**FACULTÉ DES ÉTUDES SUPÉRIEURES
ET POSTOCTORALES**

uOttawa
L'Université canadienne
Canada's university

**FACULTY OF GRADUATE AND
POSDOCTORAL STUDIES**

Bosko Vukojevic
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Master of Computer Science
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Depth First Search and Position Based Routing in Ad Hoc and Sensor Wireless Networks
TITRE DE LA THÈSE / TITLE OF THESIS

Ivan Stojmenovic
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Ramiro Liscano

Nicola Santoro

Gary W. Slater
LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES /
DEAN OF THE FACULTY OF GRADUATE AND POSTDOCORAL STUDIES

Depth First Search and Position Based Routing in Ad Hoc and Sensor Wireless Networks

by

Bosko Vukojevic

Thesis submitted to the
School of Information Technology and Engineering
in partial fulfillment of
the requirements for the degree of

Master of Computer Science

Under the auspices of the
Ottawa-Carleton Institute for Computer Science

University of Ottawa
Ottawa, Ontario, Canada

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-494-14964-7
Our file Notre référence
ISBN: 0-494-14964-7

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

I hereby declare that I am the sole author of this thesis.

I authorize the University of Ottawa to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Signature

I further authorize the University of Ottawa to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature

Acknowledgements

I would like to express my sincere gratitude to my research supervisor Professor, Dr. Ivan Stojmenovic, for his continuing patronage. His guidance and our numerous technical discussions led to the success of my research and learning experience.

Quite often Dr. Stojmenovic helped me to focus on the larger problem and work on finding alternate solutions my topic of study. His hard work and continuing dedication to research goals never fail to impress and inspire me. For all of his support during these years, I sincerely thank him.

I would like to express gratitude to my parents, my eternal inspiration and love. I thank them for all their support, for their faith in me and teaching that I should never give up.

Vesna, my wife, has been with me throughout this endeavor. She encouraged me whenever I stumbled and helped me through countless hours of work and study. Without her love and help this project would not have been possible. To her and to our daughters, Teodora and Marija, for all the support, understanding and patience, I am endlessly grateful.

Abstract

Finding a solution to the routing problem in wireless ad hoc and sensor networks has proven to be a quite difficult. A number of position-based localized algorithms have demonstrated important features like guaranteed delivery, scalability, robustness, and energy conservation, just to name a few.

This thesis proposes DFS, Depth First Search, routing algorithm. It is the first position-based localized algorithm that guarantees the delivery for (connected) ad hoc wireless networks modeled by arbitrary graphs, including inaccurate location information for a destination node. The DFS routing scheme is extended in order to provide QoS (quality of service) routing for wireless ad hoc networks. The goal of the DFS QoS is to minimize hop count, which resemble propagation delay requirement.

Another contribution of the thesis is the introduction of a new set of localized routing algorithms. The best DFS routing candidate is integrated with recently proposed modification of power and cost-aware solutions, so called progress-based power algorithms. Localized power progress-based routing schemes either minimize total power for routing of a message or maximize the total number of routing tasks that a network can perform before the power of any network node reaches zero.

Proposed integrated routing schemes, referred to as DFS Power and Cost Progress based algorithms, are combinations of known greedy power and cost-aware localized routing algorithms, and DFS the algorithm that guarantees delivery.

Experiments confirm the ability of the new set of power-aware algorithms to improve power and cost metrics and to guarantee the delivery due to the DFS routing framework. These routing strategies prove to be good routing algorithm candidates for future integration with adequate location update protocols and wakeup power-based management schemes, when satisfactory solutions to these important ad hoc wireless networks problems are found.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Since their implementation in the 1970s, ad hoc wireless networks have been growing with incredible speed. Their popularity significantly increased in the last decade, with impressive growth in the number of users, technologies, and applications.

Existing wireless networks and applications could be divided into two basic groups. The first kind includes structured wireless network built with fixed and wired gateways, known as base stations. Mobile units connect and communicate with the closest base station within their communication radius. As these units move, they change their home base stations and switch to another base station. This switch, the 'hand-off', happens in the background and is handled seamlessly by the network.

The second kind of wireless network is defined as infrastructurless network. There are no fixed routers or a central authority in these types of mobile networks. All nodes are capable of movement and their connectivity with other network nodes is dynamic and changing all the time. These networks are called mobile ad hoc wireless networks. The focus of this thesis work is limited to the simplified case, where network nodes do not move.

