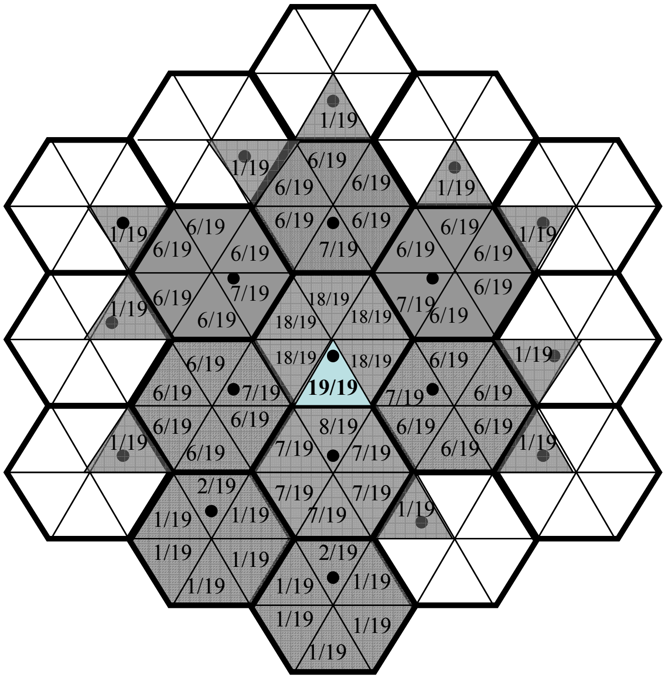


Figure 4.8: A cluster of size s=19 with the number of trees common to 19 trees that provide virtual circuits to the highlighted sector**.**

Generally, in a network with uniformly distributed traffic, where a quantity of *q* TLNs per cell each serve an equal load in a network with overlapping clusters, the total traffic load is *CL*/*q*, where *C* is the number of cells in the cluster. Since any node will have virtual circuit capacity available to it of *Cm*, the Erlang B blocking probability is therefore B(*Cm*,*CL*/*q*).

4.5 Algorithms for Choosing Trees

For multiple trees, when a message arrives at a routing node it can select any one of several routing trees, using a combination of load information for each tree and the hop count from that routing node to the root base station for that tree. This information is kept in a table at each routing node as in Table 4.1. In the original design [9], the load information for each tree *k* at node *i*, $\rho_i^{(k)}$ is passed down to the children in the tree with no strict rules regarding what value exactly should be updated in the field. In this study we assume that $\rho_i^{(k)}$ contains the number of virtual circuits, $\sigma^{(k)}$, on the most loaded node in the sub-tree and that it is updated frequently enough to provide an almost real-time view of how many virtual circuits are available in each routing sub-tree. We also assume a non-preemptive load-balancing approach; we use *n* identical routing nodes, and a number of independent call requests which arrive one by one. Each call request takes exactly one virtual circuit and must be assigned to an available routing tree serviced by the receiving node. This virtual circuit then consumes a one unit of virtual circuit resource from every node along the selected sub-tree to the base station. Once a virtual circuit is assigned, it cannot be reassigned. When there are no available resources for a virtual circuit to be established, the call is blocked.

k	Tree ID
$D_{\mathrm{i}}^{(\mathrm{k})}$	The depth of node i in tree k
$P_{\mathrm{i}}^{(\mathrm{k})}$	The parent node in tree k
$\sigma^{(\mathrm{k})}$	Largest number of virtual circuits along path to root of tree k

Table 4.1: Routing tree table of node i for static trees

The choice of routing tree to send a packet to a base station can be done in a number of ways. One method is to choose a tree with a path to a base station that has the lowest hop count. An alternative method that we have adopted is the selection of an available routing tree by choosing the least loaded tree, i.e. the tree that has the "min-max" number of used virtual circuits in any node along a tree, minimizing the maximum number of virtual circuits in a path. This algorithm finds the path to a base station with the lowest load. That is, it uses the path that leads to the smallest number of VC used along the path, and typically translates into the least number of virtual circuits that have already been established in the TLN. The disadvantage of this algorithm is that there will be likely more network resources, due to a higher hop count, consumed per virtual circuit. However, since this algorithm is likely to better spread the traffic among all cells in the system, it appears to be the most promising to maximize the use of the capacity of the network.

4.6 Dynamic Routing Trees

Routing trees can be either static or dynamic. If they are static, the TLN will always be the bottleneck if all routing nodes have the same number of virtual circuits available as described in Section 4.3.2. Moreover, if the routing trees that are originally generated carry an unevenly distributed load, either because there are a large number of child nodes of a particular TLN or because of a localized spike in traffic, some sub-trees will become overloaded while others will be under-utilized. This is depicted in Figure 4.9a where one TLN, which is the root of sub-tree T_2^1, has ample available capacity, serving 1 circuit originating from Node 5, while the other TLN which is the root of T_1^1 is using all 5 of its available virtual circuits, two each originating from Nodes 3 and 4 and one originating from Node 1. Thus, T_1^1 has no available capacity and additional connection requests to the BS will be blocked. However, if dynamic routing trees are used, then periodically the routing nodes will be reconfigured in such a way as to maximize the capacity by balancing the load. Trees can be regenerated based on the $\sigma^{(k)}$ value described above and an additional entry in the virtual circuit routing table to indicate how many virtual circuits are being used by that particular node, $v_i^{(k)}$. In the case of dynamic trees, the TLNs may not necessarily be the bottlenecks, as will be described later. Thus, $\sigma^{(k)}$ will be assigned as the greater value between $v_i^{(k)}$ and the $\rho_i^{(k)}$, which may be either the number of virtual circuits used in a parent node or the number of virtual circuits used in node *i*. The routing tree table for a node *i* using dynamic trees is shown in Table 4.2.

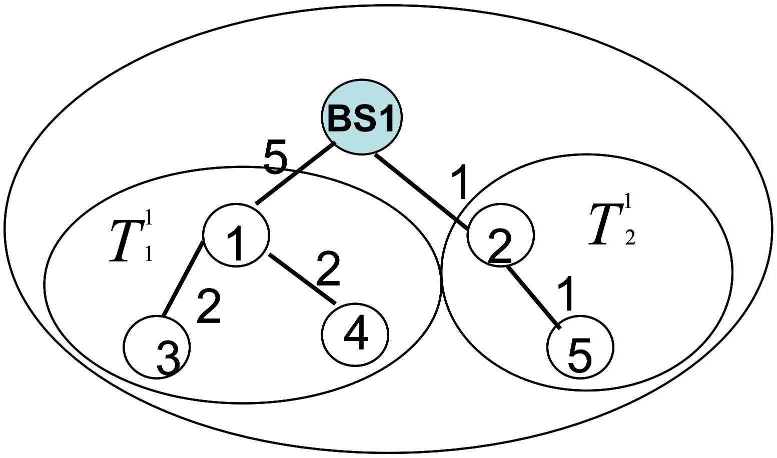


Figure 4.9a: Two sub-trees rooted at TLNs 1 and 2 with their virtual circuit connections to the BS.

k	Tree ID
$D_i^{(k)}$	The depth of node I in tree k
$P_i^{(k)}$	The parent node in tree k
$\rho_i^{(k)}$	Largest number of virtual circuits along path to root of tree k
$v_i^{(k)}$	The number of virtual circuits used by node i to the root of tree k

Table 4.2: Routing tree table of node i for dynamic trees

An example of the use of dynamic trees with the sub-trees is depicted in Figure 4.9b. Upon reconfiguring, Node 4 will join T_2^1 rather than T_1^1 since the former has more virtual circuit capacity available to the BS. This also relieves the sub-tree rooted at Node 1. The resulting reconfigured routing sub-trees are shown in Figure 4.9b. The values of Node 4's routing tree table before and after reconfiguration are shown in Tables 4.3 and 4.4 respectively. Node 4 will continue to use its virtual circuit connections through Node 1 until they terminate. All subsequent virtual circuit connections from Node 4, at least until the next reconfiguration, will be routed through Node 2.

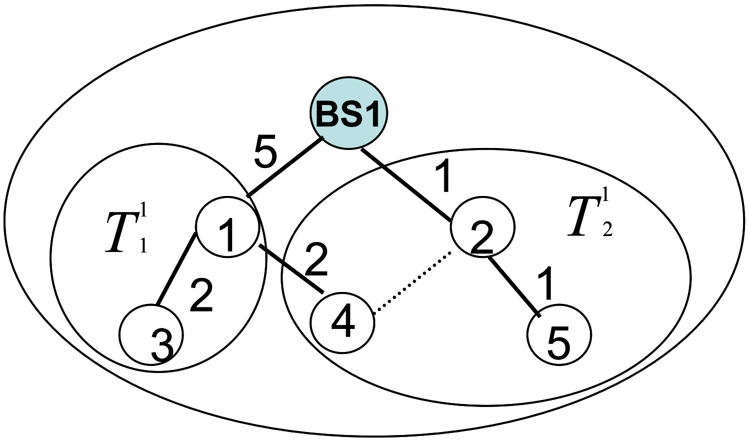


Figure 4.9b: Reconfigured sub-trees with their virtual circuit connections to the BS. New virtual circuit connections from relay Node 4 will go through Node 2.

k	Tree ID	**1**
$D_4^{(1)}$	The depth of node 4 in tree k	**2**
$P_4^{(1)}$	The parent node in tree k	**1**
$\rho_4^{(1)}$	Largest number of virtual circuits along path to root of tree k	**5**
$v_4^{(1)}$	The number of virtual circuits used by node 4 to the root of tree k	**2**

Table 4.3: Node 4’s routing tree table before reconfiguration.

k	Tree ID	**1**
$D_4^{(1)}$	The depth of node 4 in tree k	**2**
$P_4^{(1)}$	The parent node in tree k	**2**
$\rho_4^{(1)}$	Largest number of virtual circuits along path to root of tree k	**1**
$v_4^{(1)}$	The number of virtual circuits used by node 4 to the root of tree k	**2**

Table 4.4: Node 4's routing tree after reconfiguration.

Since dynamic trees more efficiently utilize available channel capacity, the blocking rate has the potential to be much lower than that of static trees. In fact, with dynamic trees, any node in the system may be able to join a tree with any of the top level nodes in the system. Therefore, the lower bound for the blocking rate with dynamic trees is simply the Erlang B formula applied to the aggregated group of last hop nodes. For nodes that join only one tree, the tree corresponding to a TLN to their home base station, the lower bound on the block rate will be B(qm,L), where m is the number of virtual circuits per TLN, q is the number of TLNs in the cell and L is the amount of traffic in the cell. For a cell cluster, the lower bound on the blocking rate for dynamic tress will be B(Cqm,CL), where C is the number of cells in the cluster.

4.6.1 Dynamic Reconfiguration of Routing Sub-Trees

Besides node failure and the addition of new routing nodes which are addressed in [8], there are two other cases when dynamic reconfiguration of routing trees is necessary:

1) When a node in the system reaches its full virtual circuit capacity and must be removed from the routing topology.
2) When a formerly fully loaded node again has available capacity and can be included back into the routing topology.

At system startup, a routing tree is created from the base station extending outward hop by hop according to the protocol described in [8]. This tree construction protocol is also used every time a routing sub-tree must be configured later on, if that becomes the case. The routing trees remain static until the virtual circuit capacity of a node is fully utilized. At that time, while the fully loaded node's active virtual circuit connections will continue to operate until they terminate voluntarily; the node will be removed from the routing topology for future virtual circuit requests since any requests to it will be blocked. Once this node is removed from the routing topology by having a null entry for its parent node then all of its child nodes must be attached to another parent node. If a TLN's active virtual circuit capacity is fully utilized then all of the descendant nodes in its sub-tree will utilize a different TLN's sub-tree to that particular base station.

When a node with fully loaded virtual circuit capacity is relieved of load and has available capacity, it is added back into the routing topology. It is attached to a parent node in a sub-tree within the system and also becomes a candidate to be a parent node for other nodes within its transmission range.

4.6.2 Problems with Dynamic Trees

One issue that decreases the effectiveness of dynamic trees is that the bottleneck may not necessarily be the TLN. Here, at least temporarily, a bottleneck may very well be an intermediate node. Figures 4.10a and 4.10b show a scenario after the creation of a new dynamic tree where the most loaded node is not a TLN.

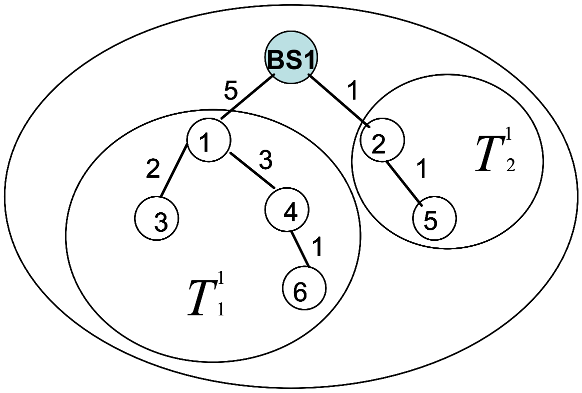


Figure 4.10a: Virtual Circuits in sub-trees 1 and 2

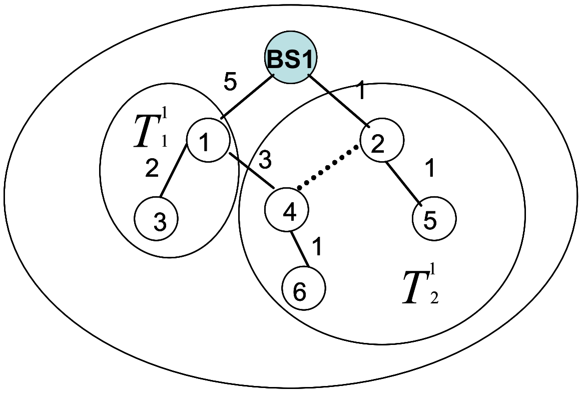


Figure 4.10b: Reconfigured Virtual Circuit Paths

In Figure 4.10a, Node 4 and Node 6 are in a sub-tree that is limited by the 5 virtual circuit capacity at the routing tree located at TLN Node 1, T_1^1. When the tree is reconfigured, Node 4 and Node 6 join the sub-tree rooted at Node 2, T_2^1. Upon reconfiguration, although the TLN Node 2, has only one channel being used for virtual circuits, Node 6's parent node, Node 4, is using 3 channels and thus the bottleneck with respect to Node 6 is Node 4. As time goes on, however, and as additional traffic is routed through the new sub-tree, the TLN will tend to become the bottleneck once again.

Another reason that dynamic trees may not achieve optimum levels of performance is related to physical topology: areas in the network where there is a lower density of nodes will cause fewer routing alternatives to be available to some nodes. Thus, when reconfiguration happens, a node might not be able to attach to the routing tree of a less loaded node. An example of this scenario is the physical topology shown in Figure 4.11. Here, two clusters of nodes, one in the vicinity of BS1 and another in the vicinity of BS2 cannot communicate with each other without passing through Node 3. Thus, Node 3 is the "bridge" node between the two clusters of nodes and no matter how a routing topology may be constructed with this physical topology, any traffic going from Nodes 1 and 2 to Nodes 4 and 5 must be routed through Node 3. Thus, subject to physical topology issues such as this one, a bottleneck may very well be in the middle of the network and not at the TLN.

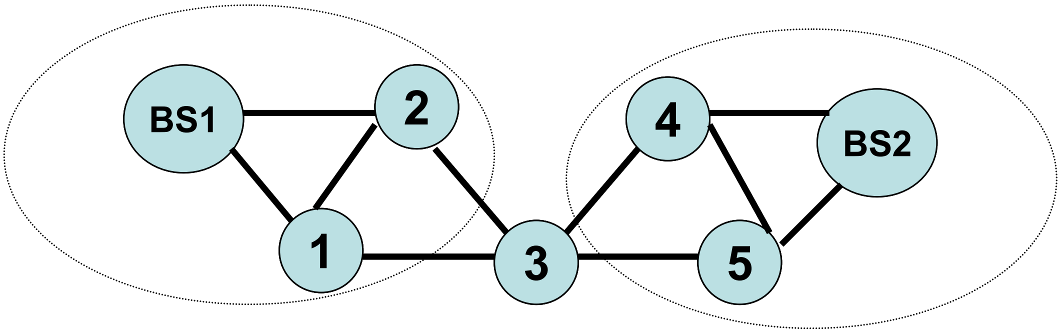


Figure 4.11: Topology-based reason for non-TLN bottleneck.

4.7 Best Case Scenario Comparisons

A summary of the lower bounds on blocking rate are in Table 4.5 below.

Scenario	Blocking Rate Lower Bound
Single tree static: Static trees in one cell	B(m,L/q)
Multiple tree static: Static trees in a cell cluster	B(Cm,CL/q)
Single tree dynamic: Dynamic trees in one cell	B(qm,L)
Multiple tree dynamic: Dynamic trees in a cell cluster	B(Cqm,CL)

Table 4.5: Lower bound of blocking rate for 4 scenarios

With:

m = Number of available virtual circuits in each TLN

L = Traffic load per cell

q = Number of Top Level Nodes (TLNs) one hop from BS

C = Number of cells in cluster

A graph of the theoretical lower bounds in Table 4.5 with parameters used the simulation to be discussed later is shown in Figure 4.12. Here, m = 5 for the virtual circuit capacity of each node, L = load per cell in Erlangs, and q = 6 since there are six sectors per cell. C = 7 for the 7-cell cluster case, and C = 19 for the 19-cell cluster case. There are six scenarios:

1) Single tree static: B(m,L/q) = B(5,L/6)
2) Single tree dynamic: B(qm,L) = B(30,L)
3) Multiple tree static, 7-cell cluster case: B(Cm,CL/q) = B(35,7L/6)
4) Multiple tree dynamic, 7-cell cluster case: B(Cqm,CL) = B(210,7L)
5) Multiple tree static, 19-cell cluster case: B(Cm,CL/q) = B(95,19L/6)
6) Multiple tree dynamic, 19-cell cluster case: B(Cqm,CL) = B(570,19L)

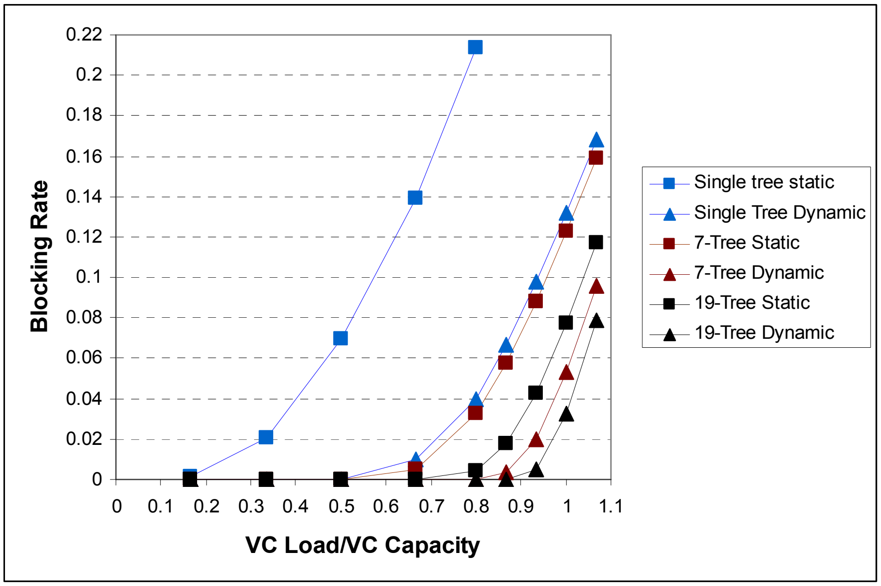


Figure 4.12 Graph of lower bound blocking probabilities

4.8 Simulation Study

We generated experimental results through simulation of our network architecture with the use of the virtual circuit protocols described previously. The goal of the simulations is to demonstrate the feasibility of our approach, to evaluate and verify the performance of our scheme, and to learn more about the parameters that affect performance in this context.

In order to compare our results to the theoretical baselines, our simulations use the four basic scenarios summarized in Table 4.5. They are,

1. Single tree static case: A multi-hop cellular network with multiple base stations that provides only one routing tree per node. In this case, virtual circuits are always routed along the tree to the base station in the node's home cell.

2. Multiple tree static case: A multi-hop cellular network with multiple base stations that provides several routing trees to each node, a routing tree to every base station in the cell cluster. Here, routing trees extend beyond cell boundaries to cover an entire cell cluster. Virtual circuits may be routed through either the home base station routing tree or a routing tree belonging to a base station in a neighboring cell. Two specific cases of the multiple tree static scenario are considered:

2a. 7-Cell Cluster: In this case, a cluster is comprised of 7 adjacent hexagonal cells. Each node in the cluster has 7 routing trees, one to each base station in the cell cluster.