The architecture of ad hoc mobile wireless networks will enable many practical applications on the commercial market. Lack of fixed infrastructure would be typical in emergency situations, remote and environment monitoring, search and rescue operations, battlefields, radar networks, conferences, and other similar cases. Just recently, a whole set of possible applications have been more and more popular and various deployments have been executed in a so called wireless sensor networks. They are typically characterized as networks with nodes that do no move or have insignificant movement patterns. Quite often, sensor nodes might host an intelligence that is capable of processing exchanged data in a real time.

In addition to the set of emergency applications, there is a group of possible everyday implementations. It is expected that wireless ad hoc networks will easily make the biggest commercial penetration within the wireless local area networks. The standards already exist and they include support for some limited ad hoc connections, or peer to peer support.

Another possible application could include our everyday mobile wireless phones. However, possible successful deployment could be prevented for some other reasons. In this scenario, the wireless networks owners would lose a portion of the total traffic if these networks were based on the ad hoc routing solutions. In addition, end users would probably not easily accept the condition that their mobile phones are forced to forward other people traffic (security and battery power drain).

1.1 Ad Hoc Wireless Networks

Ad hoc networks are comprised of wireless hosts that communicate with each other without any static network interaction. Each network host is capable of communicating directly with other network hosts in its neighborhood. A message sent by a node will reach all neighboring nodes that are located at the distances that are less than the transmission radius. However, the need for communication in these network is not only limited to surrounding neighbors. On the contrary, any two network nodes could be involved in a message exchange. Due to limited transmission radius, the routes between nodes are usually formed through several hops.

The routing problem is the problem of finding a route for sending a message from a source point to a destination point. Each network node can be a source or a destination of the communication link. In addition, nodes are asked to participate in routing other nodes packets, i.e. to act as routers. The problem of finding and maintaining routes has proven to be a significant challenge in wireless ad hoc networks.

The satisfactory routing solution needs to be successful and to satisfy various requirements that are part of the routing problem in these networks. As previously mentioned, there is no central authority in ad hoc networks or the backbone infrastructure. Peer-to-peer communication happens, most of the time, through multi hops, and in most cases depends on other nodes' participation. Network nodes are required to be connected at all times to the network for guaranteed delivery to be feasible. This requirement, connected networks, is quite important for applications that are sensitive to data loss.

One possible solution to the routing problem would be the well-known shortest path algorithm. This non-localized scheme requires that each network node possesses global knowledge about the network. It is an unrealistic and hypothetical requirement for the global knowledge to be fully distributed to all network nodes. The performance of shortest path algorithm is used as a theoretical best possible routing performance.

The information about changes in network topology, in the shortest path scenario, due to node mobility or changes in node activity, would need to be distributed across the network. In addition to activity status and location updates for all nodes in the network, the updates on the status of every possible link in the network is needed to guarantee the availability of shortest path, which is unacceptable quadratic communication overhead.

The solution to the routing problem in wireless ad hoc networks lies in localized algorithms. These schemes are based on non global behavior. Each node makes routing decisions based only on the information about neighboring nodes and the position of the destination. The search for a proper solution to a complete routing problem is complicated with an additional requirement for the proposed solutions to handle two other present complex issues - mobility of network nodes and the changes in the availability of network nodes.

One of the initially proposed location-based processing was the greedy routing algorithm, where each node forwards messages to the neighbor that is closest to the destination. This algorithm proved not to be efficient for low degree graphs (a low average number of neighbors). However, it offered a performance that was close to shortest path hop count, whenever the routing of the message was successful.

The problem of finding an appropriate and efficient routing solution has proven to be quite complex to resolve. A simplification of the problem, routing in static ad hoc wireless networks, created several routing protocols that guarantee delivery. In most cases, these solutions have not been very efficient in terms of hop counts. Nor have they had the ability to satisfy a set of power and cost aware requirements, where routing algorithm has to minimize the power of routing tasks or maximize the total number of these routing tasks that network can perform.

1.1.1 Wireless Ad Hoc Sensor Networks

A special set of applications could be based on so called wireless sensor networks. In this scenario, wireless networks are made up of a number of sensor nodes that are positioned within a limited and defined geographical area. These networks might be asked to satisfy a wide spectrum of functional requirements. They could be required to be formed in a fast and ad hoc manner, manage individual node failures, communicate with central authority, support incremental deployments, be scalable, etc.