2b. 19-Cell Cluster: Here, a cluster is comprised of 19 adjacent hexagonal cells. Each node in the cluster has 19 routing trees, one to each base station in the cell cluster.

3. Single tree dynamic case: The same scenario as #1 except that trees dynamically reconfigure when one node utilizes its maximum virtual circuit capacity. Each node is part of a single routing tree, the trees to the home base station. A node may be temporarily removed from the routing topology if it runs out of available virtual circuit capacity or its route to the base station may change if a parent node has no available capacity.

4. Multiple tree dynamic case: The same as scenario #2 above except that trees dynamically reconfigure when one node utilizes it maximum virtual circuit capacity. Each node is part of multiple routing trees, the trees to the base stations in the cell cluster. A node may be temporarily removed from the routing topology if it is out of virtual circuit capacity or its route to a base station may change if a parent node has no available capacity. As in #2, two specific cases of the multiple tree dynamic scenario were considered:

4a. 7-Cell Cluster: Same routing topology as 2a except that it uses dynamic trees.

4b. 19-Cell Cluster: Same routing topology as 2b except that it uses dynamic trees.

4.8.1 Topology Generation

In order to produce a sufficient amount of data that can be utilized to make statistically meaningful conclusions, 30 different random topologies were generated and used in the simulation. Nineteen base stations were placed in fixed positions within the network as well as 6 TLNs per base station. Figure 4.13 depicts 7 hexagonal cells with a base station at the center and the fixed placement of 6 TLNs per cell. Other nodes were placed randomly in the network. A random number of nodes for each topology was generated for each topology, ranging between a sparse topology with 196 nodes in the whole network to a more dense topology with 475 nodes. The nodes were also randomly placed according to the following steps:

1. Random assignment of cell number: a uniformly distributed random number between 1 and 19.
2. Random assignment of sector number within the cell: a uniformly distributed random number between 1 and 6.
3. Random assignment of a value of Θ, uniformly distributed between 0 and π/3. Θ is the measure of the angle from the edge of the triangular sector where the node will be placed.

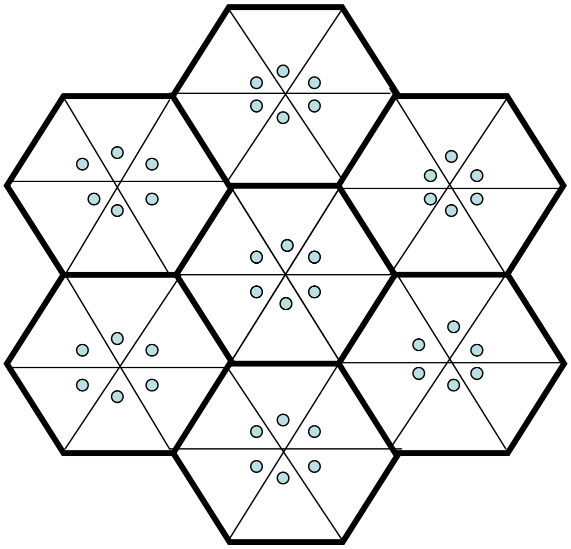


Figure 4.13: Fixed placement of cells, base stations and TLNs.

The distance from the center of the cell, *d*, is randomly assigned. This distance is the square root of a uniformly distributed random number, *z*, between 0 and 1 times the radius of the cell: $d = \mathrm{R}\sqrt{z}$. Points that end up outside the hexagon are discarded.

A further requirement on the node placement in the system is that all of the nodes must be part of a connected graph, so that there are no isolated nodes. Thus, every node must be within range of at least one other node in the system. For our system, base stations were spaced at least 175 units of distance from each other and the range for nodes in the system was 50 units of distance.

4.8.2 Traffic Model

The simulations utilized a Poisson arrival process of calls to each routing node with each node receiving the same average load. The holding time of a call is exponentially distributed with a mean of 180 seconds. With an Erlang defined as the (call arrival rate) × (holding time per call), the traffic intensity was varied from an average of 25% of the virtual circuit capacity of each cell to just over the virtual circuit capacity of a to determine the Erlang B blocking graphs for each of the scenarios. Ten thousand simulated seconds were run with each traffic increment and each topology.

4.9 Simulation Results

Simulations were performed with each of the six scenarios (1, 2a, 2b, 3, 4a, 4b) for each of the 30 topologies generated. The simulation results for each scenario are graphed and compared to the lower bounds of performance according to the Erlang B formula. To enable a more efficient presentation of the data, the graphs are grouped according to the number of routing trees per node, with 4 sets of points in each graph:

1) The blocking rates for the static tree approach.

2) The blocking rates for the dynamic tree approach.

3) The lower bounds on the blocking rate calculated from the Erlang B formula in Table 4.5 for static trees.

4) The lower bounds on the blocking rate calculated from the Erlang B formula in Table 4.5 for dynamic trees.

The graphs in Figures 4.14a, b, and c show the average blocking rate on the y-axis and traffic intensity in Erlangs as a fraction of virtual circuit capacity in the system on the x-axis. As we anticipated the best performance came from the 19-tree cluster, which provided superior load balancing qualities and thus a lower blocking probability. The single tree routing topology began to experience significant blocking at traffic intensity around 40% of the total virtual capacity, the 7-tree cluster began to experience significant blocking at just above 60% and the 19-tree cluster experienced significant blocking at over 75%. The single tree configuration, whereby virtual circuits are created on routing trees belonging to their home base station, allows for no load balancing and when capacity is used up at a TLN, then calls begin to be blocked. For the multiple tree approaches, whereby calls can be routed through any of several routing trees, system resources are utilized more efficiently by spreading traffic through the system more evenly, accomplishing better load balancing. When one tree may not have capacity due to a fully utilized TLN, each descendant node has several other routing options to go through other trees to another base station in the system.

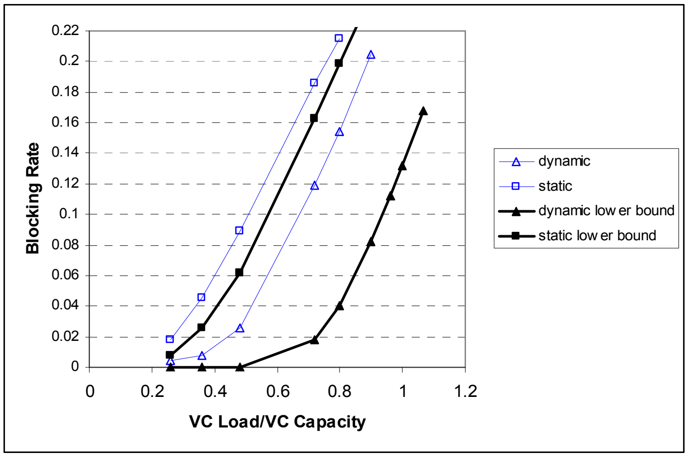


Figure 4.14a: Blocking Rates for Single Tree

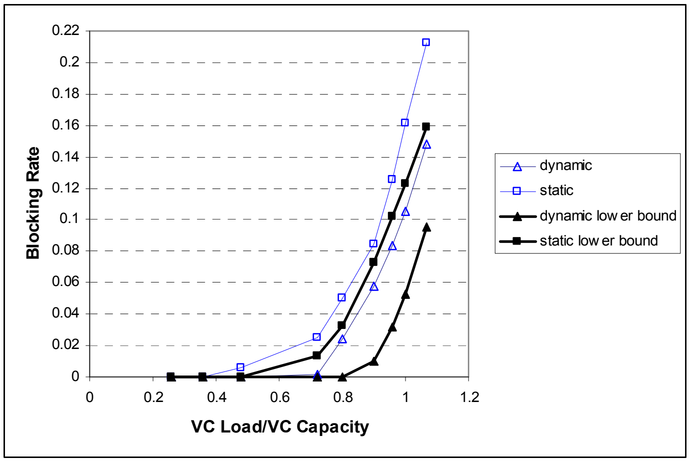


Figure 4.14b Blocking Rates for 7 Trees

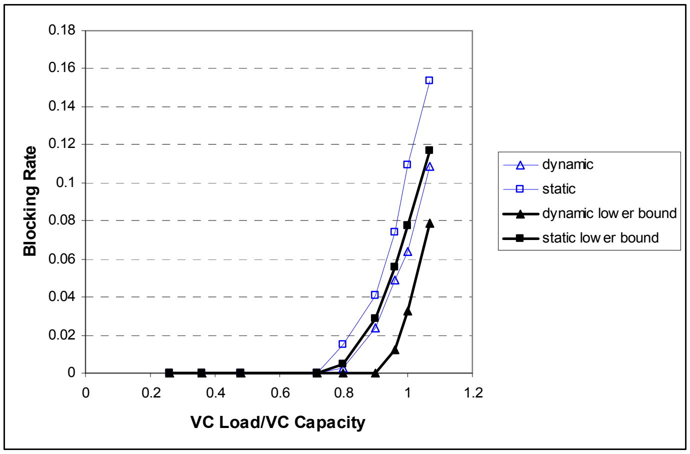


Figure 4.14c Blocking Rate for 19 Trees

Dynamic trees in each scenario did indeed improve system performance. They clearly outperform static trees and move the performance toward the lower bound blocking probabilities calculated from Table 4.5.

4.9.1 Parameters Related to System Performance

A critical factor that influences system performance for static routing trees in every scenario is how the routing trees are created. If there is a large inequity in the sizes of the routing sub-trees generated, there is a negative impact on system performance under the assumption that every node presents the same average load to the system.

For dynamic trees, the largest hindrance to system performance is related to physical topology issues mentioned in Section 4.6.2. The foremost of these is when a bottleneck is caused due to a low density of nodes in some part of the network. A node's parent and child may be the only nodes within its two-way communication range. When such a node's parent has used all of its virtual circuit capacity, there is no alternative for it to attach to a less loaded sub-tree. A scatter-plot of the number of nodes in the network versus blocking rate at the highest intensity of traffic simulated is shown in Figure 4.15. Here the topologies with the lower number of nodes tend to have mildly higher blocking rates, indicating that a sparse physical topology may lead to more non-TLN bottlenecks.

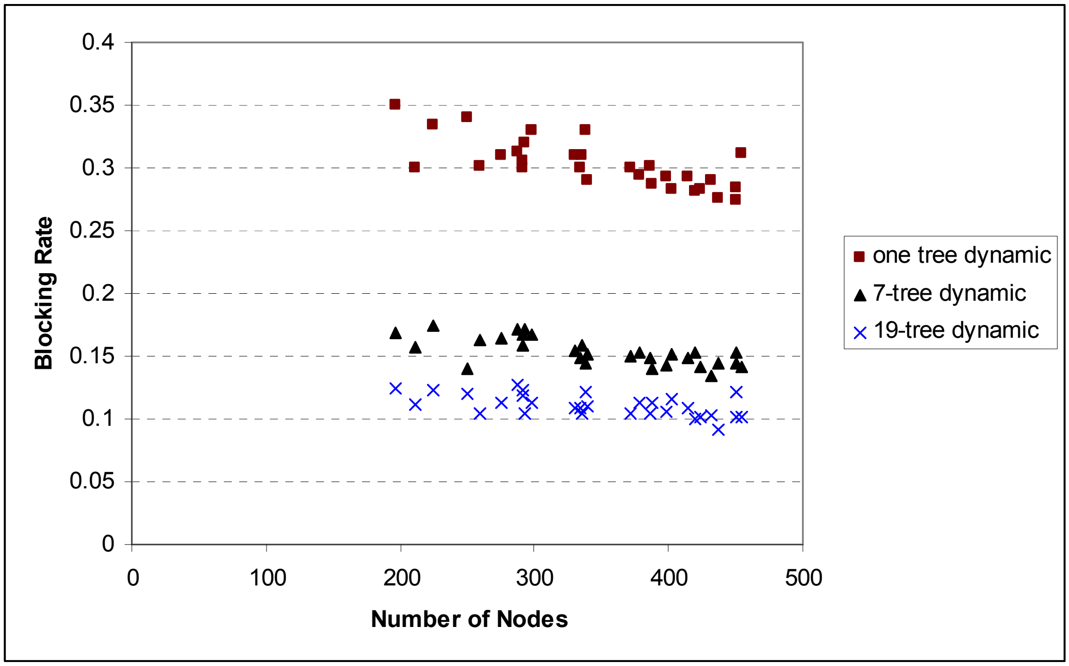


Figure 4.15: Number of nodes in network versus blocking rate at highest traffic intensity.

4.10 Conclusion

This study has introduced architectures and protocols to enable virtual circuits in an integrated multi-hop cellular network. With the main capacity constraint being the TLN, we use baseline Erlang B formulas to obtain the lower bounds of the blocking rates and compared them to experimental results. We have also demonstrated that the LSSON architecture, with its routing trees extended to a cell cluster, has good load balancing properties which enable cell clusters to share traffic load with each other and balance traffic demands among themselves. We have also found that the ability of routing trees to handle traffic can be optimized by maintaining equity in the size of the sub-trees generated that are rooted by the TLN.

We have described and demonstrated the feasibility of a dynamic routing tree approach which improves on the blocking rate of static routing trees by reconfiguring routing trees so that nodes can route traffic to the least loaded sub-trees. We have also developed a lower bound on blocking rates for the dynamic trees and listed factors that prevent dynamic routing tree scenarios from attaining optimum blocking rates.

Chapter 5

LSSON for Real-Time Vehicle Guidance

5.1 Introduction

In this chapter, we present a large-scale vehicle routing architecture and the associated protocols that aim at delivering real time vehicle guidance. By leveraging advanced wireless multi-hop network technologies we have devised a means of gathering and communicating real-time vehicle data and accordingly compute up-to-the-second link weights between roadside sensors for real-time vehicle guidance. In our system architecture, communication functions and vehicle route processing are separated into distinct subsystems, providing ubiquitous communication through a roadside mesh-like sensor network and a design framework for low-complexity real-time vehicle routing table computation. An analysis of the bandwidth requirements associated with the proposed architecture and communication protocols demonstrates the feasibility of an automated real-time vehicle routing system. The use of roadside sensor location as entries in vehicle routing tables minimizes the processing requirement within the vehicle. Vehicle routing complexity is drastically reduced by using an on-demand routing strategy coupled with a separation of local routing and long-distance routing. Finally, we present a hierarchical address aggregation method for long-distance routing.

A distinction is made in vehicle location and navigation systems between route planning and route guidance [86]. While Route planning is a static one-time only process that helps drivers plan a route prior to their journey, route guidance dynamically guides a driver to the destination in real-time during his or her trip. En-route guidance can be done either by an in-vehicle system or by a traffic management system that oversees a large area. In the former, vehicles do their own route guidance calculations by using an on-board computer and relevant highway network data available to it, applying heuristic search algorithms. While this distributes the computational load needed for routing each vehicle, it may not optimize the traffic flow for the entire network [87]. In the latter, a traffic management center with powerful computational capabilities periodically sends directions and answers queries submitted by vehicles linked to it. This approach gives more timely and accurate system-wide information at the cost of a greater concentration of computational power and higher bandwidth requirements in the supporting communications network. With continued improvement computational power and an associated improvement data network rates over time at lower costs, it is increasingly possible to attain the ideal of an integrated traffic management system that supports more optimal routing of vehicles throughout the highway network. In this research, we focus on a system that provides managed real-time turn-by-turn directions to the driver at every decision point on the journey utilizing a dense roadside sensor communications network and efficient route selection algorithms.

5.1.1 The challenge of real-time vehicle guidance

The objective of a real-time vehicle route guidance system is to provide each driver with the optimal real-time turn-by-turn directions based on system-wide traffic information [88]. The lack of system-wide, accurate, up to the second data is one serious challenge to providing optimal routes. The most common travel time data collection method in urban environments today is pavement loop detectors whereby travel times can be inferred from either single or double-loop detector arrangements. An example of this is the Illinois Department of Transportation's Traffic Systems Center (TSC) which receives its data every minute from loop detectors embedded every ½ mile in Chicagoland expressways to calculate travel times [89]. While this gives good estimates of travel times on expressways, the thousands of miles of other major roadways off of the expressways in Chicago area are not included in the system. Probe vehicles have also proven useful in generating real-time travel times, but the sample sizes needed to gather complete system-wide information vary greatly and depend on traffic patterns, the nature of the roadway network and many other variables [90]. License plate matching techniques that collect travel times between checkpoints using either video cameras or automatic license plate recognition (LPR) hold great promise, but are not yet fully developed. In the future, a fusion of available data gathering techniques will provide more complete and accurate traffic network link times [91]. Our real-time vehicle guidance solution is fully capable of being integrated with many of the vehicle recognition and data gathering techniques that are currently being developed.

5.1.2 Communications Infrastructure for Intelligent Transportation Systems

To extend the reach of existing communication networks there has been ongoing research into wireless mesh networks (WMNs), which offer a low cost infrastructure, increased coverage, added redundancy, and improved reliability [92]. They use wireless nodes that act as repeaters to transmit data from nearby nodes to peers that are too far away to reach, resulting in a network that can span a large distance, especially over rough or difficult terrain and scenarios with significant obstructions (trees or high rise buildings) [93]. Consequently, WMNs have continued to draw more general commercial and government interest, with some public transportation companies, government agencies, and research organizations looking for viable solutions for intelligent transportation systems [94]. A wireless mesh network technology provided by MeshNetworks Inc. was recently deployed to support the Portsmouth Real-Time Travel Information System (PORTAL), which allows anyone to display at more than 40 locations in the city real-time transportation service information, such as where a bus is at that time and when it is scheduled to arrive. In our proposal, we discuss the use of a very dense roadside mesh network for Intelligent Transportation Systems (ITS) to continually collect traffic data from every small segment of city streets and highways in urban areas to facilitate dynamic vehicle guidance.

5.1.3 Hierarchical routing of vehicular traffic

The complexity involved in processing the tremendous volume of vehicle data for all small road segments of a large area and in timely computing the optimum vehicle routes based on those data is another hindrance to real-time route guidance. The computational complexity for an all-pairs Dijkstra Algorithm is $O(n^2 \log_2(n))$ for roadway networks [78]. Using hierarchical routing in roadway networks may greatly reduce such computational complexity. Links can be classified according to their throughput, with high speed roads such as main arterials and interstate highways pushed up to a higher tier and lower speed links such as local avenues residing at a lower tier [95]. Alternatively a road network can be partitioned into smaller fragments and organized in a hierarchical manner by pushing up border nodes [96]. In our hierarchical scheme, we distinguish between "local" routing and "non-local" routing in a novel way to reduce routing complexity to manageable levels.

5.2 A two sub-system architecture

The main objectives and key features of our system are: low-cost sensors, minimal vehicle functions, and low-complexity vehicle route computation. To achieve these objectives, we present a system architecture that separates communication functions from vehicle route computation functions. As shown in Figure 5.1, a wireless sensor network based on our LSSON design constitutes the communication subsystem (CSS) and is responsible for collecting and forwarding vehicle data to the vehicle routing subsystem (VRS) and for distributing optimum routing information to the vehicles from the VRS. The VRS is responsible for processing the vehicle data and generating optimum routing tables *for every sensor node in the system.*

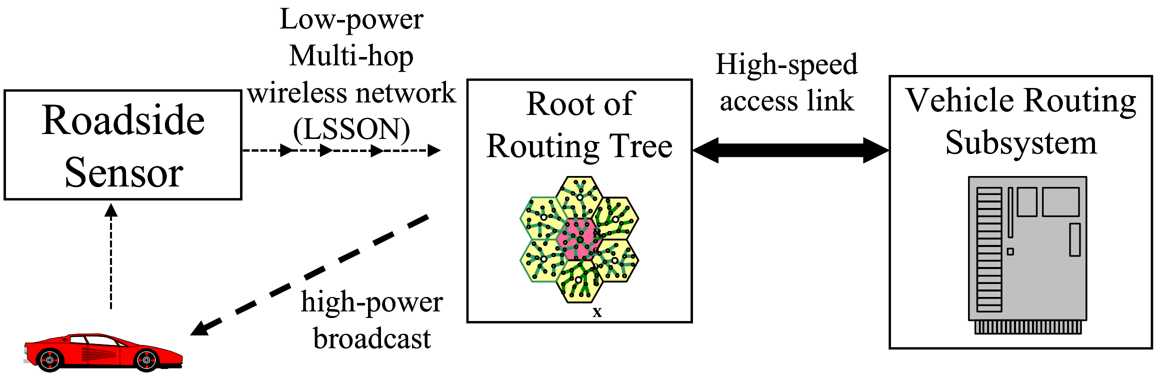


Figure 5.1 A sensor network based vehicle routing system consisting of a large-scale sensor network and communication subsystem (CSS) and a vehicle routing subsystem (VRS). The sensor network provides the communication infrastructure and the VRS does all the computations related to optimum vehicle routing.

5.2.1 The Communications Subsystem (CSS)

The CSS is made up of a large number of low-power densely populated roadside sensors throughout the system. In particular, there must be at least one sensor between two neighboring intersections. The sensors are self-organized into a multiple-routing tree hierarchical routing network [8,9] as described in Chapter 3 which allows data packets carrying vehicle information to be routed efficiently to a base station or "root" node. Roots are strategically deployed throughout the sensor network and each of them is capable of broadcasting messages to sensors and vehicles within its coverage area and like base stations, the coverage areas of neighboring sensors must overlap in order to ensure complete coverage. We shall call the coverage area of a root a "cell" and the union of the cells of a root and all its neighboring roots a "cell cluster" associated with this root, which is analogous to the same term used in cellular systems. To illustrate, Figure 5.2 shows a cell cluster associated with Root X. A root supervises the self-organizing process within its routing tree through which sensors within the tree forward data messages upstream to this root following the tree. Sensor nodes may be placed along the roadway such as on signs or lamp posts.

Based on the self-organizing protocol developed by Hester and Lee [8], the routing tree owned by a root can in principle cover the entire network [9] thus allowing every sensor in the network to send data to any root in the network. However, with a huge mesh sensor network, the routing complexity involved in every sensor node would be tremendous and sensor simplicity would have to be compromised if every routing tree were allowed to cover the entire network. To maintain sensor simplicity, we limit the reach of each routing tree to the cell cluster associated with the root of tree as described in Section 1.3.1. We note that such a routing tree constraint is readily achieved by a modified self-organizing protocol. As a result, a sensor can belong to a limited number of routing trees where the number is essentially the number of cells in a cell cluster. Each root maintains its own routing tree, coordinates medium access, initiates periodic maintenance tasks for its tree, and broadcasts real-time vehicle routing tables associated with every sensor within its cell. As a result, the roots free the sensors from routing related processing burdens and as such, the sensors require little processing power and memory. We note that the roots essentially serve as the interface between the CSS and the VRS and as such they can be considered as belonging to both.

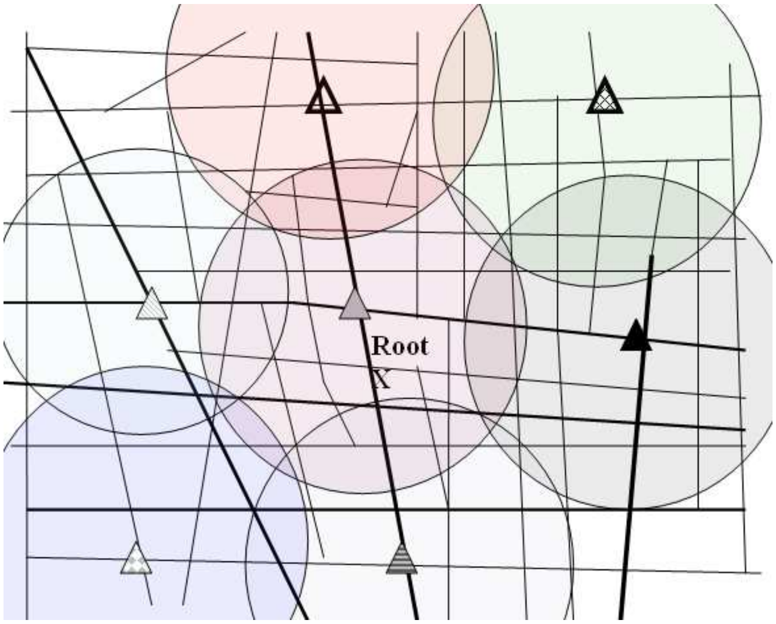


Figure 5.2 A cell cluster centered at Root X. The range of each root (triangles in diagram) covers areas in respective circles above. Roots broadcast routing tables to passing vehicles and maintain the routing trees for sensors. In the above figure is a cluster with seven cells, similar to a cellular system.

We define a sensor as a communication node with a very small range, and is responsible for only the following functions:

1. Continually broadcasting a presence signal that polls vehicles passing by.
2. Receiving data packets from vehicles responding to its broadcast.
3. Inserting its ID and a time stamp in each vehicle packet it receives and forwarding it to the parent node with respect to a selected tree (root).
4. Forwarding packets from other sensor nodes to its parent node with respect to the selected tree specified in the packet.

While passing a sensor, a vehicle transmits a small packet containing its vehicle ID and its intended destination ID which is encoded geographically (to be discussed later). Upon reception of a vehicle packet, the sensor inserts its own ID and a timestamp in the packet and forwards it to its parent node – a neighboring sensor, en route to a root. The packet formats of the poll, vehicle data, and what is forwarded from a sensor are shown in Figure 5.3.

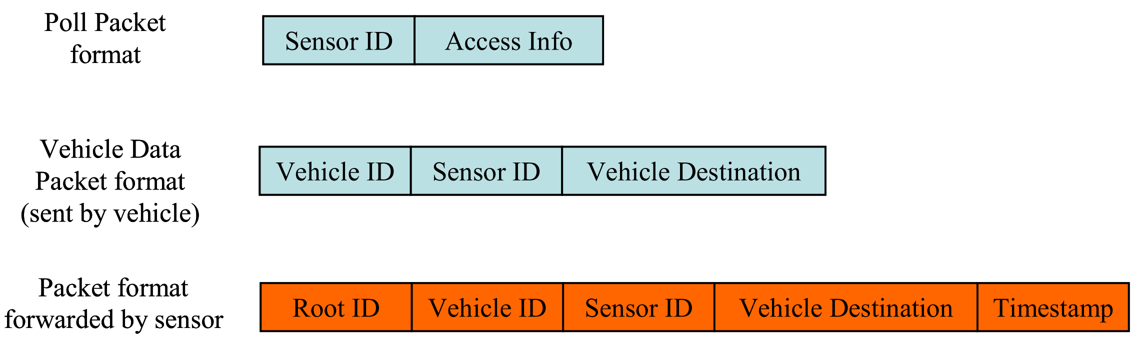


Figure 5:3: CSS upstream packet formats

The CSS supports both up-stream and down-stream communications. Up-stream communication involves a single-hop vehicle-to-sensor protocol, a multi-hop sensor-to-root protocol, and a high-bandwidth root-to-VRS protocol. The first two have been sketched above. With regard to the third, we shall assume that roots are capable of communicating within the VRS using a separate communication infrastructure in which roots have little constraint in power and bandwidth. Through this root-level infrastructure, the vehicle data are sent to the VRS with little delay. Downstream communication involves distributing the optimum vehicle routing tables generated by the VRS to the roots via the root-level infrastructure and broadcasting the relevant vehicle routing tables to the vehicles. A control channel, dubbed Channel *R* is used by the roots for broadcasting vehicle routing tables to vehicles within their respective cells. We note that a root is responsible for only routing tables associated with source locations (sensors) within its cell and the sensors do not maintain these routing tables associated with themselves.

5.2.2 The Vehicle Routing Subsystem (VRS)

A distinctive feature of the overall system architecture is that it separates communication functions from vehicle route processing functions. The VRS obtains data from vehicles and accordingly calculates link weights within the roadway network. Then, based on a distance metric, it incorporates travel times and optionally other criteria such as system load balancing and special policies assigned to roads or specified by drivers, to generate vehicle routing tables for each selected source location in the system. The number of entries in a routing table equals the number of selected destination locations from the source location and each entry specifies the "optimum next sensor" that allows the vehicle to decide upon its action at the upcoming intersection.

An example of the function of the VRS can be seen from a roadway network shown in Figure 5.4. The encircled cell contains many roadside sensor locations, including the sensors numbered 1 through 7. Based on passing vehicle data, link weights are calculated among all of the source-destination sensor locations and tracked in a link weight matrix kept by the VRS. A sample link weight matrix based on this example is shown in Figure 5.5. The resulting vehicle routing table that is broadcast by the root provides a source sensor id, a destination sensor id and the "next hop" sensor as shown in Figure 5.6. From this table and using Figure 5.4, at Sensor 1 and traveling eastbound toward Sensor 7, would be directed to Sensor 4 as the next hop. This would translate into a "turn right" instruction at the next intersection for the vehicle at Sensor 1.

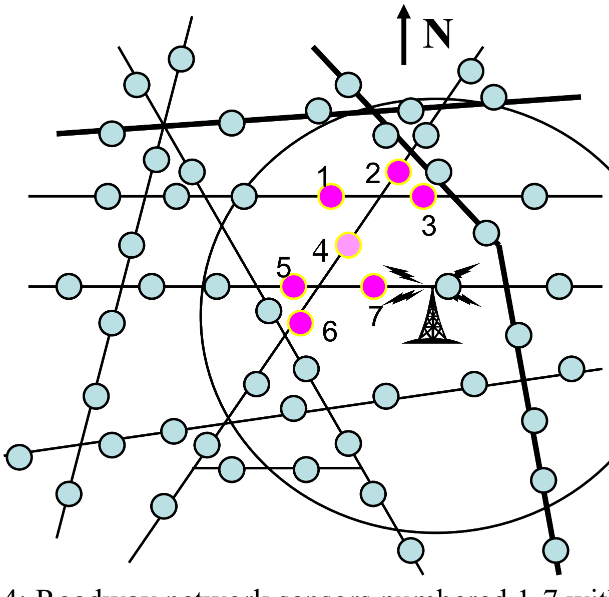


Figure 5.4: Roadway network sensors numbered 1-7 within a circular cell.

$$\begin{pmatrix} 0 & 5 & 3 & 4 & 0 & 0 & 0 \\ 4 & 0 & 3 & 4 & 0 & 0 & 0 \\ 3 & 1 & 0 & 4 & 0 & 0 & 0 \\ 3 & 4 & 5 & 0 & 3 & 4 & 5 \\ 0 & 0 & 0 & 3 & 0 & 3 & 6 \\ 0 & 0 & 0 & 3 & 6 & 0 & 7 \\ 0 & 0 & 0 & 7 & 6 & 8 & 0 \end{pmatrix}$$

Figure 5.5: Sample link weight matrix generated by VRS based on passing vehicle data.

Node	Destination	Next	Weight
1	2	2	5
1	3	3	3
1	4	4	4
1	5	4	7
1	6	4	8
1	7	4	9
.	.	.	.
.	.	.	.
.	.	.	.

Figure 5.6: Sample vehicle routing table for at vehicle at Sensor 1.

Computational challenges for this arise from the large size of the sensor networks, as is reflected by the number of sensors deployed, which is in the order of the number of edges of the topological graph where the vertices are the sensors and the edges are the street/highway paths between adjacent sensors. Without specifying the root-level communication infrastructure and protocol in this paper, our proposed system leaves open the VRS architecture. At one extreme, the VRS can be a single control center that receives all the vehicle data and is responsible for all the routing table computations and distributions on a timely basis. At the other extreme, the VRS can assume a fully distributed control architecture where the roots are lowest-level VRS nodes responsible for processing optimum vehicle routing for source-destination pairs within their respective cell clusters and where higher-level VRS nodes, if they exist, deal with vehicle routing over longer distances. This second approach enables the VRS to capitalize on parallel computing for optimum vehicle routing tables thus distributing the computational burden to a number of hierarchically structured VRS nodes. In this thesis, we discuss only the computational complexity that the VRS as a whole must deal with and a means of reducing it while leaving alone specific VRS architecture.

5.3 Vehicle Function

As a result of the system architecture presented, the required functions for a vehicle in the system are minimal:

1. It constantly listens to Channel S for the presence of roadside sensors.
2. It transmits a short packet containing its ID and its intended destination to a newly detected sensor (upon reception of the broadcast from the latter).
3. It listens to the broadcast on Channel R, selects the routing table associated with the last sensor it detected, and from the table the destination entry that captures its intended destination.

5.4 Bandwidth Requirements

To achieve real-time dynamic vehicle routing, a vehicle sending a packet to the CSS must receive the newest vehicle routing table containing its intended destination before it reaches the next decision point – the upcoming intersection. Let the roundtrip time from when a vehicle emits its packet *P* to a roadside sensor *i* until a routing table for sensor *i*, which incorporates *P* in its computation, is received by the vehicle may be denoted T_r. This value must be somewhat smaller than *T*, the time it takes the vehicle to travel from its current location to the next intersection. T_r is comprised of three components: the up-stream communications delay, t_u, the VRS processing delay of updating the routing tables, t_p, and the downstream communication delay, t_d. The real-time objective is met if

$$\boldsymbol{T} > \boldsymbol{T}_{\mathrm{r}} = \boldsymbol{t}_{\mathrm{u}} + \boldsymbol{t}_{\mathrm{p}} + \boldsymbol{t}_{\mathrm{d}} \qquad (1)$$

Among the three delay components, td is negligible due to the high bandwidth assumption on the root-level communication infrastructure. In this section we discuss the bandwidth requirement for the sensor network to contain tu. In the next section, we discuss complexity issues related to vehicle route computation and present means of reducing the complexities in order to contain tp.

It is readily shown that multi-hop transmission and propagation delays are negligible compared to *T* and, therefore, if queuing at the sensors can be constrained, upstream communication delay t_u will not pose a problem for the real-time objective (1). Since queuing can be contained by having sufficient bandwidth for sensors to forward messages, in the following we analyze a bandwidth requirement for the sensor network.

Macroscopic traffic flow theory relates traffic flow, traffic density and vehicle velocity. The fundamental relationship [97] is $Q=KV$, where Q = traffic flow, K = traffic density and V = average velocity of vehicles. From this relationship, traffic density and velocity are inversely proportional. We can also infer a τ-second rule which places an upper limit on traffic flow and thus vehicle arrival rate from the sensor's perspective. The Federal Highway Administration has pegged the maximum design capacity per lane of roadway in vehicles per hour as 2,200 [98].

This results in an upper limit of $0.61L$ vehicles per second on a roadway, where L is the number of lanes. Consider our model where sensors receive data from vehicles as they pass by. The data rate received by a sensor node is then related to the flow of vehicles on the roadway, Q. If the length of a data packet sent by a vehicle and forwarded by the sensor, including all the overheads, is P bits, then the maximum bit rate at each sensor is $R = 0.61LP$ in bits per second. This leads to the conclusion that bandwidth requirements for the roadside mesh network are not burdensome. For example, an 8-lane highway with vehicles transmitting 1500-bits packets to roadside sensors has a maximum data rate received by the sensor from passing vehicles R = 7.32 Kbps.

Consider a d-meter street section, the range of a sensor located in the middle of the section. Let:

N = number of vehicles inside the section (time-varying)

L = number of lanes

H = depth of routing tree in hops (a key sensor network parameter)

V = average vehicle speed inside the section in meters/sec (time-varying)

P = Length of packet sent by vehicle in bits to the sensor once polled (including overheads)

Then with increasing N, there is a smaller spacing between vehicles and thus a decreasing speed, V. This suggests that NV = constant, k, which is consistent with a τ-second rule, or 2-second vehicle spacing rule of thumb, as mentioned previously. This amounts to k = Ld/2, based on Little's Result, and gives a vehicle arrival rate of:

$$Q = L/\tau \quad \text{vehicles/sec} \qquad (2)$$

Using (2), for vehicles driving past a roadway sensor as shown in Figure 5.7, the maximum data traffic from the vehicles to the sensor is: $R = QP = LX / \tau$ bps. The bandwidth requirement for a sensor that is H hops from the base station is thus:

$$B = HLP / \tau \text{ bps} \qquad (3)$$

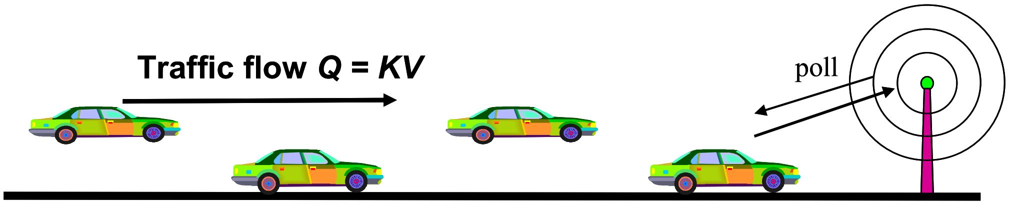


Figure 5.7: Bandwidth requirement of roadside sensor. With $H = 10$ hops, $L = 4$ lanes, $P = 2000$ bits, $\tau = 2$ seconds, the resulting bandwidth required is 40 Kbps.

The analysis of the bandwidth requirements of roadside sensors strongly suggests that the communication capacity of such a data network is not a major bottleneck of the system.

5.5 Managing Vehicle Routing Complexity

Once the communication delays are contained, the major factor that determines whether the real-time objective (1) can be met is the time it takes the VRS to process vehicle data and accordingly produce updated routing tables. The VRS uses vehicle data to calculate the travel times between adjacent sensors thereby maintaining link weights from every sensor node to each of its neighboring nodes. These link weights are the basic building blocks for constructing optimum vehicle routes for every source-destination of interest.

A conventional approach to estimating link weights is based on aggregate flow through an intersection [99]. However, these estimates may not be accurate due to traffic exiting or entering the roadway segment between the two end points. By comparing two packets from the same vehicle passing by two successive sensors, the travel time between these sensors can be accurately and frequently updated. Note that the VRS is able to derive the travel time between successive sensors because vehicle data packets contain vehicle ID, sensor ID, and sensor timestamps. The collection of the travel times between successive sensors across the entire system allows the VRS to establish and maintain an up-to-the-second link weight matrix for the network.

Note that the VRS must compute a routing table for every sensor node which represents the location of vehicles that sent their last vehicle data packet to this node. Using Dijkstra's algorithm to determine the optimum routing table, the complexity for each routing table will be O(|E| log n) [78], where |E| is the number of edges and |V| the number of vertices on the topological graph. Since edges exist only between adjacent sensors, the per-sensor complexity is g(n) = O(n log n) since |E| is of O(n), where

$$n=|V|=\text{number of sensors in the entire system} \tag{4}$$

In particular, for a grid street topology, $|E| \sim 3n$ and as a result, the overall computational complexity for the VRS to produce n routing tables is $f(n) = ng(n) = O(n^2 \log n)$ where n is given in (4). Since n is a huge number, such a complexity may pose a problem for the VRS. Furthermore, distributing large routing tables entails a lot of downstream bandwidth in VRS-to-root communication and more importantly, a lot of broadcast bandwidth in the root-to-vehicle communication.

In the following, we discuss means of reducing g(*n*), the per-sensor computational complexity, for the VRS and per-sensor communication complexity for the CSS.

In practice, at a given time the VRS needs to provide only optimum vehicle routing information to active vehicles, i.e., vehicles from which the VRS just received packets. Therefore, we may substantially reduce the processing complexity by exploiting "on-demand" routing, only generating new routing tables for those source locations (sensors) from which the VRS just received vehicle data and a table containing optimum next hops (next sensors) for only those destinations indicated in the vehicle data. In this manner, the total number of source-destination pairs that have to be dealt with by the VRS at any given time is no larger than the number of vehicles in the system at that time.

Although an on-demand routing strategy will greatly reduce the number and size of routing tables within a given time interval, it may not yield as much savings when it comes to the computational complexity involved in updating them. In the following, we present an approach analogous to internet routing that circumvents this very problem.

First of all, we separate two types of vehicle routing: intra-cluster routing and inter-cluster routing. The former is for vehicles whose destination is within the cell cluster of its current location, and the latter is for vehicles heading for distant destinations beyond their current cluster boundaries. Since a root owns a (message) routing tree that spans a cell cluster, it receives all the vehicle data from sensor locations within the cell cluster. As such, a root has all the roadway link weights within the cell cluster and can therefore compute all the possible intra-cluster routes. Specifically, a root is responsible for computing and broadcasting the vehicle routing tables for all the sensors within its cell and each of these tables covers demanded destinations throughout the cell cluster. Although the worst-case complexity for producing a routing table remains to be $g(n) == O(n^2 \log n)$, the number n is now no larger than the number of nodes in a cell cluster:

$$n = n_0 = \text{ number of sensors in a cell cluster}$$

To route a vehicle to a destination beyond the cell cluster associated with its current location, inter-cluster routing is used instead. To contain the complexity of inter-cluster route computation, we use a destination address aggregation approach. Such a hierarchical routing scheme necessitates the decomposition of a graph into a set of fragment graphs and a boundary graph which summarizes the fragment graphs. In one possible approach, links are classified according to their throughput, with high speed roads such as main arterials and interstate highways pushed up to a higher tier and lower speed links such as local avenues residing at a lower tier [100-103]. While this scheme can be used to calculate a minimum path cost rapidly, it does not guarantee an optimal shortest path. Another approach, which partitions a large graph into smaller graph fragments and organizes them by pushing up border nodes to a higher tier, can guarantee path optimality. One such proposal is HEPV (Hierarchical Encoded Path View) which partitions the road network into smaller fragments and organizes them in a hierarchical manner by pushing up border nodes [104]. Here, all-pair shortest paths are calculated within each fragment and all-pairs shortest paths are calculated among neighboring border nodes. In another similar concept, “Hub Indexing” is used to compute shortest paths between boundary nodes in each fragment and between boundary nodes and their interior nodes [105]. Another method, named “HiTi” (for Hierarchical MulTi) partitions a graph into smaller sub-graphs and pushes up pre-computed shortest paths between boundary nodes of each sub-graph [106]. It does not pre-compute shortest paths for intra-partition nodes but proposes calculating those on the fly. For all of these methods, partitioning and parallel computation reduce the complexity of an all-pairs shortest path algorithm on a large roadway graph from $O(n^2 \log_2(n))$ to $O(n^3/f^t)$ in each fragment where f is the number of partitions and t is the number of tiers in the hierarchy. Our approach uses the above principles related to geographic partitioning. This partitioning scheme is similar to the one mentioned in Chapter 3 and shown in Figure 3.6. However, the partitioning scheme in Chapter 3 is related to message routing while in this case it is applied to vehicle routing. Moreover, we note that while there are similarities between inter-cluster message routing in Chapter 3 and inter-cluster vehicle routing in this chapter, intra-cluster routing is very different. For intra-cluster message routing we used a routing tree; such a scheme is nonsensical for routing vehicles. Therefore, as we have described above, intra-cluster routing uses an optimum all pairs shortest path algorithm such as Dijkstra’s algorithm.