Their movement pattern is typically not present or considered insignificant. Each sensor represents a typical network node with wireless communication capabilities. They are required to communicate with immediate neighbors in order to support special types of applications. These applications might require sensor nodes to host a certain level of intelligence needed for signal and data processing.

A possible set of applications, based on these sensor hosts, is wide and could include the following:

- Wireless military sensor networks (used for detection and processing information about enemy movements, explosions, etc.).
- Wireless Sensor networks to monitor environmental changes in critical objects and areas (airplanes, forests, oceans, etc).
- Wireless sensor networks for detection and immediate processing of various possible chemical, biological, radiological or nuclear attacks or materials.
- Wireless surveillance sensor networks (support for security issues in kinder gardens, shopping malls, residential buildings, etc.).
- Wireless parking lot sensor networks (detection and monitoring of available parking spots).

1.2 Motivation for the Research

A localized routing algorithm, described in [BMSU], called FACE, guarantees the message delivery in connected unit graphs. Other localized routing solutions, based on FACE framework, have been proposed as well. The same paper, [BMSU], introduced an improved version of FACE algorithm, the so-called GFG (Greedy-Face-Greedy) routing algorithm. Their hop count performance was significantly improved by amendments introduced by [SD] authors.

Finding alternative and independent routing strategies to those discussed in [SD], which would offer better performance than the relatively successful FACE based algorithms, has been a challenge. This thesis will introduce and analyze a set of various DFS based routing strategies as a possible solution. Deeper analysis of their performance and how they compare to other successful approaches is presented as well.

Traditionally, in addition to the information about the delivery rate, hop count was the second most important parameter used to understand the quality of the routing solution. In these models, energy model was using the constant amount of power per hop.

[RM] proposed a different approach where consumed power, whenever a message is sent or received, is a function of the distance d between two network nodes:

$$u(d) = d^{\alpha} + c$$

for some constants α and c. This new model opened a possibility of replacing the legacy constant metric with a new, power-based metric. [SL] authors proposed new power and cost aware approaches, based on this new model. Newly proposed fully distributed routing algorithms were based on power, cost and power-cost metrics. Nodes make routing decisions solely on the basis of location of their neighbors and destination. Power-aware localized routing algorithm attempts to minimize the total power needed to route a message between a source and a destination. Cost-aware localized algorithm tries to maximize the number of messages that the network can route before any node's battery dies. The combination of power and cost localized algorithms attempts to minimize the total power needed and attempts to avoid nodes with battery with a shorter lifetime.

Just recently, a new set of so called progress-based routing algorithms was introduced by [KNS3]. These algorithms, derived from [SL] algorithms, are based on the notion of proportional progress. They appear to be simpler than the schemes that were initially introduced and they have better or the same performance as algorithms initially introduced by [SL]. The common element for all of these power-aware algorithms is that they do not have good delivery rates for sparse graphs. This thesis addresses this problem by introducing a new set of power- and cost-aware algorithms, which are built upon a framework of DFS guaranteed delivery routing.

1.3 Main Contribution of the Research

This thesis has generated two direct results. The first one is an introduction of a set of DFS based routing algorithms. The best selected DFS routing candidates from the first step are then used for a new set of DFS and power aware routing algorithms.

In addition, one of the contributions of this thesis has already been published as [SRV1]. One or two more articles are expected to be submitted for further publishing as a direct result of this work and proposed future research around DFS routing improvements.

Depth First Search (DFS) routing algorithm:

DFS (Depth First Search) routing scheme is defined and introduced. It is the first localized algorithm that guarantees delivery for (connected) ad hoc wireless networks even in a case of inaccurate location information for the destination node. This is enabled by an algorithm internal design that is built upon the fact that DFS is not a stateless routing scheme. Memorizing information about past traffic, at the node level, helps the routing logic to keep switching between greedy and recovery modes, providing the message delivery even in a case of invalid location information.

Hop count performance of the initial DFS algorithm was not too far from the shortest path benchmark for a well-connected network. However, sparse networks, with network density parameter values less or equal to six (or seven), left room for further improvements. This thesis investigated several possible ways to resolve the issue of DFS hop count performance.

Various approaches were researched and applied. Several definitions of dominating set and internal nodes were used in these efforts. The basic principle of dominating set theory, the reduction of the network to the subset of nodes that belong to the dominating set, is the core of the improvement strategies that were used. DFS is extended to route messages only across internal nodes that form dominating sets. Dominating sets, as proposed and defined by [SSZ, WL], were based on one of the selected set of internal nodes: intermediate, intergateway and gateway.

Another suggestion for enhancement of the DFS routing protocol was a special enhancement request to the internal design of DFS. The idea of this proposal is to avoid communication between neighboring nodes that should not happen, since this knowledge could be extracted from the past traffic. A set of simulation experiments demonstrated associated improvements of the proposed amended routing scheme.

This small change in the DFS internal design imposed an additional request on the way network nodes remember and monitor ongoing traffic. In order to support proposed improvements, each node is required to create and maintain a list of neighboring nodes that should be ignored for the ongoing routing task of the message. The impact of memorizing past traffic and the memory requirements imposed on network nodes are separately discussed in this paper.

All proposed versions of DFS routing schemes are simulated in various types of networks. The performance patterns were confirmed and conditions that influence and trigger them have been analyzed with more detail. An associated shortest path hop count metric was used as a benchmark. The best DFS routing candidates were chosen for different scenarios (network sizes and network density values) and explained.

DFS and Power and Cost Aware progress based routing algorithms:

A simulation phase on a proposed set of DFS routing algorithms helped the best DFS routing candidate(s) to be identified. This thesis proposes a new set of routing algorithms that are created by integrating the best DFS routing candidate with a recently proposed version of progress-based power and cost-aware routing algorithms.

The major difference between DFS and power-aware DFS protocols is that nodes use different priority metrics for sorting best neighbors. Instead of distance to destination, the appropriately defined power- or cost-aware metric is used.

The goal here is to leverage the best of both worlds. A new set of algorithms is required to guarantee the message delivery, as promised by the DFS framework. At the same time, power- and cost-aware performance of new routing schemes is expected to improve these characteristics of the selected DFS routing candidate.

An extensive set of simulation experiments have been conducted. The new set of algorithms have shown important improvement in power and cost aware performance in comparison to DFS routing schemes. The power required to route the message is significantly improved over the initial version of the DFS algorithm. As expected, the cost-aware performance, the ability of the network to route as many routing tasks as possible, is achieved over the cost-aware performance of the best DFS routing candidates.

1.4 Simulation Platform

The full suite of experiments was developed in order to determine the performance of DFS routing algorithms and proposed improvements. They helped to gain deeper understanding of the performance metrics of a newly introduced set of protocols. A simulation platform was built and enhanced to support various simulation requests. These are the most important functionalities offered by used simulation framework:

1. Generates graphs from given wireless ad hoc network parameters e.g. maximum number of nodes, the size of network area, random selection of x and y coordinates of network nodes, network density value (d).
2. First suite of test cases independently executes all proposed improved DFS algorithms. Included is the calculation of the hop count performance to the non-localized shortest path problem (for randomly generated ad hoc wireless networks).
3. Second suite of test cases covers power- and cost-aware experiments. Both power models, [RM] and [HCB] were tested. The power- and cost-aware performance of new protocols was compared to the power/cost aware performance of various DFS algorithms.
4. Generates reports on hop counts, consumed power for routing tasks, the maximum number of routing that network is capable of executing before any network node exhausts its battery.

1.5 Conditions, Assumptions and Limitations

In a contrast to many existing routing solutions, DFS routing algorithms do not require complete information about the destination node for its guaranteed delivery capability. The DFS routing framework will enable the delivery even if the destination knowledge is not correct.

This thesis did not take into account the mobility of network nodes. The research work was simplified by analyzing a special case, ad hoc wireless networks with nodes that do not move. As discussed in [S-lu], current location update approaches have important deficiencies and no satisfactory solution to this problem has yet been found. The focus of the research was the introduction of the DFS routing algorithm and a set of improvements to the initial DFS scheme within the context of wireless ad hoc networks, with nodes that do not move. Similarly, a proposed set of power- and cost-aware algorithms, based on the best chosen DFS routing candidate, were thoroughly simulated and monitored in this simplified environment as well.