A major benefit of geographic partitioning is that shortest paths in each partition can then executed in parallel. In parallel computing, the execution time of a problem such as all-pairs shortest paths is determined by the execution of the slowest processor. The model for total execution time on parallel processors, each with a processing time of T_{indiv} is $T_E = T_{calc} + T_{comm}$

where T_{calc} is the time required to accomplish calculations and T_{comm} is the time needed for communication among the processors. If the $T_{calc} >> T_{indiv}$, then there is significant idle processing time and the partitioning is not very efficient [107]. The graph partitioning problem whereby a graph divided into "balanced" partitions is known to be NP-hard, and therefore the only way to cope is to use heuristics and to close to the optimal solution [108]. There are many partitioning algorithms for maps. One of these is a software package called METIS that divides graphs into *k* equally sized parts to minimize the number of edges that separate partitions [109]. Other algorithms divide the map into *k-D* trees which recursively bisect cells through their longest axis, so that an equal number of nodes lie in each sub-volume [110]. A simpler partitioning structure, quad-trees, are used to partition a two dimensional space by recursively subdividing it into four quadrants. The easiest way to partition a space is with a grid which is the approach that use.

A transportation network can be modeled with n nodes as an $n = m$ x m grid. It is then easy to partition this into equal sized sub-graphs that are smaller grids. Therefore, we can partition an n node m x m grid graph into $p = f$ x f fragments. In this case, each fragment will have n/p nodes.

The basic idea is to aggregate the remote destinations (from the vehicle's current location) based on the *M*-tier hierarchical network partitioning method described in Chapter 3 as shown here in Figure 5.8. The addressing scheme is exactly the same as the one shown in Chapter 3, Figure 3.6.

As a vehicle traverses the network, it progresses from one cell cluster to the next. Before it reaches the cell cluster in which its destination is situated, it follows a dynamic "inter-cluster route" where its destination is absorbed by a "super node" - a root representing a high-tier zone whose address matches the prefix of the destination address. As soon as the packet reaches the cell cluster of its destination, it switches to "intra-cluster route" thus following the intra-cluster routing table en route to its destination. All sensors in a cell use a common inter-cluster routing table where the degree of destination address aggregation is reflected in the address prefix length. We note that the "next root" indicated in the table is readily translated to "next hop" using the intra-cluster routing table because "next root" is within the cell cluster of the sensor with which this inter-cluster routing table is associated and the intra-cluster routing table contains all the source-destination pairs within the cluster. Therefore despite the fact that all the sensor locations in a cell share the same inter-cluster routing table and thus the same next root for a given remote destination, they may have different next hops for that same destination.

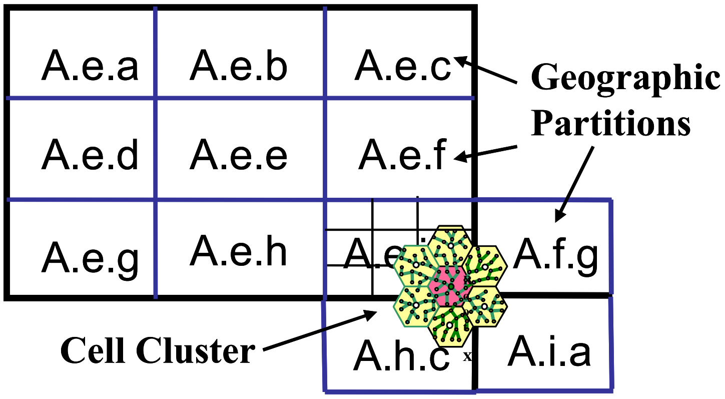


Figure 5:8: Hierarchical vehicle routing model: Cell clusters distribute precise vehicle routing information for destinations within the cluster. Longer distance routes use representative root information for geographic partitions.

Essentially, routing complexity reduction is achieved when the destinations are aggregated with degree of aggregation proportional to the Euclidean or L_∞ distance between the current location and the destination. This drastically reduces the number of possible destinations that a single routing table (for a sensor) has to deal with and accordingly the per-sensor computational complexity for maintaining inter-cluster routing information is $g(n) = O(n \log n)$ where

$n = \min \{(K\text{-}1)M$, number of roots (cells) in the system$\}$

We note that the underlying idea presented is analogous to Fisheye State Routing [111] (FSR) techniques such as Landmark Routing [112]. However, these existing techniques were used directly to compute (packet) routes, while we use the idea just to aggregate addresses based on which optimum vehicle routes are computed using link weight information maintained by the VRS. As a result, we are able to reduce the routing complexity and complexity associated with distributing routing information to a level that is mostly independent of the dimension of the system.

It is important to point out that the hierarchical zone structure is fixed and is independent of the wireless cell and cell cluster boundaries. The vehicle routing table broadcasted by the root on behalf of a sensor contains both intra-cluster and inter-cluster destinations where the former consists of complete addresses and the latter aggregated addresses. The inter-cluster entries essentially capture root-level shortest distances and in that sense they are sub-optimal. They indicate the suboptimal “next root” from this sensor location to the following selected destination zones where each of the selected zones must contain at least one root:

- All the tier-1 zones contained in the tier-2 zone that contains this root
- All the tier-2 zones surrounding the tier-2 zone that contains this root
- All the tier-3 zones surrounding the tier-3 zone that contains this root
- All the tier-M zones surrounding the tier-(M-1) zone that contains this root

In the event that no root exists in a Tier-k zone listed above, this zone is absorbed in the Tier-(k+1) zone that contains this zone, provided that a root exists in that Tier-(k+1) zone.

Chapter 6

MAC scheduling in a Simple Multi-hop Wireless Network

6.1 Overview

In any multi-hop communication network, medium access control (MAC) is an important and difficult issue because it dictates how the wireless spectrum is shared by the base station, relay station, and the mobile stations. This chapter examines the conditions under which there is an improvement in throughput by the effective use of relay nodes and presents a MAC scheduling solution for a simple multi-hop cellular network. We do so through the use of a centralized scheduling scheme where the BS chooses the best relay node for each SS/MS to communicate through in the uplink direction and properly allocates the TDMA data slots that are shared by all the active nodes in the cell. It also effectively manages the queues in the relay nodes so as to render queueing delay in the relay nodes a negligible part of overall delay for users. We show analytically that this approach can improve system throughput and demonstrate our results by simulation.

Relaying is widely considered as a viable means of improving capacity and coverage in wireless networks. The following benefits have been identified:

1. Coverage can be improved in areas with high shadowing. In places with significant obstructions, such as indoors and in built-up urban environments, better connectivity with the use of relay nodes can be offered to subscribers.
2. Coverage can be extended with relays beyond what one base station can provide. Furthermore, improved connectivity can be offered to subscribers on edge of cells at a relatively large distance from the base station.
3. Overall power transmission levels can be reduced in multi-hop networks compared to those of one hop. This enables subscriber stations to use less power in their transmission, increasing battery life.
4. Relay nodes can be deployed to provide temporary coverage where permanent coverage is not necessary. This can facilitate communication and connectivity during emergencies and special events.

6.1.1 Capacity Potential of Multi-hop Cellular Networks

In classical cellular systems, all subscriber stations (the abbreviations SS for subscriber station and MS for mobile station are used synonymously in this chapter) communicate directly with a base station (BS). However, when a SS is too far away from a BS or is in an area obstructed from the BS, communication at high data rates is impossible. Here, utilizing intermediate nodes to relay traffic is a means to improving capacity and performance in cellular networks. Relay nodes shorten the distances over which communication takes place, thus improving channel qualities. As a result, the use of relays has been the subject of much active research in recent years. In the seminal paper by Gupta and Kumar [15] they have shown that the maximum achievable throughput under optimal conditions is $1/\sqrt{n}$ times the transmission rate and as the number of nodes per unit area increases, the throughput decreases accordingly. In [113] Gasper and Vertterli used the same physical model as [15] and found that by using relays traffic capacity is improved to log(n) times the transmission rate. Other information theoretic analysis has quantified a significant potential for the increase in capacity of wireless networks with the strategic addition of infrastructure nodes [114], [115]. Relay nodes have also been shown to reduce overall power consumption and increase the overall transmission capacity [116], [117]. Capacity can be improved because smaller radiated power also means smaller interference between or within cells. Using intelligent scheduling, it has been shown that relaying may increase data throughput in smaller networks by up to 60% [118]. The potential augmentation of capacity given by relay nodes enables the network to better support emerging services that demand higher bandwidth and require high quality of service (QoS), such as high quality multimedia services.

6.1.2 Relay Alternatives in Multi-hop Cellular Networks

Relaying through the use of ad hoc relay stations, fixed relay nodes or a combination of the two has been suggested by researchers. Network architectures such as those in [26-29] use mobile nodes or even automobiles [119] as routers, eliminating the need for additional costly infrastructure but adding a high level of complexity in the routing protocols and sophistication in the mobile nodes. Others, such as those in [30-35] use fixed relay nodes for more reliable performance at a slightly higher cost. Still others such as [29] and [34] are designed for use with both ad hoc and fixed nodes, incorporating the reliability of fixed relay systems and the flexibility of ad hoc relays. According to the 802.16 standard, three types of relay nodes are defined: nomadic, mobile and fixed [120]. Nomadic nodes are those that are used for temporary coverage for a particular event, an emergency situation or for disaster recovery. Mobile relays create a transient physical topology such as in a scenario where a cellular subscriber might route calls through other mobile subscribers. Mobile relay nodes cannot guarantee connectivity though - a mobile user cannot be connected through a mobile relay node if none is present within its range. Fixed relay nodes, which we deal with exclusively, offer predictable connectivity and may be mounted on towers, poles, buildings, lamp posts and the like. They are generally deployed and owned by the infrastructure provider to provide better quality (multi-hop) links between subscribers and a base station. It is assumed that relay nodes do not have a wired connection to the backhaul network.

6.2 Medium Access Control (MAC) in Multi-hop networks

Medium Access Control (MAC) techniques used to allocate scarce channel resources among competing users are vital to the efficiency of any wireless network. Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA), and Space Division Multiple Access (SDMA) are common methods used to allocate mutually exclusive channel resources to each connected user by a centralized control station, such as a mobile switching center in conventional cellular networks [121]. Such centralized approaches avoid contention among users, but they are inefficient for bursty traffic as a connected user may use its allocated resource for a small fraction of time. Furthermore, these circuit-based MAC solutions “hard-partition” the spectral resource to needy users such that each connected user has only a small share of the overall spectral resource and thus a low data throughput. Distributed scheduling techniques using random access MAC protocols, such as Slotted ALOHA [122] for cellular paging channel access and IEEE 802.11 Distributed Coordination Function (DCF) [123] for wireless LANs allow users to dynamically access a channel by a contention approach. Random wait times are introduced in the case of a busy channel and when there are collisions among stations on the shared medium. These distributed techniques are more efficient when dealing with bursty user traffic as they offer a user who successfully gains access to the shared channel the maximal spectral resource and thus high data throughput when the traffic volume is low. However, they suffer from significant throughput degradation due to heavy contention during periods of high traffic volume.

In many broadband access networks, such as cable broadband (under DOCSIS standards), and Fixed Wireless (under the IEEE802.16d standard) and WiMAX (under the IEEE802.16e standard), a reservation-based TDMA MAC protocol is used. Reservation-based TDMA is a compromise between the centrally controlled TDMA method and the decentralized random access MAC approach such as in ALOHA. In reservation-based TDMA, needy nodes make time slot reservations for their packet transmissions on a random access basis. The centralized controller has a MAC scheduler that processes successful reservations and assigns time slots in response to these reservations in subsequent TDM frames. Each TDM frame consists of a contention region (in contiguous time slots within the frame) in which needy nodes transmit their reservations on a contention basis and a payload region in which the scheduler does slot allocation in response to the successful reservations and in accordance with a desired scheduling algorithm. Collisions are thus confined to the contention region. Generally, the provision of the contention region is generous enough such that channel waste due to contention is quite limited. As a result, the throughput of reservation-based TDMA is much higher than that of slotted ALOHA and is more preferable than conventional TDMA when dealing with bursty data traffic which is common in internet access applications.

Although reservation-based TDMA yields a decent throughput in the aforementioned broadband access networks (DOCSIS, WiMAX, etc.), these networks are known to be of the “point-to-multi-point” (PMP) type. That is, all the user nodes (multi-points) share the broadband channel and they directly communicate to the centralized controller (single point) – the cable modem termination system or the WiMAX base station. When reservation TDMA is applied to a multi-hop environment, the scheduling task becomes much more complex. Specifically, within a single cell of a multi-hop cellular network, the controller will be the Base Station (BS) while the nodes that make reservations include Relay Stations (RS) and Mobile Stations (MS), which may also be referred to as subscriber stations (SS). These are defined in the IEEE 802.16j standard [124] which specifies mobile multi-hop option for WiMAX. In a multi-hop network, multiple access issues involve not only communications between MS and BS, but also communications between MS and RS and between RS and BS [125]. Therefore, in a centralized scheduling approach, the BS has to collect reservations from all the RSs and all the MSs and allocate time slots to all of these nodes in a way that maximizes the system throughput on one hand while maintaining fairness on the other. Certainly, as one of the proposed MAC solutions in IEEE802.16j, there is an alternative solution in which spectrum resources are partitioned between the two levels of communications: those between MSs and RSs and those between RSs and the BS. Such an approach will in effect break the MAC problem into the two levels such that an RS behaves like the BS from the MS's perspective and an RS behaves like an MS from the BS perspective. However the main drawback of this approach is the complexity and cost of the RSs. Therefore, our work focuses on the centralized scheduling approach where the BS controls all the uplink transmissions: those from the MS to the RS, those from the MS to the BS, and those from the RS to the BS. We capitalize on the fact that the BS has complete knowledge of the input traffic and output traffic for each RS at all times (because the BS controls their input and output traffic flows through time slot allocations and RS assignments to MSs) and accordingly design a MAC scheduler that constantly assigns time slots to each RS in such a manner that queues never develop in the relay stations and queueing delay at the RS is therefore negligible. We shall assume that the RSs are strategically deployed such that the channel quality between each RS and the BS is high enough to support the highest modulation and coding scheme available. The channel qualities between a given MS and an RS and between the MS and the BS are distance-dependent. A fixed path loss model is used in Section 6.3.5 to assess analytically the throughput advantage of the multi-hop system over the single-hop system. Finally, a distance-dependent random MCS model is used in Section 6.5.2 to assess the throughput advantage of the optimized multi-hop system over the single-hop system and the un-optimized multi-hop system by simulations.

We shall limit our study to a simple two-hop network where each MS is at most two hops away from the BS. Therefore upstream transmissions from a MS to the BS can use either a one-hop path directly to the BS, or a two-hop path through a selected RS. The decisions as to whether it should use the one-hop transmission or a two-hop transmission and in the latter case which RS it should use as relay agent is among the responsibilities of the MAC scheduler run by the BS. To enable the scheduler to carry out these responsibilities, the MSs constantly notify the BS of their channel qualities from the BS and from all the RSs from which they can receive control signals. The BS uses this information to construct and maintain the dynamic network topology which it uses to make scheduling decisions when the MSs reserve bandwidth through a contention process. Note that bandwidth requests may be sent to the BS by relay in the event that a MS does not have a direct path to the BS. In every scheduling cycle, the BS processes all the successful bandwidth reservations and broadcasts its scheduling decisions (including relay paths associated with each allocation) to the MSs.

6.3 Benefits of Multi-hop Relays

The main disadvantages of using relays are the redundant transmissions involved and potential queueing delays at the relay nodes. However, redundant transmissions may be more than offset by being able to do all these transmissions at significantly better channel qualities and thus at higher data rates. Thus relaying may potentially yield a net gain in throughput, provided that queueing delay does not pose a problem at the relay nodes. Specifically, relaying becomes more attractive when the data rates between a subscriber station and the relay node and then between the relay node and the base station are sufficiently higher than the data rate directly between the subscriber station and the base station and when the relay node does not add a significant queueing delay.

6.3.1 Access Delay

Besides transmission delay, which is related to the data rate of a link, queueing delay and propagation delay also contribute to the time it takes to deliver a packet from the MS to the BS. We shall use the term "access delay" to capture all the delay components along the multi-hop path: it is defined as the time since a packet is generated by a MS until it is successfully received by the BS. Clearly a large queue at the RS renders the multi-hop communication through the relay less attractive. However, as described earlier, our scheduling approach will contain queueing delays in relay nodes by always assuring that a packet heading for a relay station be promptly serviced.

As far as propagation delay is concerned, since wireless signals travel at 3×10^8 m/s, it is negligible in short range radio path compared to transmission delays.

6.3.2 Modulation and Coding Schemes (MCS)

In order to improve system capacity in mobile communication systems, the signal transmitted to and from mobiles can be dynamically adjusted according to the measured received channel quality. Adaptive modulation and coding allows the flexibility to dynamically adjust the modulation-coding scheme to the prevailing channel conditions for each mobile and hence maximize its throughput. Basically, mobiles having higher channel qualities are assigned higher modulation formats and/or higher channel code rates. In other words, a higher channel quality is associated with a higher level of modulation and coding scheme (MCS) which yields a higher data transmission rate. Generally, channel quality is represented by received signal-to-noise ratio (SNR) or signal-to-noise-and-interference ratio (SINR), depending on the medium access control protocol used. At a higher level, channel quality can be represented by the highest MCS level the channel can support, subject to a constraint on bit error rate (BER) or frame error rate (FER). Practically, in IEEE 802.16 channel quality is quantized according to a fixed set of MCS levels. The modulation formats used in these MCS levels include BPSK, QPSK, 16-QAM, and 64-QAM. The channel code rates used are ½, 2/3, and ¾. Table 6.1 displays the modulation format, the channel code rate and the required received SNR (or SINR) needed to support each MCS level and its resulting user throughput, according to the IEEE 802.16 PHY/MAC specifications [126,127,128].

Modulation	Code Rate	Received SNR (dB)	Throughput (Mbps)
BPSK	1/2	3	6.1
QPSK	1/2	6	12.19
QPSK	3/4	8.5	18.59
16QAM	1/2	11.5	24.69
16QAM	3/4	15	37.19
64QAM	2/3	19	49.68
64QAM	¾	21	55.78

Table 6.1: Modulation and Code rates for 802.16 and associated throughput.

6.3.3 Path Loss Exponent and MCS Levels

Signal to noise ratio generally declines according to the distance from the transmitter raised to a "path loss exponent" α, whose value is normally in the range of 2 to 5, where $\alpha = 2$ is for propagation in free space, and $\alpha = 5$ is for a highly lossy environment. Let r_α denote the cell radius which is the maximum distance from the base station such that a mobile can communicate with the base station using the lowest MCS level, BPSK with rate ½, when the path loss exponent is α. In the sequel, we shall call this MCS level "BPSK ½" for short.

Let

$\rho_n \equiv$ minimum SNR (or SINR) needed to support the nth MCS level

and

$d_n \equiv$ distance from the BS where the SNR (or SINR) equals ρ_n

Then for a fixed value of α, the following relationship applies:

$$\rho_i / \rho_j = (d_j^{\alpha})/(d_i^{\alpha}) \qquad (1)$$

Thus, using the cell radius r_α, with $\alpha = 2$, $\alpha = 3$, and $\alpha = 4$, respectively, the transmission radius for each MCS level can be determined according to (1) and is presented in Table 6.2. The proportion of the circular cell area corresponding to each MCS level based on these transmission radii for different path loss exponents is also calculated and shown in Table 6.3.