As discussed in this paper, it is presumed that integration of DFS power-aware algorithms with satisfactory location update scheme is not going to be difficult. The ability of DFS to guarantee the delivery even when location information is not fully correct will play major role within the mobility context of the routing problem. That condition, the inaccuracy of the location information due to the node mobility, is going to be present often and the ability of routing solution to handle that issue will have a critical impact on the routing performance.

1.6 Thesis Organization

This thesis is composed of seven chapters and three appendices. This chapter, Chapter 1, gives the executive summary of the whole research work. The routing problem in wireless networks is defined. A few relevant existing solutions are presented with some of their main deficiencies. These inadequacies were the main motivating force for the research. Both contributions of the thesis work are presented and discussed.

Chapter 2 provides a discussion of research background and related work. A literature review of relevant results and solutions for the routing problem is included as well. Besides a few of the most important papers in the space of wireless ad hoc networking, existing solutions and findings in a few other areas are discussed as well. These include the research in a space of position-based routing algorithms, the theory of dominating sets, and power- and cost-aware routing algorithms.

Chapter 3 presents the first focus of the thesis work, Depth First Search algorithm. DFS routing is introduced and analyzed. Various improvements of the initially proposed routing scheme are discussed and proposed changes to internal design details of the DFS protocol are examined.

Chapter 4 introduces a new set of routing algorithms. They are generated by integrating the best DFS routing candidate with progress based power and cost aware localized routing scheme. This chapter includes a separate section that discusses relevant assumptions and conditions for this research work.

Chapter 5 discusses the simulation efforts. The implemented network simulator and results of various simulation experiments are presented here.

Chapter 6 summarizes the conclusions of this research and offers suggestions for further investigation. It is followed by Chapter 7, which includes a list of used references.

Chapter 2

Background and Related Work

The first and second sections discuss relevant work in the area of routing problems and associated qualitative and quantitative independent metrics.

A subset of routing solutions, based on the available location information, as provided by GPS enabled devices, is presented in section 2.3. These position-based algorithms have proved to be effective solution to the routing problem.

A new and innovative approach to the routing problem, based on the theory of dominating sets, is presented in the section 2.4. Some existing dominating set definitions are presented and the way they are constructed. Various research papers covered the topic and confirmed the significant performance improvements of routing solutions. QoS problem and proposed QoS routing solution are discussed in section 2.5.

A different approach to judge the performance of existing and new routing solutions is discussed in section 2.6. The conventional way of analyzing potential routing solutions, the hop count, is supplemented with two new metrics, power and cost aware. They enable additional insights in the quality of proposed routing solutions. These metrics are new tools in comparing and locating the best possible routing answer.

2.1 Routing Problem in Wireless Ad Hoc Networks

Mobile ad hoc networks, often referred to as MANET networks, consist of wireless hosts that communicate with each other in the absence of a fixed infrastructure. Mobile ad hoc wireless networks are best modeled by a *unit graph model.*

The *unit* graph model could be defined in the following way. Two nodes *A* and *B* in the network are neighbors and joined by an edge if the Euclidean distance between their coordinates in the network is at most *R,* where *R* is the transmission radius which is equal for all nodes in the network. A variation of this model includes unit graphs with obstacles (or subgraph of unit graph), minpower graphs where each node has its own transmission radius and links are unidirectional or allowed only when bidirectional communication is possible. However, no credible research was done in literature on any model other than a unit graph model (one important exception in [BFNO]). A special case would include power and cost savings and congestion-aware methods. In these routing algorithms nodes may adjust their transmission power to merely reach an intended receiver.

Figure 2-1 gives an example of a unit graph with transmission radius as indicated. This thesis, which proposes DFS-based power-aware algorithms, was exclusively based on the unit graph model.

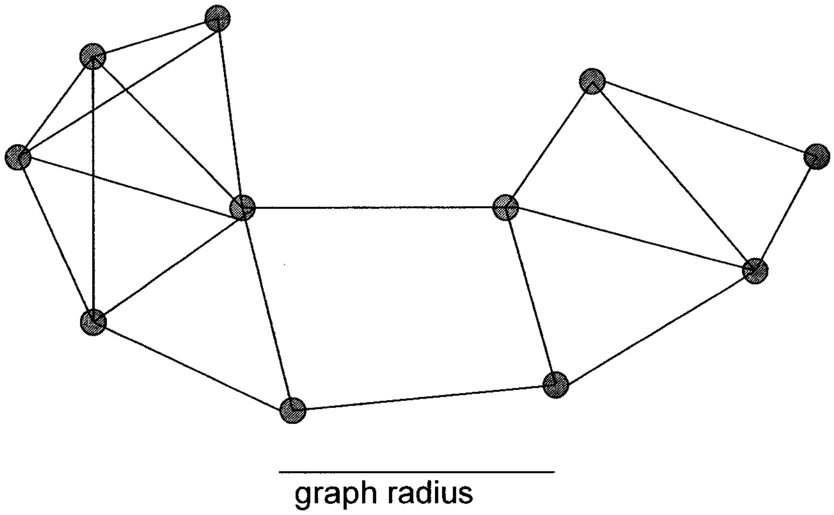


Figure 2-1 ***Unit graph and graph radius***

The routing problem centers on finding a route to send a message from a source to a given destination. Routing becomes very difficult in wireless networks. In addition, the proposed routing solution has to satisfy handful qualitative and quantitative characteristics, as it is discussed in the section 2.2. In highly mobile situation, the flooding scheme might be the only reliable solution for sending data packets. Since the link channel resource is very scarce and battery power is limited, more efficient schemes must be devised for low and moderate mobility rates. Numerous routing protocols have been proposed in recent years that try to offer a routing solution that would satisfy desired features – just to name a few: guaranteed delivery, hop count, power consumption, cost aware performance, loop freedom, and others.

2.2 Routing Algorithms Adopted Metrics

Macker and Corson [MC] discussed qualitative and quantitative independent metrics for judging the performance of routing protocols. Desirable qualitative properties [MC] include: *distributed operation*, *loop-freedom, demand-based operation* and *'sleep' period operation.* Hop count and delivery rates are among quantitative metrics. These properties and metrics are further discussed in this section, with a special attention to the actual thesis contribution, which is a new set of DFS-based power and cost-aware routing solutions. The goal of the research is to design a set of routing protocols with the following properties.

a) Minimize the energy required per routing task: Hop count was traditionally used to measure energy consumption of a routing task. This model was based on constant power metric per hop. However, if nodes can adjust their transmission power (based on the location of their neighbors) then the constant metric can be replaced by a power metric that depends on distance between nodes [RM, HCB]. The distance between neighboring nodes can be estimated on the basis of incoming signal strengths or during the initialize phase (exchange of 'hello' messages). Relative coordinates of neighboring nodes can be obtained by exchanging such information between neighbors [CHH]. In addition, the location of nodes may be provided by small GPS (Global Positioning System) devices that communicate with a satellite. This thesis work is based on the location-based routing. Therefore, it is assumed that location information is available to all network nodes.

b) *Loop-freedom*: The proposed routing protocols should be inherently loop-free, to avoid timeout or memorizing past traffic as cumbersome exit strategies.

c) *Maximize the number of routing tasks that network can perform*: Some nodes participate in routing packets for many source-destination pairs, and the increased energy consumption may result in their failure. Thus a pure power consumption metric may be misguided in the long term [SWR]. A longer path that passes through nodes that have plenty of energy may be a better solution [SWR].

d) *Minimize the communication overhead:* Due to limited battery power, the communication overhead should be minimized if the number of routing tasks is to be maximized. Proactive methods that maintain routing tables with up-to-date routing information or global network information at each node are certainly an unsatisfactory solution, especially when node mobility is high with respect to data traffic. For instance, shortest path based solutions are too sensitive to small changes in local topology and activity status (the later even does not involve node movement).

e) *Avoid memorizing past traffic or route*: Solutions that require nodes to memorize route or past traffic are sensitive to node queue size, changes in node activity and node mobility while routing is ongoing (e.g. monitoring environment). Ideally, memorizing past traffic should be avoided if possible. However, the need to memorize is not necessarily a demand for new significant resources. The impact of a memorization request on the proposed routing solution needs to be analyzed, if such a memorizing feature is required to be present.

f) *Localized algorithms*: Localized algorithms [EGHK] are distributed algorithms that resemble greedy algorithms. This is the case where simple local behavior achieves a desired global objective. In a localized routing algorithm, each node makes a decision regarding which neighbor to forward the message to based on its own location, and the location of its neighboring nodes and the destination. While neighboring nodes may update each other location whenever an edge is broken or created, the accuracy of destination location is a serious problem. In some cases, such as monitoring environment by sensor networks, the destination is a fixed node known to all nodes (i.e. a monitoring center). The proposed routing algorithms in this thesis are directly applicable in such environments. All non-localized routing algorithms proposed in literature are variations of shortest weighted path algorithm (e.g. [CN, LL, RM, SWR]).