Modulation	**Code Rate**	**Radius α = 2**	**Radius α = 3**	**Radius α = 4**
BPSK	1/2	r_2	r_3	r_4
QPSK	1/2	$0.708r_2$	$0.794r_3$	$0.841r_4$
QPSK	3/4	$0.531r_2$	$0.656r_3$	$0.729r_4$
16QAM	1/2	$0.375r_2$	$0.521r_3$	$0.613r_4$
16QAM	3/4	$0.251r_2$	$0.398r_3$	$0.501r_4$
64QAM	2/3	$0.158r_2$	$0.293r_3$	$0.398r_4$
64QAM	3/4	$0.126r_2$	$0.251r_3$	$0.355r_4$

Table 6.2: Relative radii by MCS level and path loss exponent.

Modulation	**Code Rate**	**% of Area α = 2**	**% of Area α = 3**	**% of Area α = 4**
BPSK	½	49.88%	36.90%	29.21%
QPSK	½	21.93%	20.11%	17.71%
QPSK	¾	14.06%	15.86%	15.50%
16QAM	½	7.82%	11.27%	12.46%
16QAM	¾	3.80%	7.27%	9.27%
64QAM	2/3	0.93%	2.27%	3.26%
64QAM	¾	1.58%	6.31%	12.59%

Table 6.3: Proportion of area by MCS level and path loss exponent.

Figures 6.1a, 6.1b, and 6.1c show the division of a circular cell among the various coding and modulation schemes according to the values in Table 6.2. They show the cell radii corresponding to the selected break-point SNRs for each MSC level at α=2, α=3, and α=4. A higher path loss exponent significantly reduces the total coverage area of a BS. With r_2 being the maximum distance at which BPSK ½ can be used at α=2, the maximum radius to afford BPSK ½ at α=3, r_3, is approximately $r_2^{2/3}$. Likewise the maximum radius for BPSK at α=4 is approximately $\sqrt{r_2}$. Although the cell radii for higher path loss exponents can be maintained at r_2 by increasing transmitted power, we will not consider that because power consumption is as important as bandwidth conservation in a wireless communication system and a power constraint is always in place for such systems.

As can be seen from the concentric circles in the figures, the higher the path loss exponent, the larger the proportion of the cell area for the higher MCS levels and thus higher data rate. As an example in Table 6.2, the radii associated with 64QAM 2/3 ranges from 15.8% of the cell radius for α=2 to 39.8% of it at α=4. Moreover, the area associated with BPSK ½ decreases from nearly half of the cell area at α=2 to less than 30% of the cell area at α=4.

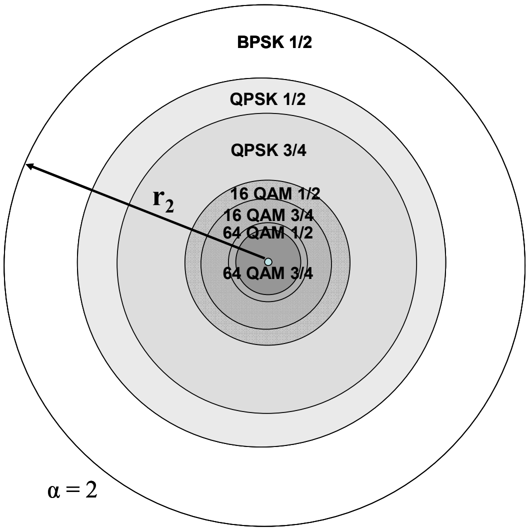


Figure 6.1a: With α=2, radii corresponding to 7 MCS Levels

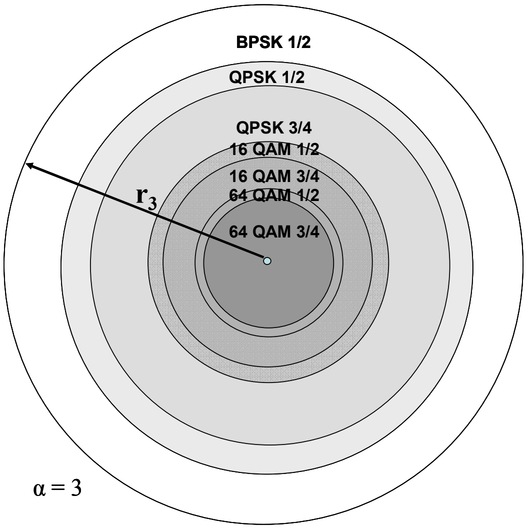


Figure 6.1b: Radii for α=3, where r_3 proportional to $r_2^{2/3}$.

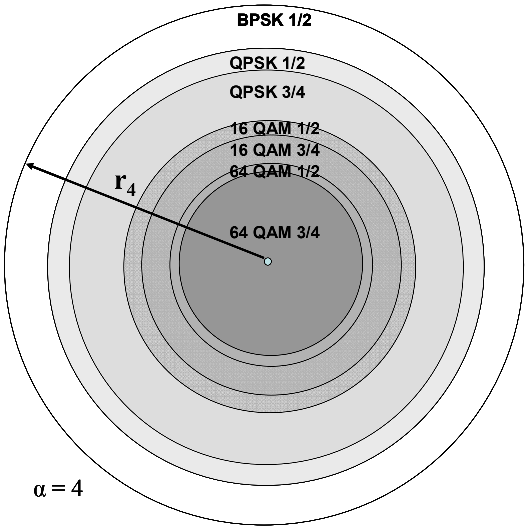


Figure 6.1c: Radii for α=4, where r$_4$ proportional to $\sqrt{r_2}$.

6.3.4 Average Cell Throughput

The average throughput of a cell, defined as the bit rate achievable by a user in the cell whose location is uniformly distributed within the cell, can be calculated for a multi-rate system using *n* MCS levels each with a corresponding bit rate R$_i$ and radius d$_i$ from the BS. As a result, the one hop cell throughput, *Th*, is derived as follows:

$$Th = \frac{\pi(d_1)^2}{\pi(d_n)^2} R_1 + \frac{\pi(d_2)^2 - \pi(d_1)^2}{\pi(d_n)^2} R_2 + ... + \frac{\pi(d_n)^2 - \pi(d_{n-1})^2}{\pi(d_n)^2} R_n$$

$$= \frac{(d_1)^2 R_1 + [(d_2)^2 - (d_1)^2] R_2 + ... + [(d_n)^2 - (d_{n-1})^2] R_n}{(d_n)^2} \qquad (2)$$

6.3.5 Throughput Improvement with Relays

Figure 6.2 shows the scenario where a MS may transmit directly to the BS in a single hop with a data rate R_{MB}. Alternatively the MS may transmit its data to the RS with a data rate R_{MR}; the RS then relays the data to the BS with a data rate R_{RB}. Therefore the MS has a single-hop throughput of

$$R_1 = R_{MB}$$

if it chooses to transmit directly to the BS. On the other hand, it has an effective throughput of

$$R_2 = 1/(1/R_{MR} + 1/R_{RB}) \qquad (3)$$

if it chooses the relay option instead. In the following we shall use $R_{MB(i)}$ to denote the data rate from a mobile to the base station using the ith MCS level and $R_{MR(i)}$ and $R_{RB(i)}$ the same for MS-to-RS and for RS-to-BS communications, respectively.

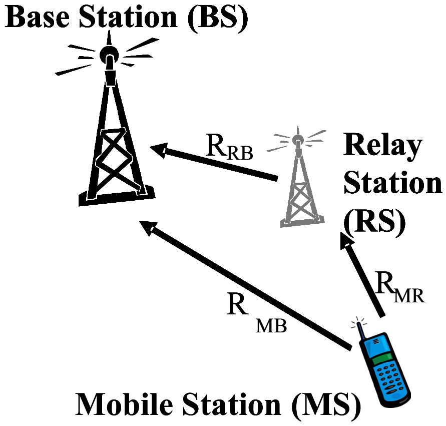


Figure 6.2: Data rates from MS to BS directly (R_{MB}) and through a RS ($1/(1/R_{MR} + 1/R_{RB})$).

In a multi-hop network, the addition of relay stations allows a MS to have multiple paths of transmitting data to the BS, including direct transmission to the BS or via one or more relay stations. System throughput is maximized if every MS is able to choose a path with the highest effective throughput. According to (3) the two-hop throughput would be $1/(1/R_{MR(i)} + 1/R_{RB(j)})$, if the BS is located in the i^{th} MCS level region from the RS and RS is located in the j^{th} MCS level region from the BS. Therefore if the k^{th} MS chooses its path to optimize its throughput, then its throughput is given by:

$$R_k = \max(R_{MB(i(k))}, \quad \frac{1}{(1/R_{MR(j(k))} + 1/R_{RB(t(k))})} \qquad (4)$$

where *i(k)* is the MCS levels for the MCS levels that the k^{th} MS can use when using the one-hop option and *j(k)*, and *t(k)* are the MCS level that the k^{th} MS and its relay node can use, respectively when it chooses the relay option. We can compute the average throughput of an MS whose location is uniformly distributed within the entire cell, $E[R_k]$ by using the distance based MCS statistics.

We expect that the addition of relay stations to increase the average cell throughput especially in environments with high path loss exponents. Since we can assume that relay stations are strategically placed such that they are LOS with the BS regardless of the path loss for MSs within the cell, the bit rate between the RS and BS will be high. Thus, the overall throughput for the two-hop case will most probably be dominated by the $R_{MR(i)}$ values. In particular, for MSs that are at a relatively large distance from the BS and are close to a relay node, we would expect significant throughput improvements.

In the following, the average user throughput within a cell using relays is analyzed by considering a base station surrounded by 6 relay nodes at a distance of 2r/3 from the BS and at 60 degree increments where r is the cell radius. It is assumed that relay stations are strategically positioned so that they have a line of sight channel (LOS), with path loss exponent α=2, to the base station. That is, we assume that the signal between the relay nodes and the BS propagates over an unobstructed path from the transmitter to receiver. In order to compare the performance with and without relay nodes, we consider 3 propagation scenarios, namely, α=2, α=3, and α=4, for the channel between SSs and the BS. In each scenario, we assume that the same path loss exponent applies throughout the entire circular cell, with the exception of the channel quality between the BS and RSs where we always assume that α=2. The coverage rings of each of the MCS levels within the cells for the three different path loss exponents are shown in Figures 6.3a-6.3c.

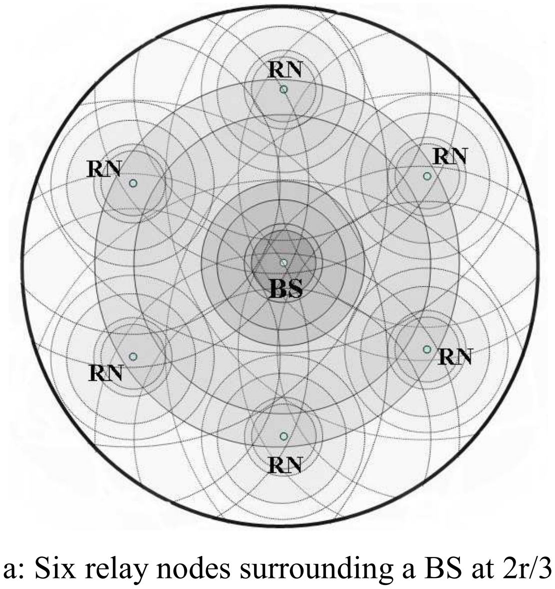


Figure 6.3a: Six relay nodes surrounding a BS at 2r/3 with α=2.

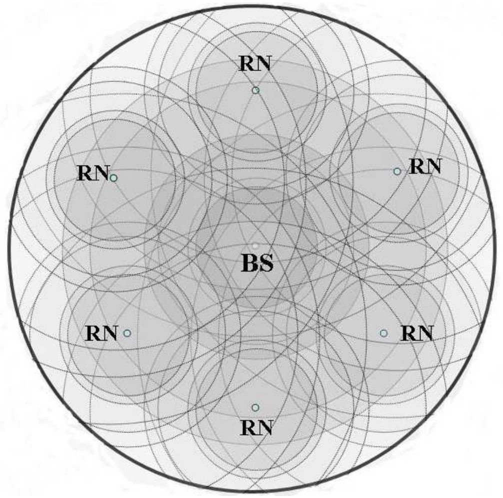


Figure 6.3b: Six relay nodes surrounding a BS at 2r/3 with α=3.

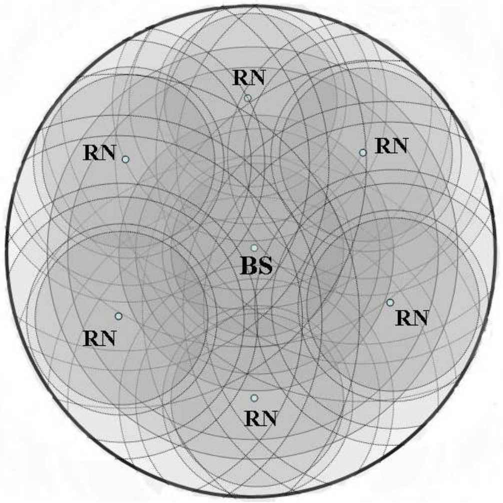


Figure 6.3c: Six relay nodes surrounding a BS at 2r/3 with α=4.

We assume that the cell radius associated with a given path exponent α is the maximum distance from the base station such that the lowest MCS level, BPSK ½ can be supported. Therefore the cell radius for α = 2 is much larger than that for α =3 and the cell radius for α = 3 is much larger than that for α =4. Within the cell radius corresponding to loss exponents α = 2, 3, and 4, respectively, the average throughput for a MS in the cell is calculated for both one-hop and two-hop communication options. Specifically, the value of E[R_k] is obtained by numerical computation given the areas corresponding to the MCS levels from each of the RSs and the BS. The same values were calculated using Shannon's AWGN channel capacity formula [129] which determines the maximum link throughput as a continuous function of SNR rather than using quantized MCS levels. These average throughput values were compared to E[$R_{MB(i)}$] to determine the average throughput advantages attained by using six relay nodes. The results are displayed in Table 6.4.

Path loss exponent	Throughput advantage with 7 MCS levels	Throughput advantage based on Shannon Capacity
α=2	18.1%	22.3%
α=3	58.4%	93.2%
α=4	47.4%	105.7%

Table 6.4: Throughput advantage for relay (with 6 relay nodes) compared to single-hop

Note that the increase in throughput with relays is partly due to the superior channel quality between the relay nodes and the base station. With $\alpha = 2$ within the cell, since the channel quality between the MS and the BS is also based on a path loss law of $\alpha = 2$, relay does not yield much throughput advantage because the channel quality improvement is not very large after offsetting the inefficiency of redundant communications. However, with $\alpha = 3$ or $\alpha = 4$, LOS transmissions ($\alpha = 2$) between the RSs and the BS yield much higher data rates than MS-to-BS links on one hand while shorter distance (on average) between the MS and the RS yields a larger data rate gain than the $\alpha = 2$ case on the other. Thus, with α=3 for both MS-to-BS and MS-to-RS links, and $\alpha = 2$ for the RS-to-BS link, the large improvement in throughput achieved by the two-hop case is attributed to the higher throughput between the RS and the BS and by the larger areas encompassed by the better MCS levels from the MS to the RS. Notice that the throughput advantage (compared to direct MS-to-BS communications) for α = 4 case is not as impressive as the α = 3 case for the seven-MCS level scenario. This is because the cell radius for α = 4 case is smaller than that for the α = 3 case and as a result, the proportion of the area within the cell where direct MS-to-BS communications have high MCS levels is larger for α = 4 case than for α = 3 case. In the mean time, the high data rate for RS-to-BS communications stays the same for the α = 4 case compared to the α = 3 case. When Shannon's AWGN channel capacity formula is used, instead of the seven discrete MCS levels, we maintain the assumption that the RS-to-BS communication is LOS: α = 2. The resulting throughput advantages for relay are also displayed in Table 6.4. As the Shannon capacity does not impose restrictions on MCS levels, cell throughput advantage is higher with α = 4 than with α = 3 due mainly to higher throughput between the RS and BS with a smaller cell radius and thus a shorter distance between the RSs and the BS.

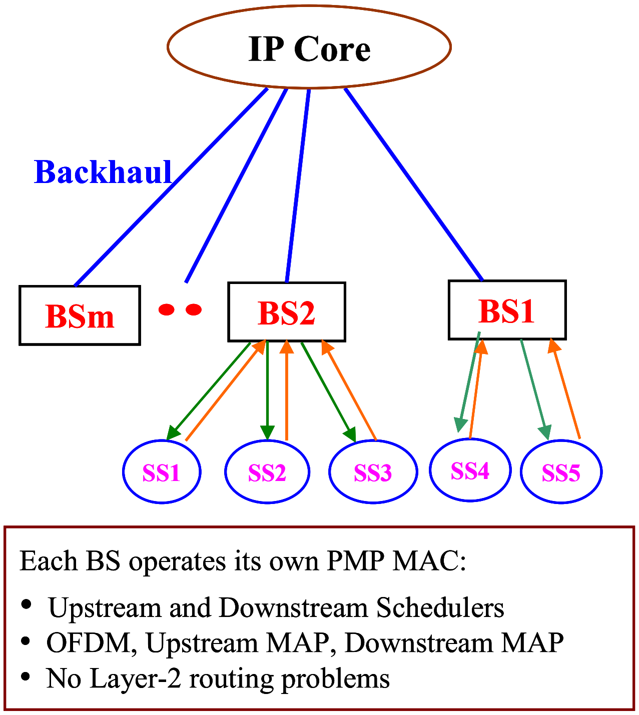
6.4 A Multi-hop Relay MAC Scheduling Problem

This section deals with uplink MAC scheduling in a limited two-hop wireless network. Specifically, we consider a centralized scheduling problem within a single cell where a base station is responsible for scheduling all the upstream transmissions from the subscriber stations. Each subscriber station can either transmit directly to the base station or uses one of the relay stations as a repeating node; in the latter case, two-hop communication is involved in delivering an uplink data packet from the subscriber station to the base station. It is assumed that all the nodes within the cell, including all the subscriber stations and all the relay stations, follow a common reservation TDMA MAC protocol and that there is no frequency reuse within the cell. Uplink traffic are generated randomly by the subscriber stations and have to be either transmitted directly to the BS or transmitted to a selected relay station first and then retransmitted to the base station by the relay station. We assume that a fixed proportion of time slots within each TDM frame (or uplink sub-frame if TDD is used) are allocated for signaling and control including bandwidth request transmissions and relays. The remaining time slots within a time frame will be referred to as data slots and these are the time slots for which the MAC scheduler is responsible for allocating to needy SSs and RSs. The subscriber stations utilize the signaling and control slots to make time slot reservations for their transmissions on a random access manner and we assume that the provisions for signaling and control is sufficient such that there is no stability problem caused by the random access based reservation process. Since the data slots are centrally allocated and since there is no frequency reuse, during any given time slot there will be at most one node transmitting.

The scheduling problem consists of two parts: deciding the upstream path for each needy SS and allocating the time slots within each TDMA frame, or upstream sub-frame if a TDD (time division duplex) MAC protocol is involved, to the needy SS/MSs and the RSs.

6.4.1 Relevance to IEEE802.16MMR

The IEEE802.16 protocol has been defined for fixed wireless SSs (802.16d) and has been extended to include mobile SSs (802.16e). More recently, the IEEE802.16j [130] initiative has been working to capitalize on the promising advantages of multi-hop wireless access in terms of coverage, capacity, and power. This entails creating innovative MAC-layer solutions for SSs and MSs to communicate with Base Stations (BSs) through the “routing fabric” made up of Relay Nodes (RSs). Figures 6.4 and 6.5 display the contrast between basic 802.16 (802.16 d, e) and 802.16-MMR. Basic 802.16 uses a point-to-multi-point (PMP) architecture whereby BSs are connected to the land-based core through a backhaul network that may be either wireless or land-based. This is illustrated in Figure 6.4 where one BS serves many Subscriber Stations (SS) directly just like a Cable Modem Termination System controlling internet access of many cable modems in a cable broadband network. Figure 6.5 illustrates the 802.16-MMR network that introduces RSs between the BSs and the MSs/SSs such that wireless broadband access uses multi-hop relay.



Each BS operates its own PMP MAC:

- Upstream and Downstream Schedulers
- OFDM, Upstream MAP, Downstream MAP
- No Layer-2 routing problems

Figure 6.4: WiMAX P2MP

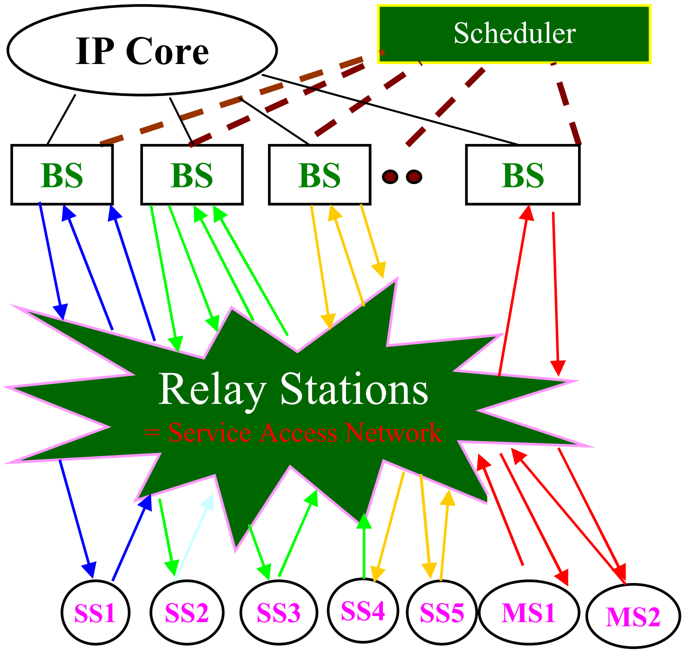


Figure 6.5: Multi-hop relay

The introduction of multi-hop relay to the IEEE802.16 access service network (ASN) entails an extension of the IEEE802.16 MAC protocol which was designed for PMP operation. To achieve the MMR design objective that an RS behaves as an MS from the perspective of a BS and as a BS from the perspective of an MS, unavoidably, the basic IEEE802.16 MAC framework must be retained at the BS and at the RS. However, many unique MAC issues arise with MMR; these include *spectrum allocation* between access links and relay links, upstream and downstream packet scheduling at an RS, spatial reuse between RSs and optimizing relay link spectral efficiency, multi-hop unsolicited grant service (UGS), and backward compatibility with 802.16 PMP MAC specifications.

Essentially the MMR network provides multi-hop links for the MS/SS to access the backhaul network and beyond. Therefore the time it takes a packet to traverse the MMR network is equivalent to access delay in a single-hop access channel and is a key performance indicator of the MMR network. As described in Section 6.3.1, *access delay* defined as the time it takes the MMR to deliver a packet from an MS/SS to a BS consists of *transmission* delays, *propagation* delays, *RS processing* times, and *RS queueing* delays. While minimizing transmission delays entails a routing path with maximal effective throughput, a routing path having the lowest *RS loads* is most likely to yield minimal queueing delay. Path throughput is determined by link qualities along the routing path and RS load is readily represented by the RS queue length. An optimum upstream routing path can be decided based on using MMR access delay as distance metric, which captures the two key attributes, *link quality* and *RS queue length*. The routing tree is established by having each participating RS determine and advertise its shortest distance to the BS. A needy MS/SS can then utilize among all the RSs within its range the best RS for relaying its messages based on the shortest distance information for each RS coupled with its specific *first-hop* channel quality information.

Basically, we consider a MMR-like multi-hop network with the following characteristics:

1. The uplink path from an SS/MS to the BS is either one-hop or two-hop. That is, every SS/MS must send its upstream packets either directly to the BS or to a selected relay node which then retransmits them to the BS.
2. The BS controls all the upstream data transmissions including selecting an upstream path for a needy SS/MS and allocating data slots to needy SS/MS and RSs.

3. The MAC protocol is reservation TDMA and the data slots are shared by all the active SS/MS nodes and all the RSs

Note that, since relay nodes do not generate traffic and since the BS dictates the input and output of each relay node, the BS has precise knowledge of the queue length of each RS. Furthermore, the overall traffic going from all the SS/MS to the RSs equal the overall traffic going from all the RSs to the BS. As such, we shall present a scheduling algorithm which strives to keep RS queues empty such that queueing delay at the RSs do not contribute significantly to the uplink access delay.

6.4.2 A Two-hop Wireless Network Scheduling Problem

This section lays out parameters in a centrally scheduled MAC transmission system and defines inter-relationships among them in the design of an efficient scheduling solution for subscriber stations and relay nodes. The scheme aims at maximizing throughput, minimizing average delay per packet, while treating subscriber stations fairly.

Figure 6.6 shows a base station with *k* relay stations and a subscriber station cloud made up of *N* possible users that may transmit either directly to the BS or via a RS. The system uses a MAC protocol in which the base station manages a constant number, *T*, time slots per frame and must allocate those slots among the SSs and RSs for their upstream transmissions. The *N* SSs each has their own channel quality to the BS and to each RS. These channel quality values can be tracked in a matrix, *S* as shown in Figure 6.7. The entries of the matrix are the number of slots per packet for the particular channel. Specifically, s_{ij} is the number of slots to deliver a packet from SS i to RS *j*, where $1 \le i \le N,$ and $1 \le j \le k$; and $s_{i,k+1}$ is the number of slots to deliver a packet from SS *i* directly to the BS. Clearly, a low number of slots per packet results from a higher channel quality and thus a high MCS level. A channel with no connectivity is given a value of ∞. For k relay nodes, the subscriber station channel quality matrix is a (k+1) x N matrix and indicates the number of slots required for each subscriber station to send one packet to each of the k relay nodes and the base station (which is the k+1 entry). The values for these entries range from 1 slot for excellent channel quality to infinity for a subscriber station with no connectivity. The SS channel quality matrix may change over time subject to the mobility of the subscribers and evolving channel conditions.

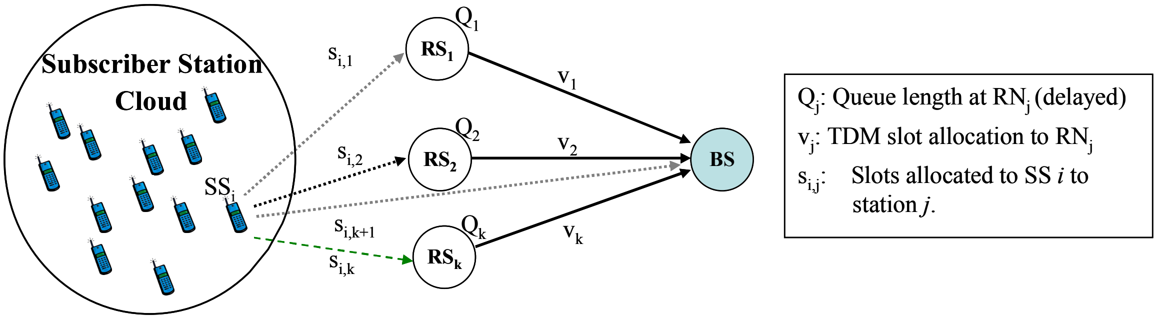

Subscriber Station Cloud
SS_i
s_{i,1}
s_{i,2}
s_{i,k+1}
s_{i,k}
RS_1
RS_2
RS_k
Q_1
Q_2
Q_k
v_1
v_2
v_k
BS
Q_j: Queue length at RN_j (delayed)
v_j: TDM slot allocation to RN_j
s_{i,j}: Slots allocated to SS i to station j.

Figure 6.6: Base station and relay stations with different channel qualities among them.

$$S = \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,k+1} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,k+1} \\ \cdots & \cdots & \cdots & \cdots \\ s_{N,1} & \cdots & \cdots & s_{N,k+1} \end{bmatrix}$$

Figure 6.7: Channel quality matrix: number of slots from SS *i* to destination *j*.

In addition to a subscriber station channel quality matrix, there is a “RS channel quality vector” **v**, which represents the channel quality between the relay nodes and the base station. Each entry v_j in this k × 1 vector contains for each of the relay nodes the number of slots that are required to send a packet from that relay node to the base station. The number of slots per packet can be calculated from the MCS level which depends on the quality of the channel in the same way as the subscriber station channel quality matrix. Since the RSs are stationary, unlike the mobile subscriber stations, their channel qualities will likely not change much over time and are thus assumed fixed in our study.

6.4.3 A Dynamic Markov Traffic Model

We construct a simple two-state Markov Chain to model the SS’s traffic state as shown in Figure 6.8. That the Markov Chain model is valid can be verified by the fact that the probabilistic behavior of a subscriber station in the next time increment depends only on the present state. According to this model, within a frame each subscriber station in state 0 has a probability of p that it will generate a new packet to send, thus moving it from state 0 to state 1. For a packet that is generated by a SS and therefore in state 1, there is a probability of q that it will be serviced by the system, thus bringing the SS back to state 0, where it may generate another packet. A SS may only generate a packet to send when it is in state 0. When a SS is in state 1; it remains in state 1 until the packet is transmitted. Packets in the system are serviced on a first-come, first-served basis.

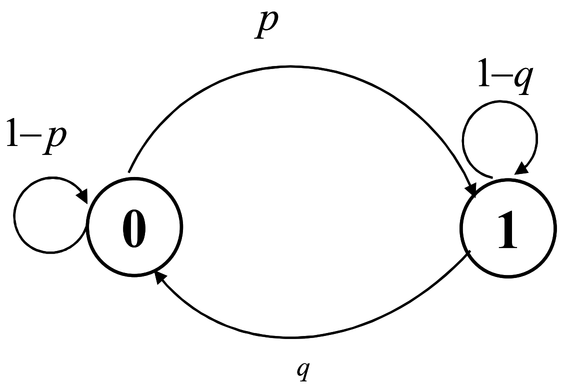


Figure 6.8: Two-state Markov chain model for a SS's traffic state

The Markov chain is thus represented by a transition matrix $P = \begin{bmatrix} 1-p & p \\ q & 1-q \end{bmatrix}$

Since it is a process described by a single, time dependent matrix, *P*, it is a time-homogeneous Markov chain that will reach equilibrium over time. These steady state probabilities are represented by the vector Π = [π_1, π_2] which satisfies the limiting theorem:

$$\Pi P = P$$

The vector *Π* gives the fraction of time that the SS is in states 0 and 1. Specifically, the solution of the above equation is given by :

$$\pi_0 = q/(p+q) \text{ and } \pi_1 = p/(p+q)$$

It must be noted that the calculation of the value of these probabilities is not trivial since the variables q and π_1 are mutually dependent. The probability of a packet being serviced, q, is also dependent on the capacity of the system and the number of slots used per constant sized packet. Variables that affect the value of q are discussed in the next section.

6.4.4 Transition Probability q

Clearly, the probability that a packet is serviced within a frame while the SS is in state 1, *q,* depends on the traffic load. We assume that the channel qualities of SSs are independent, and thus the number of slots each SS must use to send a packet are statistically independent. Let X denote the number of slots needed by a SS that is in state 1. This variable may either take the value of the number of slots per packet for direct transmission to the base station or the sum of the number of slots needed to transmit the packet from the SS to the RS and then from the RS to the BS. Let

$\mu_{SS} = E[X]$ and $\sigma^2_{ss} = VAR[X]$

be the mean and variance of X, which depend on packet size distribution and the channel quality statistics of the SSs. The average number of SSs that are in state 1 within a frame is given by

$m = N\pi_1$

Using these notations, the average and the variance of the number of slots needed by all the SSs within a frame can be expressed as $\mu = m\mu_{ss}$ and $\sigma^2 = m\sigma^2_{ss}$, respectively. Moreover, when the number of SSs is large, the probability distribution of the number of slots needed by all the SSs in state 1 within a frame approaches Gaussian with mean μ and variance σ^2. That is, the distribution of the number of total number of data slots per frame is approximately $N(m\mu_{ss}, m\sigma^2_{ss})$. A summary of the variables involved in the calculation of q is displayed in Table 6.5:

Variable	Description	Relation
N	Total number of SSs in system.	Independent variable
p	Probability of SS in state 0 generating a packet	Independent variable
T	Total number of available of data slots per frame.	Independent variable
μ_{SS}	Average channel quality of SS in slots per packet.	Independent variable
σ_{SS}	Standard deviation of SS slots per packet	Independent variable
π_0	Long term probability of SSs being in state 0.	q/(p + q)
π_1	Long term probability of SSs being in state 1.	p/(p + q)
m	Average number of SS in state 1	$N\pi_1$
q	Probability that a SS in state 0 gets its packet transmitted within a frame.	depends on *N*, *p*, μ_{ss}, σ^2_{ss}, and *T*

Table 6.5: Variables in MAC scheduling problem.

In each frame, the maximum number of slots that can be sent in the shared channel is *T*, which is assumed to be constant for every frame and is based on system bandwidth. When $X > T$, then *T* slots are serviced and $X - T$ are not. Therefore, for $X > T$, a fraction T / X of the SS traffic is serviced. The probability of exceeding the number of slots is related to the distribution of the aggregate within a frame demand whose probability distribution, as described above, is approximately $N(m\mu_{ss}\ ,m\sigma^2_{ss})$. Thus, the random variable $Z_x = (X - \mu)/\sigma$ is unit Gaussian and hence we can write

$$q = P(z \leq Z_T) + \sum_{X=T+1}^{\infty}[P(z \leq Z_X) - P(z \leq Z_{X-1})]\,(T/X) \qquad (5)$$

Unfortunately, we are unable to obtain a closed form solution for *q* because of a complicated recursive relationship in (5). It can be solved numerically, however, given values of the independent variables *N*, *p*, μ_{ss}, σ^2_{ss}, and *T*.

6.4.5 Scheduling Traffic Through a RS or the BS

Note that minimizing the value of μ_{SS} for each SS means maximizing bandwidth efficiency which will lead to maximal system throughput. If relaying can produce a lower average number of slots per packet per SS, including both hops of transmission, then it would be advantageous. Note that for any SS *I*, if there exists a relay station *j* such that $(s_{i,j} + v_j) < s_{i,k+1}$ (one-hop channel quality) then using two hops is advantageous, and will thus decrease the value of μ_{SS} compared to using the number of slots for direct communication to the BS. Moreover, when a SS is in proximity to more than one relay node, the best route is the one which minimizes the slot count per packet. In section 3, under the assumption that the path loss exponent is the same for all SS-to-RS communications, the best RS for an SS is always the closest one. Generally, however, the best relay node for a SS to use is not necessarily the relay node closest to the SS: it is the relay node for the subscriber station that has the best combination of channel qualities thus yielding the lowest average slot count for the SS.

Whether or not to use a relay node and, if one is used, which relay node is best for a SS, can be determined based on the channel quality matrix, *S*, and a “RS channel quality matrix”, V, determined from the RS channel quality vector **v** defined in Section 6.4.2. Specifically, the RS channel quality matrix V is shown in Figure 6.9 results simply from replicating the RS channel quality vector N-1 times and then adding a $(k + 1)^{st}$ all-zero row to account for direct transmission from the SS to BS for which there are no RS to BS slots required. The result is a $(k + 1)$ by *N* matrix.

$$V = \begin{bmatrix} v_1 & v_2 & \cdots & v_K & 0 \\ v_1 & v_2 & \cdots & v_K & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ v_1 & v_2 & \cdots & v_K & 0 \end{bmatrix}$$

Figure 6.9: Relay station channel quality matrix

Then the matrix $M = S + V$ will reveal the best relay strategy for SS i as follows: choose RS m(i) for SS_i if m(i) = arg min M_{ij}, $1 \le j \le k+1$. That is, $M_{im(i)}$ gives the minimum count of slots per SS to transmit a packet to the base station including both hops in case of a relay. Let **μ** = $[M_{i,m(i)}]$ be a vector that contains the optimum relaying strategies for all the SSs. Note that each number in the vector **μ**, namely $M_{i,m(i)}$ is random with mean μ_{ss} and variance σ^2_{ss} which have been defined in Table 6.5. Clearly if a relay strategy produces a smaller μ_{ss} and σ^2_{ss} compared to the others, it should lead to higher system throughput.

6.5 Simulation Study

6.5.1 Path Loss Model

The path loss model assumed in Section 6.3 does not consider the fact that various surrounding environmental obstacles and clutter may be distributed throughout a cell. While the model works well to model the average values of throughput in a cell, specific measures of path loss are random and often described in terms of a log-normal distribution about the mean distance-dependent value [10, 131]. The log-normal distribution models the shadowing effects that occur over many measurement locations that have the same separation from the receiver, but have different levels of clutter in the propagation path. In practice though, the terrain over which mobile communication takes place is often irregular, and there may be many obstacles such as trees and buildings. Thus, numerous propagation models have been developed to represent signal strength at particular receiving points in a service area. Some of the other well-known propagation model distributions are the Longley-Rice model [10, 132], Durkin's model [10, 133], Okumura Model [10, 134], Hata Model [10, 135], Walfisch-Bertoni Model [10, 136], and a Wideband PCS Microcell Model [10, 137]. Generally, path loss is random and related to the propagation distance from transmission. In our study, we use a high-level path loss model which uses a distance-dependent probability distribution of MSC level to represent the path loss. We then demonstrate the benefits of selecting the best relay node for multi-hop communication based on this high-level path loss model.

Consider the MCS Levels shown originally in Table 6.1 and displayed as eight levels in Table 6.6, where we label MCS levels from 1 to 8, from best to worst. The 8th MCS level is for the case where the SS does not have connectivity because its SNR is below the minimum required threshold for the 7th MCS level, BPSK with ½ channel code rate. The accompanying coverage rings are shown in Figure 6.10 and based on α=2 where the inner rings are likely to have higher MCS levels because of shorter distance to the receiver. Within each ring, we randomize the MCS level; that is the highest MCS level within a ring is achievable only if α=2 while three lower MCS levels are also equally probable. Specifically, we assume that within the K^{th} ring, the highest MCS level is K which has a probability of ¼, the next three MCS level, $(K + 1)^{th}$, $(K + 2)^{th}$, and $(K + 3)^{th}$, each has a probability of ¼ as well. Since there are only 8 MCS levels, the level above the 8th is considered as the 8th MCS level. The probability distribution of the MCS levels for each ring in Figure 6.10 is shown in Table 6.7.

MCS Level	Modulation	Code Rate	Received SNR (dB)	Throughput (Mbps)
1	64QAM	3/4	21	55.78
2	64QAM	2/3	19	49.68
3	16QAM	3/4	15	37.19
4	16QAM	1/2	11.5	24.69
5	QPSK	3/4	8.5	18.59
6	QPSK	1/2	6	12.19
7	BPSK	1/2	3	6.1
8	N/A	N/A	N/A	0

Table 6:6: Eight possible MCS Levels and their throughputs

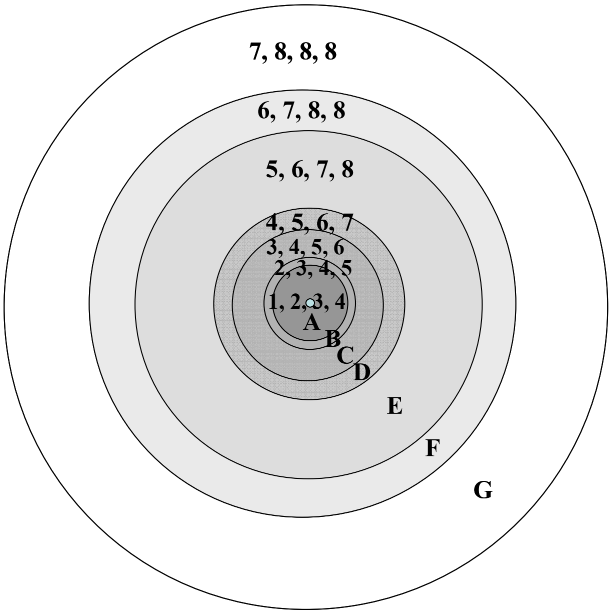


Figure 6.10: Seven rings of random MCS levels

	Ring A	Ring B	Ring C	Ring D	Ring E	Ring F	Ring G
P(MCS=1)	1/4	0	0	0	0	0	0
P(MCS=2)	1/4	1/4	0	0	0	0	0
P(MCS=3)	1/4	1/4	1/4	0	0	0	0
P(MCS=4)	1/4	1/4	1/4	1/4	0	0	0
P(MCS=5)	0	1/4	1/4	1/4	1/4	0	0
P(MCS=6)	0	0	1/4	1/4	1/4	1/4	0
P(MCS=7)	0	0	0	1/4	1/4	1/4	1/4
P(MCS=8)	0	0	0	0	1/4	1/2	3/4

Table 6.7: Probability Distribution Function of MCS level by rings around BS.

6.5.2 Simulations

We perform simulations to demonstrate the performance advantage of our proposed scheduling solution. We assume that the BS performs routing calculations that best optimize the path of packets from each of the SSs to the BS. The choice of path for the SS is not necessarily through the closest RS or BS because of random assignment of MCS level at any distance according to our model. Thus it must be decided whether the direct path to the BS or through a RS is the best path. If the best choice is not the direct path to the BS, then it must be determined which RS is the best. The routing choices may, depending on the scenario, have a significant impact on total throughput in the system.

As an example of the above case, we use a cloud of randomly positioned SSs within a cell with the random assignment of channel quality in terms of one of the eight MCS levels described above. Then based on the parameters described previously, we calculate the total throughput of the system with 3 scenarios:

1) The throughput of the SSs using their direct connectivity to the BS.

2) The throughput of the SSs using the relay station that is the closest by distance.
3) The throughput of the SSs using the relay station that gives them the lowest average total slot count to the BS, including both hops.

A series of simulations were run with 100 different sets of 100 uniformly distributed randomly placed SSs within a cell, whose radius is set to be the maximum distance from the base station such that the 7th MCS level can be supported when α=2. The 8 possible MCS levels were used as described in the previous section and shown in Figure 6.9 and Table 6.7. For each set of SSs, 10,000 different simulations were run for 15 simulated minutes: 100 simulations for each selected probability of generating a packet to transmit, where the selected probability *p* ranges from *p* = 0.01 to *p* = 0.25 in increments of 0.01, with all SSs having the same *p*. Six relay nodes were positioned at 60 degree increments from the BS and 2/3 of the radius away from the BS. The BS scheduled each transmission from a SS according to the scheme we described in the previous sections, determining the optimal path and then servicing the packet on a first-come, first-served basis.

6.5.3 Simulation Results

The results of the simulation are shown in Figure 6.11. They indicate, as expected, that although many SSs have to make two hops to the BS using the relay nodes, relays still yield higher throughput because the channel qualities to the relay nodes and then to the BS are far superior to a potentially lower-rate direct communication. In fact, the use of relay stations for these obstructed SSs improves the overall throughput of the system very significantly. If the closest RS is used, the improvement in throughput is 143% and when the best relay node is chosen the throughput is improved by 160%. The average throughput in each of the three cases reaches its upper limit as the traffic load increases to a certain point. Note that not only does the optimum relay approach have a higher maximum throughput but it can tolerate higher traffic load. This is further confirmed by the comparison based on average packet delay versus system throughput as presented below.

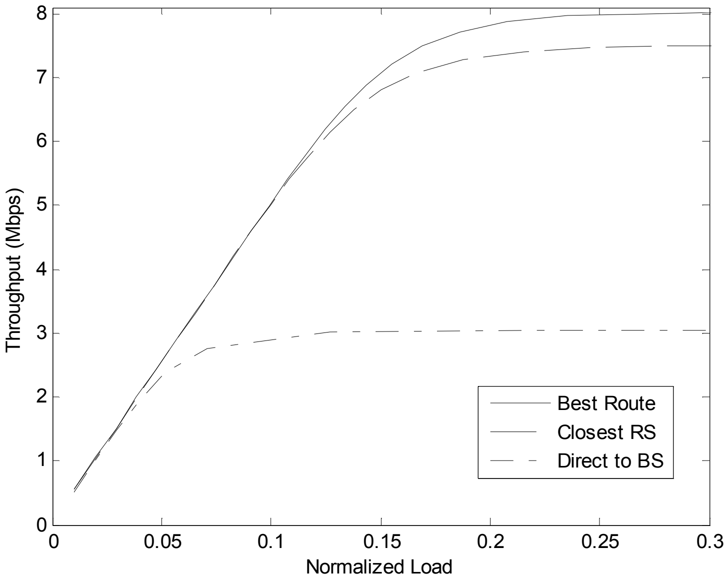


Figure 6.11: Average throughput for 3 scenarios- SS direct to BS, SS using the closest RN, and SS using best route to the BS.

Scenario	Max Throughput
One hop to BS	3.05 Mbps
Closest RS	7.49 Mbps
Best RS	8.02 Mbps

Table 6.8: Maximum average throughput by scenario

Figure 6.12 shows average packet delay versus system throughput for the three strategies under consideration. The packet delay here is the access delay defined earlier and consists of two delay components: waiting time at the SS transmit buffer, and the 2-hop transmission time, with propagation delay and relay node processing time neglected. It is observed that the performance of the optimum relay approach is drastically better than that of the single-hop approach and is clearly better than the un-optimized relay strategy.

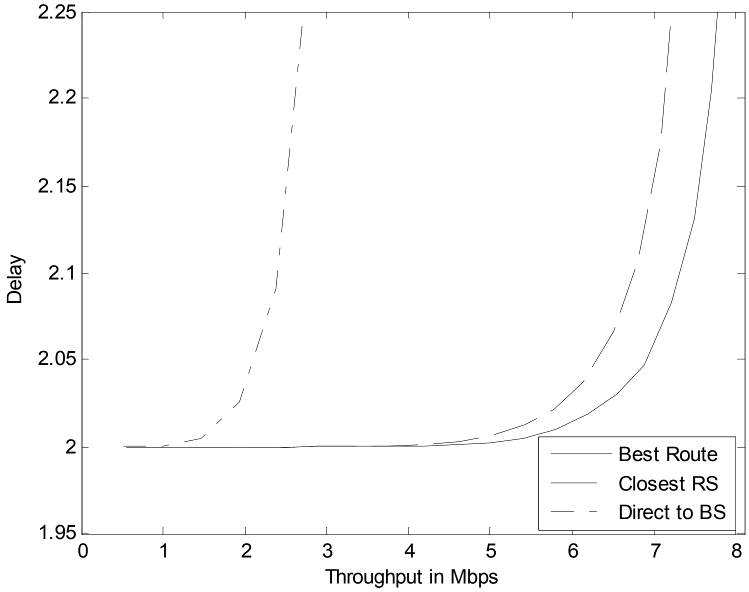


Figure 6.12: Throughput/ delay graph

6.6 Conclusion

This chapter has exploited the scenarios under which the use of relay stations improves throughput in a multi-hop cellular system. If the channel qualities using multiple-hop transmission are significantly improved over the conventional one-hop approach such that the gain in link throughput more than offsets the inherent redundant transmissions, the multiple-hop approach results in higher system throughput. We have shown that this improvement is more pronounced in harsher propagation environments with higher path loss exponents. Since relay stations can be strategically placed in LOS with the BS, they can provide significantly improved throughput for SSs especially in high path loss environments.

We have also developed a solution to a MAC scheduling problem, involving the scheduling of SS and RS transmissions by a cellular base station. Scheduling is performed in a centralized manner and involves assigning each SS an optimal route to the BS based on the one-hop and two-hop channel qualities. The BS manages the queues at the relay stations so that queueing delay at the relays are negligible, while it schedules upstream packet transmissions (to itself) from subscriber stations through an optimal relay station. Through simulation of our model, we have found and quantified that the intelligent scheduling and optimum relaying of packets from SSs improves system throughput dramatically compared to a single-hop system. We have also quantified the throughput improvement related to the use of the best choice among multiple relay nodes, which is the relay station that provides the highest throughput according to a randomized MSC level. That relays can substantially improve the throughput of multi-hop cellular network helps to validate the multi-hop cellular LSSON model that has been described in this Chapter 3.

Chapter 7

Conclusion

In this chapter, we summarize the contributions of this work and outline directions for future research.

7.1 Thesis Contributions

This work has taken on several important problems related to multi-hop wireless networks and has presented architectures, protocols and applications that successfully address those issues. In particular, we have designed low-complexity routing architecture and algorithms for a novel Large Scale Self-Organizing Network, we have designed a virtual circuit protocol for a multi-hop cellular network that exploited load-balancing among cells and analyzed its capacity advantages, we have tackled a medium access control problem in a simple multi-hop network and designed a scheduling solution that greatly improves the system throughput through optimum relay, and we have also presented an intriguing application of our architecture and protocols to support a real-time vehicle guidance in Intelligent Transportation Systems.

The issue of routing in multi-hop wireless networks is still relatively new and there are major challenges inherent in moving data from a source to a destination in a large wireless internetwork. One of those challenges is the complexity required to precisely route packets in a large scale network, which is on the order of O(n^2log$n$) for optimal approaches such as Dijkstra's algorithm. The use of hierarchical routing methods on the Internet backbone has successfully addressed this problem for wired internetworks, but such methods are not directly applicable to wireless networks since one cannot assume that links among nodes are fixed and stable. We have leveraged previous work on routing in wireless networks to develop a novel network architecture and associated protocols that are self-healing, self-organizing, and enable load balancing. Further, we present a routing methodology for a dense multi-hop cellular network that scales well to a very large number of nodes. As a result, complexity for large scale routing is reduced dramatically, enabling routing nodes in the network to be simple, inexpensive and easy to deploy. This is all accomplished through a proactive routing scheme that employs two-tiers: an upper tier for large scale cell-level paths and a lower tier for local tree-based routing. The upper-tier routing is accomplished by tracking a next hop cell for virtual routers that represent sectors emanating from a source root thereby forwarding the message to the parent node associated with the next cell's routing tree. Lower-tier routing involves simply routing to the parent node of the tree with respect to a destination base station. Once there, the message is broadcast to the destination user. Routing complexity is reduced to O(1) with a relatively small deviation from optimal hop counts. Simulations were performed on different sized networks ranging from 70,000 nodes to nearly 700,000 nodes to demonstrate successful performance of the large-scale routing approach.

With increasing use of real-time multimedia and voice applications on wireless networks there has been an ever-increasing need to ensure that these applications receive better bandwidth guarantees than the best-effort services offered in the past. Such assurances of quality of service have been especially elusive in multi-hop wireless networks which depend on every link in the path from source to destination to consistently provide the necessary resources for reliable communication. To deal with such a quality of service demand, we have developed a means to allocate network resources through the use of virtual circuits within the tree routing structure of our LSSON architecture. Algorithms and protocols have been designed to efficiently establish, track, and tear down virtual circuit connections in the network as necessary. Our solution seeks to load balance the loads among the cells thereby maximizing the virtual circuit capacity of the network, which is evaluated in terms of Erlang based connection blocking rate.

Medium access control (MAC) is a critically important issue in multi-hop cellular networks because it dictates how the wireless spectrum is shared among the base station, relay station, and the mobile stations. While contention-based MAC protocols such as 802.11 have matured and become widely used in wireless LAN environments, and centralized scheduling of fixed resources such as TDMA have been widely used in cellular networks, these MAC protocols are used almost exclusively for single hop scenarios. MAC protocols for multi-hop environments are still under development and subject to future standardization and thus provide a fertile ground for research. In our research, we have established the conditions necessary for improvements in user data throughput by the effective use of relay nodes in a multi-hop cellular network and have demonstrated that environments of high path loss are where relay stations can provide the most benefit to subscribers. We have also presented a MAC scheduling solution for a relay based multi-hop cellular network. In this centralized design, the base station chooses the best relay station for each subscriber to communicate through, it properly allocates the TDMA data slots that are shared by all the active nodes in the cell and effectively manages the queues in the relay nodes so as to render queueing delay in the relay nodes a negligible part of overall delay for users. Analysis, evaluation and extensive simulation of the system show significant performance benefits of our scheme.

Increased congestion on our nation's highways poses a significant opportunity cost for motorists and passengers because of the non-productive time spent waiting in traffic. Costly delays of shipped goods, wasted fuel and additional air pollution are further reasons why the United States Department of Transportation (DOT) continues to spend billions of dollars on projects to relieve traffic congestion on our highways. As one initiative to reduce congestion, the US DOT has been working on an Intelligent Transportation System (ITS) model which includes providing motorists with optimal real-time turn-by-turn directions based on system-wide traffic information to efficiently route traffic [88]. There are many technological hurdles which include efficiently collecting, processing and transmitting vehicle data and routing information. In our research, we have applied our network architecture and associated protocols to deliver real-time vehicle guidance. The wireless multi-hop network we have devised can be used as an effective means of gathering real-time vehicle data, communicating it to a processing center, and then transmitting the up-to-the-second vehicle guidance information to motorists. In our system architecture, communication functions and vehicle route processing are separated into distinct subsystems, providing ubiquitous communication through a roadside mesh-like sensor network and a design framework for low-complexity real-time vehicle routing table computation. An analysis of the bandwidth requirements associated with the proposed architecture and communication protocols demonstrates the feasibility of the automated real-time vehicle routing system. Through the use of roadside sensor locations as entries in vehicle routing tables, processing is minimized within the vehicle. Vehicle routing complexity is drastically reduced by using an on-demand routing strategy and a hierarchical vehicle routing approach which separates local routing from long-distance routing.

7.2 Directions for Future Research

Our work provides several potential springboards for future research. These include further development of the LSSON routing scheme with more complex network configurations, expanding on the vehicle routing subsystem we have proposed with new methods to route vehicle traffic, and an extension of our multi-hop MAC scheduling approach to accommodate downstream multi-hop communication to subscriber stations from a cellular base station. Additionally, a subscriber who is more than two hops from the cellular base station may be considered in the MAC scheduling model as well.

7.2.1 Further Developments of Large Scale Routing

Since we have demonstrated the feasibility of a large scale routing methodology that delivers near-optimal routing at a very low complexity, further delving into other aspects the model appears to be a promising avenue of research. As part of on-going development of LSSON, more advanced routing environments could be taken into account. Although we have explored different network sizes and topologies in our studies, our approaches are either geography based or hop-count based. It would be useful to develop low-complexity routing protocols based on a more general distance metric. The need of pursuing routing research based on a more advanced distance metric is also revealed in our work in the MAC scheduling of multi-hop transmissions. In particular, we have shown in Chapter 6 that the route with the lowest hop count to the base station is not always the best route. As such, it would be useful to adapt and test this type of network with metrics other than hop count, such as with a delay metric. Furthermore, running direct comparisons of our large scale routing protocol to other protocols such as AODV and DREAM might be useful in developing new hybrid protocols that deliver even better results.

7.2.2 Vehicle Routing Systems

The topic of vehicle routing is a large discipline of its own within transportation engineering. However, there is much to be learned in cross-departmental and inter-disciplinary study, and continued collaboration between algorithm experts in EECS-related disciplines and transportation experts is likely to yield interesting new results and may even be necessary in discovering more solutions to the insidious traffic congestion problems we all face. Other hierarchical routing solutions somewhat similar to ours have been proposed by transportation experts, but there are none that we are aware of that bring the complexity of vehicle routing down to O(1) as we have done with our packet routing solution. Vehicle routing is in many ways much more complex than packet routing since, regardless of guidance, motorists are human beings and ultimately exercise their own will when driving on the roadway. Even so, there is much to be gained in pursuit of these solutions. Moreover, constructive future research is also possible in the design of the Vehicle Routing Subsystem (VRS) we have proposed. The design of the VRS, as we detailed in the dissertation, left as an open issue whether the processing of vehicle routes is done centrally or in a distributed manner. With the increased capability of parallel processing techniques and the advancement of theory related to it, along with a better understanding of how to partition best route computations, useful algorithmic processing strategies can be developed to efficiently develop good routing heuristics.

7.2.3 Downstream Packet Scheduling

In our MAC scheduling solution we have only considered efficient upstream scheduling of possible multi-hop packets from subscribers whereas downstream transmissions are assumed to be one-hop, by high-power base stations. Consideration of the downstream multi-hop protocols is also likely to yield fruitful research results. At this time, our protocols assume a downstream broadcast protocol which works very well in a multi-hop cellular environment since it is legitimately assumed in a vast majority of cases that a cellular base station has sufficient transmission power for its signal to reach all nodes within its cell. However, there may be cases when the destination node is completely obstructed from the downstream broadcast. In this case, although the node may have multi-hop upstream connectivity to the base station, if there is no multi-hop downstream protocol, the node has no real connectivity to the base station. A multi-hop downstream protocol may include a source routing solution for which the route is learned from multi-hop upstream routing. MAC scheduling that involves downstream traffic will be a challenging and interesting area of research in multi-hop wireless networks using multiple routing trees.

7.2.4 Scheduling Distant Subscribers

Extending our MAC scheduling model to include subscribers with three hop or more connectivity to a cellular base station poses challenges due to bandwidth constraints and the complexities of scheduling transmissions from a combination of nodes one hop from the base station, two hops from the base station, three hops from the base station, and so on. The order with which these are scheduled in order to not starve needy nodes, fairly allocating bandwidth, and managing the queues in the nodes are among the important issues that need to be considered.

References

[1] M. Weiser, "The Computer for the Twenty-First Century," *Scientific American*, September, 1991.

[2] I. Chlamtac, M. Conti, J.J-N. Liu, "Mobile Ad Hoc Networking: Imperatives and Challenges," *Ad Hoc Networks*, vol 1, no. 1, July, 2003, pp 13-64.

[3] I Akyildiz, X. Wang, W. Wang, "A Survey on Wireless Mesh Networks," *IEEE Communications Magazine*, September, 2005.

[4] R. Bruno, M. Conti, "Mesh Networks: Commodity Multihop Ad Hoc Networks," *IEEE Communications Magazine*, March, 2005.

[5] Y.-D Lin, Y-C. Hsu, "Multihop Cellular : A New Architecture for Wireless Communications," *INFOCOM 2000*. IEEE, 2000.

[6] H. Wu, C. Qiao, S. De, O. Tongus, "Integrated Cellular and Ad Hoc Relaying Systems: iCAR," IEEE JSAC, vol 19, no. 10, Oct 2001, pp 2105-15.

[7] R. Pabst et al, "Relay-based deployment concepts for wireless and mobile broadband radio," *IEEE Communications Magazine*, vol 42, no. 9, pp 80-89, September, 2004.

[8] L. Hester, "A Self-Organizing Wireless Network Protocol," PhD Dissertation, Northwestern University, Department of Electrical and Computer Engineering, June 2001.

[9] H. Lee, "Protocols and Analysis of Self-Organizing Wireless Networks," PhD Dissertation, Northwestern University, Department of Electrical and Computer Engineering, June 2003.

[10] T.S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, 1996.

[11] IEEE Standard 802.16-2004 (Revision of IEEE Std 802.11-2001) "IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems," 2004.

[12] N. Esseling, H. S. Vandra, and B. Walke, "A Forwarding Concept for HiperLAN/2," Proc. Euro. Wireless 2000, Sept, 2000, Dresden, Germany, pp 13-18.

[13] www.meshnetworks.com which is re-directed to the Motorola.com site.

[14] MIT Roofnet: http://pdos.csail.mit.edu/roofnet/doku.php

[15] P. Gupta and P.R. Kumar, “The Capacity of Wireless Networks,” *IEEE Transactions on Information Theory*, vol 46, no 2, pp388-404 March 2000.

[16] U. Kozat, L. Tassiulas, “Throughput Capacity of Random Ad Hoc Networks with Infrastructure Support,” *MobiCom03*, 2003.

[17] A. Zemlianov, G. de Veciana, “Capacity of Ad Hoc Wireless Networks With Infrastructure Support,” *IEEE Journal on Selected Areas in Communications*, Vol 23, No 3, March, 2005, pp657-667.

[18] G.R. Grimmet, *Percolation*, Second Edition, Springer-Verlag: Berlon, 1999.

[19] E. N. Gilbert, “Random Plane Networks,” *Journal of SIAM 9*, 1961, pp 533-543.

[20] R. Meester, R. Roy, *Continuum Percolation*, Cambridge University Press, 1996.

[21] M. Franceschetti, L. Booth, M. Cook, R. Meester, J. Bruck, “Percolation in multi-hop wireless networks,” Submitted to IEEE Transactions on Information Theory, 2003.

[22] O Dousse, P Thiran, M Hasler, “Connectivity in ad-hoc and hybrid networks,” *Proc. Infocom 2002*.

[23] S. Shakkottai, R. Srikant, N. Shroff, “Unreliable Sensor Grids: Coverage, Connectivity, and Diameter,” *IEEE Infocom 2003*.

[24] H.-Y Hseih, R. Sivakumar, “Towards a Hybrid Network Model for Wireless Packet Data Networks,” *Proc of the Seventh International Symposium on Computers and Communications* (ISCC ’02), 2002.

[25] Y.-D. Lin and Y.-C Hsu, “Multihop Cellular: A New Architecture for Wireless Communications,” *Proc of IEEE INFOCOMM*, Tel-Aviv, Israel, March, 2000.

[26] H. Luo, R. Ramjee, P. Sinha, L. Li, S. Lu, “UCAN: A Unified Cellular and Ad-hoc Network Architecture,” *Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking* (Mobicom), September, 2003.

[27] J. Zhou, Y.R. Yang, “PARCelS: Pervasive Ad-hoc Relaying for Cellular Systems,” Med-Hoc-Net ’02, September, 2002.

[28] B. Bhargava, X. Wu, Y. Lu, W. Wang, “Integrating Heterogeneous Wireless Technologies: A Cellular Aided Mobile Ad hoc Network (CAMA), ACM Mobile Network and Applications, 2003.

[29] M. Lott, M. Weckerle, W. Zirwas, H. Li, E. Schultz, “Hierarchical Cellular Multihop Networks,” *EPMCC*, March, 2003.

[30] H. Wu, C. Qiao, S. De, O. Tongus, “Integrated Cellular and Ad Hoc Relaying Systems: iCAR,” IEEE JSAC, vol 19, no. 10, Oct 2001, pp 2105-15.

[31] C. Qiao, H. Wu, O. Tonguz, “Load Balancing Via Relay in Next Generation Wireless Systems,” *Proc of IEEE Conf Mobile Ad Hoc Networking Computing*, August, 2000, pp149-150.

[32] H. Vismanathan, S. Mukherjee, “Performance of Cellular Networks with Relays and Centralized Scheduling,” *Proc. IEEE VTC*, Orlando, FL, October, 2003.

[33] H. Wei, S. Ganguly, R. Izmailov, “Adhoc Relay Network Planning for Improving Cellular Data Coverage,” *The 15th IEEE International Symposium on Personal Indoor and Mobile Radio Communications* (PIRMC ’04), September, 2004.

[34] B. Walke, R. Pabst, and D. Schultz, “A Mobile Broadband System Based on Fixed Wireless Routers,” *Proc. Intl Conf. Commun. Tech*, 2003, Beijing, China, pp 1310-17.

[35] B. Walke, R. Pabst, and D. Schultz, et al, “Relay-Based Deployment Concepts for Wireless and Mobile Broadband Radio,” *IEEE Communications Magazine*, September, 2004.

[36] Y. Sun, E. Belding-Royer, X. Gao, J. Kempf, “Real-Time Traffic Support in Large Scale Mobile Ad Hoc Networks,” *Proc. of BroadWIM 2004*, San Jose, CA, October 2004.

[37] S. Wasserman and K. Faust, *Social Network Analysis*, Cambridge University Press, Cambridge, 1994.

[38] C. Koch and G. Laurent: Complexity and the nervous system. *Science* 284 (2 Apr 99): 96.

[39] Weng, G., Bhalla, U.S., and Iyengar, R. Complexity in Biological Signaling Systems. *Science* 284:92-96, 1999.

[40] A.-L. Barabasi, R. Albert, “Emergence of Scaling in Random Networks,” *Science* 286:509-512, October, 1999.

[41] S Dixit, E Yanmaz, K. Tonguz, “On the Design of Self-Organized Cellular Wireless Networks,” IEEE Communications Magazine, July 2005, pp86-93

[42] S. Dixit, E. Yanmaz, "Self-Organization of Relay-Based Next Generation Radio Access Networks (RANs)," *IEEE Intel Conf Pers Wireless Commun*, New Delhi, India, Jan 23-25, 2005.

[43] A. Itwata, C.C. Chiang, G. Pei, M. Gerla, and T.-W Chen, "Scalable Routing Strategies for Ad Hoc Wireless Networks," *IEEE JSAC*, vol 17, no 8, Aug 1999, pp 1369-79.

[44] G. Pei, M. Gerla, T.-W Chen, "Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks," *Proceedings of ICC 2000*, New Orleans, LA, June, 2000.

[45] C. Santivanez, R. Ramanathan, I. Stavrakakis, "Making Link State Routing Scale for Ad Hoc Networks," *Proc of 2001 ACM International Symposium on Mobile Ad Hoc Networking and Computing* (Mobihoc2001), Long Beach, CA, June 2001.

[46] P. Jacquet, P. Muhlethaler, A. Qayyum, A. Laouiti, L. Viennot and T. Clausen, "Optimized Link State Routing Protocol," draft-ietf-manet-olsr-05.txt, Internet Draft, IETF MANET Working Group, November, 2000.

[47] B. Bellur and R.G. Ogier, "A Reliable, Efficient Topology Broadcast Protocol for Dynamic Networks," *Proc of IEEE INFOCOM '99*, New York, March, 1999.

[48] R.G. Ogier, F.L. Templin, B. Bellur and M.G. Lewis, "Topology-Broadcast based on Reverse Path Forwarding (TBRPF)," dradt-ietf-manet-tbrpf-05.txt, Internet-Draft, Manet working group, March, 2002.

[49] D.B. Johnson, D.A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, edited by T. Imielinski and H. Korth, Chapter 5, Kluwer Publishing Company, 1996, pp153-181.

[50] C.E. Perkins and E.M. Royer, "Ad-hoc On-Demand Distance Vector Routing," *Proc of IEEE WMCSA '99*, New Orleans, LA, February, 1999, pp 90-100.

[51] T.-W. Chen, M. Gerla, "Global State Routing: A New Routing Scheme for Ad-hoc Wireless Networks," *IEEE ICC'98*, June 1998.

[52] C. Santivanez, R. Ramanathan, I. Stavrakakis, "Making Link-State Routing Scale for Ad Hoc Networks," *Proceedings of 2001 ACM Intl Symposium on Mobile Ad Hoc Networking and Computing*, Long Beach, CA, October, 2001.

[53] www.firetide.com

[54] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen. Scalable Routing Strategies for Ad-hoc Wireless Networks. *IEEE JSAC*, August 1999.

[55] The zone routing protocol (ZRP) for ad hoc networks (Internet-Draft),"
ZJ Haas, MR Pearlman - Mobile Ad-hoc Network (MANET) Working Group, IETF, Aug, 1998

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

[57] H. Frey, "Scalable Geographic Routing Algorithms for Wireless Ad Hoc Networks," *IEEE Network Magazine*, January 2004, pp 18-22.

[58] B. Karp, B. H.T. Kung,, "Greedy Perimeter Stateless Routing for Wireless Networks", ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000), pp. 243-254.

[59] S. Basagni, I. Chlamtac, V.R. Syrotiuk and B.A. Woodward, "A Distance Routing Effect Algorithm for Mobility (DREAM)," *ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom98)*, 1998, pp 76-84.

[60] Y.-B. Ko, N.H. Vaidya, "Locaiton-Aided Routing (LAR) in Mobile Ad Hoc Networks," *ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom98)*, 1998, pp 66-75.

[61] H. Li, M. Lott, W. Zirwas, E. Schulz, "Multihop communication in future mobile networks," *Proceedings of PIMRC*, September, 2002.

[62] R. Ananthapadmanabha, B.S. Manoj, C. Siva Ram Murthy, "Multi-hop Cellular Networks: The Architecture and Routing Protocols," *Proc. PIMRC*, 2001.

[63] H. Lee, C.-C. Lee, "An Integrated Multi-hop Cellular Data Network," *Proc VTC '03*, Orlando, FL, October, 2003.

[64] H. Lee, C.-C. Lee, "Layer Scheduling in a Multiple-Layer Self-Organizing Wireless Network," *Proc. of Globecom '02*, Taipei, Taiwan, November, 2002.

[65] L. Hester, "A Self-Organizing Wireless Network Protocol," PhD Dissertation, Program of Electrical & Computer Engineering, Northwestern University, June, 2001.

[66] T. Lin, S. F. Midkiff, J. S. Park, J.S, “A framework for wireless ad hoc routing protocols,” *Wireless Communications and Networking*, 2003, March 2003 Volume: 2, pp 1162- 1167 vol.2.

[67] M. Castro, P. Druschel, Y. C. Hu, A. Rowstron, Topology-Aware Routing in “Structured Peer-to-Peer Overlay Networks,” *Future Directions in Distributed Computing: Research and Position Papers*, Volume 2584/2003, Springer Berlin / Heidelberg, 2003. pp 103-107. And references therein.

[68] D. Subramanian, P. Druschel, J. Chen, “Ants and Reinforcement Learning: A case study in routing dynamic data networks,” *Proceedings of IJCAI-97*, 1997.

[69] D. Braginsky, D. Estrin, “Rumor Routing for Sensor Networks,” *International Conference on Distributed Computing Systems*, 2002. See references therein for other sensornet algorithms mentioned.

[70] B. Bhargava, X. Wu, Y. Lu, W. Wang, “Integrating Heterogeneous Wireless Technologies: A Cellular Aided Mobile Ad hoc Network (CAMA), *ACM Mobile Network and Applications*, 2003. References therein.

[71] B. Walke, R. Pabst, and D. Schultz, “A Mobile Broadband System Based on Fixed Wireless Routers,” *Proc. Intl Conf. Commun. Tech*, 2003, Beijing, China, pp 1310-17.

[72] S Dixit, E Yanmaz, K. Tonguz, “On the Design of Self-Organized Cellular Wireless Networks,” *IEEE Communications Magazine*, July 2005, pp86-93.

[73] S. Hanan, *The Design and Analysis of Spatial Data Structures*, Addison Wesley, 1990.

[74] J.L. Bentley, “Multidimensional Binary Search Trees Used for Associative Searching,” *Communication of the ACM*, 18(9), September 1975.

[75] D. Moore, “The Cost of Balancing Gneralized Quadtrees,” SMA '95: *Proceedings of the Third Symposium on Solid Modeling and Applications*, February, 1995.

[76] M.S. Daskin, *Network and Discrete Location: Models, Algorithms, and Applications*, Wiley, 1995.

[77] S.L. Hakimi, “Optimum Locations of Switching Centers and the Absolute Centers and Median of a Graph,” Operations Research, 12:450-459, 1964.

[78] T. Cormen, C. Leiserson, R. Rivest, C. Stein, *Introduction to Algorithms*, MIT Press, 2001.

[79] http://www.apple.com/iphone/

[80] http://www.pvnetsolutions.com/

[81] G.R. Ash, J.S. Chen, A.E. Frey, B.D. Huang, C.K. Lee and G.L. McDonald, "Real-Time Network Routing in the AT&T Network - Improved Service Quality at Lower Cost," *Proceeding of the Thirteenth International Teletraffic Congress*, Copenhagen, Denmark, June 1991.

[82] D. Mitra and J.B. Seery, "Comparative Evaluations of Randomized and Dynamic Routing Strategies for Circuit-Switched Networks," *IEEE Transactions on Communications, Vol. 39*, No. 1, January 1991.

[83] C. Qiao, H. Wu, O. Tonguz, "Load Balancing via Relay in Next Generation Wireless Systems," *Proceedings of IEEE Mobile Ad Hoc Networking & Computing*, 2000.

[84] Sriram, Reddy, Manoj, Murthy, "On the End-to-end Call Acceptance and the Possibility of Deterministic QoS Guarantees in the Ad hoc Wireless Networks," *International Symposium on Mobile Ad Hoc Networking & Computing*, 2005.

[85] C. Qiao, H. Wu, "iCAR: an Integrated Cellular and Ad-hoc Relay System," IEEE International Conference on Computer Communication and Network, pp. 154--161, 2000.

[86] Yilin Zhao, *Vehicle and Navigation Systems*, Boston, Artech House, Inc. 1997.

[87] Wunderlich, Kaufman, Smith, "Improved Link Travel Time Prediction for Decentralized Route Guidance Architectures", *IEEE Transactions on ITS*, March 2000.

[88] U.S. Department of Transportation, *Theory of Operations, National Intelligent Transportation Systems Architecture*, Washington, DC, 1998.

[89] See http://www.gcmtravel.com/gcm/home.jsp

[90] Shawn M. Turner, William L. Eisele, Robert J. Benz, and Douglas J. Holdener, "Travel time collection handbook," Tech Rep. FHWA-PL-98-035, Federal Highway Administration, Washington, DC, March 1998.

[91] Ramachandran, R. Sun, C., Arr, G. Ritchie, S.G. Intelligent Multi-detector fusion for Vehicle Re-identification. *82nd Annual Meeting of the Transportation Research Board*, 2003.

[92] B. Shrick and M. Riezenman, "Wireless broadband in a box," *IEEE Spectrum Magazine*, pp 38-43, June, 2002.

[93] D. Beyer, "Fundamental characteristics and benefits of wireless routing ("mesh") networks," in *Proc. Of the International Technical Symposium of the Wireless Communications Association*, (San Jose, CA), January, 2002.

[94] R. Bruno, M. Conti, E. Gregori, "Mesh Networks: Commodity Multihop Ad Hoc Networks," *IEEE Communications Magazine*, March, 2005, pp 123-131.

[95] Y. Huang, N. Jing, and E. Rudensteiner, "Hierarchical Path Views: A Model Based on Fragmentation and Transportation Road Types," *Proc. 3rd ACM Workshop Geographic Information Systems*, pp 93-100, 1995.

[96] N. Jing, Y.-W. Huang, and E.A. Rudensteiner, "Hierarchical Optimization of Optimal Path Finding for Transportation Applications," *Proc. Fifth Intl Conf. Information and Knowledge Management* (CIKM '96), pp 261-268, 1996.

[97] D.L. Gerlough, M. Huber, "Traffic Flow Theory: A Monograph TRB Special Report 165" Transportation Research Board, Washington, DC, 1975.

[98] *Highway Capacity Manual 2000*; 4th Edition, Transportation Research Board, National Research Council, Washington, DC, 2000.

[99] Roess, Prassas, McShane, *Traffic Engineering*; 3rd Edition, Pearson Prentice Hall, Upper Saddle River, NJ, 2004.

[100] Y. Huang, N. Jing, and E. Rudensteiner, "Hierarchical Path Views: A Model Based on Fragmentation and Transportation Road Types," *Proc. Third ACM Workshop Geographic Information Systems*, pp 93-100, 1995.

[101] K. Ishikawa, M. Ogawa, S. Azume, and T. Ito, "Map Navigation Software of the Electro Multivision of the '91 Toyota Soarer," *Proc. Intl Conf. Vehicle Navigation and Information Systems* (VNIS IVHS), pp 463-473, 1991.

[102] B. Liu, S. Choo, S. Lok, S. Leong, S. Lee, F. Poon, and H. Tan, "Integrating Case-Based Reasoning, Knowledge-Based Approach for Dijkstra Algorithm for Route Finding," *Proc. 10th Conf. Artificial Intelligence for Applications* (CAIA '94), pp 149-155, 1994.

[103] J. Shapiro, J. Waxman, and D. Nir, "Level Graphs and Approximate Shortest Path Algorithms," *Networks*, vol 22, pp 691-717, 1992.

[104] N. Jing, Y.-W. Huang, and E.A. Rudensteiner, “Hierarchical Optimization of Optimal Path Finding for Transportation Applications,” *Proc. Fifth Intl Conf. Information and Knowledge Management* (CIKM ’96), pp 261-268, 1996.

[105] R. Goldman, N. Shivakumar, S. Venkatasubramanian, and H. Garcia-Molina, “Proximity Search in Databases,” *Proc. 24th VLDB Conf.*, pp 26-37, 1998.

[106] S. Jung, S. Pramanik, “An Efficient Path Computation Model for Hierarchically Structured Topographical Road Maps,” *IEEE Transactions on Knowledge and Data Engineering,* Vol 14, No 5, Sep/Octr 2002.

[107] V. Taylor, B. Holmer, E. Schwabe, M. Hribar, “Balancing Load versus Decreasing Communication: Exploring the Tradeoffs,” *Proc. of the 29th Annual Intl Conf on System Sciences*, pp 585-593, 1996.

[108] L. Kucera, “Expected Complexity of Graph Partitioning Problems,” *Discrete Applied Mathematics*, Vol. 57, Issue 2-3, February 1995.

[109] G. Karypis, E. Han, V. Kumar, “A Hierarchical Clustering Algorithm using Dynamic Modeling,” IEEE Computer, 32(8), pp 68-75. 1999. See downloadable software at www-users.cs.umn.edu/~karypis/metis/.

[110] J.L. Bentley, “Multidimensional Binary Search Trees Used for Associative Searching,” *Communications of the ACM*, 18(9), September 1975.

[111] L. Kleinrock and K. Stevens. "Fisheye: A Lenslike Computer Display Transformation" Technical report, UCLA, Computer Science Department, 1971.

[112] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen, "Scalable Routing Strategies for Ad Hoc Wireless Networks" *IEEE Journal on Selected Areas in Communications, Special Issue on Ad-Hoc Networks*, Aug. 1999, pp.1369-79

[113] M. Gasper, M. Vertterli, “On the Capacity of Wireless Networks: The Relay Case,” IEEE, 2002.

[114] U. Kozat, L. Tassiulas, “Throughput Capacity of Random Ad Hoc Networks with Infrastructure Support,” *MobiCom03*, 2003.

[115] A. Zemlianov, G. de Veciana, “Capacity of Ad Hoc Wireless Networks With Infrastructure Support,” *IEEE Journal on Selected Areas in Communications*, Vol 23, No 3, March, 2005, pp657-667.

[116] L. Andrew, A. Kusuma, “Minimum Power Routing for Multihop Cellular Networks,” *Proc of IEEE Global Telecommunications Conference GLOBCOM’02*, Taipei, ROC, Nov, 2002, vol1. pp37-41.

[117] H. Karl, S. Mengesha, “Relay Routing and Scheduling for Capacity Improvement in Cellular WLANs,” *Proc. Modeling and Optimization in Mobile Ad Hoc and Wireless Networks* (WiOpt ’03), Sophia-Antipolis, France, March, 2003.

[118] H. Karl, S. Mengesha, “Analyzing Capacity Improvements in Wireless Networks by Relaying,” *In Proc. IEEE Intl Conf. Wireless LANs and Home Networks*, pp 339-348, Singapore, December, 2001.

[119] R. Morris, J. Jannotti, F. Kaashoek, J. Li, D. Decouto, “CarNet: A Scalable Ad Hos Wireless Network System,” *Proceedings of the 9th ACM SIGOPS Europeon Workshop*, September, 2000.

[120] P. Kyasanur and N.H. Vaidya, “Routing and Interface Assignment in Multi- Channel Multi-Interface Wireless Networks,” *Proceedings of IEEE WCNC 2005*, March, 2005.

[121] A. Tanenbaum, *Computer Networks,* Prentice Hall, 2003.

[122] N. Abramson, “The ALOHA System- Another Alternative for Computer Communications,” Proc. AFIPS Conf, Fall Joint Computer Conference, vol 37, 1970, pp. 281-285.

[123] IEEE Computer Soceity LAN MAN Standards Committee, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” IEEE Standard 802.11-1999. New York, NY: IEEE 1999.

[124] “Definition of terminology used in Mobile Multihop Relay”, IEEE 802.16j MMR Contribution, May, 2006.

[125] P Mach, R. Bestak, “Performance of IEEE 802.16 with Relay Stations,” *IEEE 6th Conference on Telecommunications*, May, 2007.

[126] IEEE 802.16tg-YY/n: “Channel Models for Fixed Wireless Applications”, V. Erceg, et al, 2001-07-16, IEEE 802.16 Broadband Wireless Access Working Group http://ieee802.org/16

[127] IEEE 802.16-2004, IEEE Standard for Local and Metropolitan Area Networks – Part 16: Air Interface for Fixed Broadband Wireless Access Systems, October 2004.

[128] IEEE 802.16e-2005 and IEEE 802.16-2004/Cor1-2005, IEEE Standard for Local and Metropolitan Networks – Part 16: Air Interface for Fixed Broadband Wireless Access Systems, Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1, February, 2006.

[129] S. Haykin, *Digital Communications*, New York: Wiley, 1988.

[130] P Mach, R. Bestak, "Performance of IEEE 802.16 with Relay Stations," *IEEE 6th Conference on Telecommunications*, May, 2007.

[131] D.C. Cox, R. Murray, A. Norris, "Measurements of 800 MHz Radio Transmission into Buildings with Metalic Walls," Bell Systems Technical Journal, Vol 62, No. 9, pp 764, 1987.

[132] A.G. Longley, P. L. Rice, "Prediction of Tropospheric Radio Transmission Loss Over Irregular Terrain; A Computer Method," ESSA Technical Report, ERL 79-ITS 67, 1968.

[133] R. Edwards, J. Durkin, "Computer Prediction of Service Area for VHF Mobile Radio Networks," Proceedings of IEE, Vol 116, No, 9, pp. 1493-1500, 1969.

[134] T. Okumura, E. Ohmori, K. Fukuda, "Field Strength and Its Variability in VHFand UHF Land Mobile Service," *Review Electrical Communication Laboratory*, Vol 16, no. 9-10, pp 825-873, September-October 1968.

[135] M. Hata, "Empirical Formula for Propagation Loss in Land Mobile Radio Services," IEEE Transactions on Vehicular Technology, Vol VT-29, No. 3, pp 317-325, August, 1980.

[136] J. Walfisch, H. Bertoni, "A Theoretical Model of UHF Propagation in Urban Environments," *IEEE Transactions on Antennas and Propagation*, Vol AP-36, pp. 1788-1796, October, 1988.

[137] M. Feuerstein, K. Blackard, T. Rappaport, S. Seidel, H. Xia, "Path Loss, Delay Spread, and Outage Models as Functions of Antenna Height for Microcellular System Design," IEEE Transactions on Vehicular Technology, Vol 43, No 3, pp 487-498, August, 1994.