g) *Single-path routing algorithms:* The task of finding and maintaining routes in mobile networks is nontrivial since host mobility causes frequent unpredictable topological changes. Most previously proposed position-based routing algorithms (e.g. [BCSW, KV]) for wireless ad hoc networks were based on forwarding the actual message along multiple paths toward an area where the destination is hopefully located, hoping to achieve robustness. However, it has been shown that single-path strategies may be even more robust. [BMSU] proposed solutions guarantee delivery do that with less communication overhead. The significant communication overhead can be avoided if a variant of a source-initiated on-demand routing strategy [BMJHJ, RT] is applied. In this strategy, the source node issues several search 'tickets' (each ticket is a 'short' message containing sender's *id* and location, destination's *id* and best known location and time when that location was reported, and constant amount of additional information) that will look for the exact position of the destination node. When the first ticket arrives at the destination node *D, D* will report back to the source with brief message containing its exact location, and possibly creating a route for the source. The source node then sends full data message ('long' message) toward exact location of the destination. The efficiency of the destination search depends on the corresponding location update scheme. A quorum-based location update scheme is being developed in [S-q]. Other schemes may be used, with various trade-offs between the success and flooding rates (including an occasional flooding). If the routing problem is divided as described, the mobility issue is algorithmically separated from the routing issue, which allows us to consider (in this thesis) only the case of static networks with known destination in our algorithms and experiments. The choice is justified whenever the destination does not move significantly between its detection and message delivery, and information about neighboring nodes is regularly maintained. Yet another routing method may forward messages toward imprecise destination locations, hoping that closer nodes will locate the destination more accurately.

h) *Guaranteed delivery:* The proposed localized algorithms [SL] achieve a very high delivery rate for dense networks, but low delivery rate for sparse networks. In this thesis, a new set of routing protocols have been designed. These power, cost, and power-cost routing algorithms guarantee delivery, for arbitrary unit connected graph. The message delivered is guaranteed even if destination information is not correct.

2.3 Position Based Routing Algorithms

Position-based algorithms have been proven to be a scalable and efficient solution to the problem of routing in ad hoc wireless networks. This research has only focused on these types of routing algorithms. DFS and discussed variations of power- and cost-aware protocols are examples of position based routing schemes.

It is assumed that correct position information is available to each network node as provided by means of described GPS-enabled devices. It is expected that each mobile entity would be provided with location information, latitude, longitude and if needed, height. The accuracy of the information has increased significantly and currently it is set to only a few meters.

In addition, it is accepted that network nodes have knowledge of relative coordinates of neighboring nodes. This could be achieved in several ways. For example, an exchange of 'hello' messages between neighboring nodes could help this information to be established in the network. The alternative way would be based on estimates due to the analysis of incoming signal strengths, in networks with a constant power transmission energy, or looking into time delays in direct communications.

When required to send a packet to an arbitrary node in the network, each node, besides the knowledge of its location and neighboring relative distances, is assumed to know the location of a destination node as well. The premise that the source network node knows the correct destination information is a simplification of a routing problem in real ad-hoc networks. A quality of localized position-based routing is that routing decisions at each node does not require the global knowledge. It is only based on the node's neighbors' information and the destination information. It does not require the establishment or maintenance of routes through the network.

The reliability of location information, the destination and other network nodes, is evidently questionable due to many reasons. The main two include mobility of network nodes and possible ad hoc network enforced sleep scheduling protocol.

Mobility in ad hoc networks can cause frequent unpredictable topological changes. Various location update schemes have been developed, none of which provide a satisfactory solution. A good review of existing location update schemes is presented in [S-lu]. [KFWM] recently proposed a hierarchical location service as a good potential candidate for the solution to mobility problem. However, no complete and full solution has been presented yet. The problem has proven to be quite difficult to resolve.

An additional problem to be considered is the node activity, where nodes can randomly decide to go offline, or 'sleep' mode. [T] presents a good overview of currently available research solutions for wake-up based power management schemes and concludes that none of existing power-save management approaches offer a satisfactory solution. The survey structures all available solutions into three categories: scheduled rendezvous protocols, asynchronous protocols, and on-demand protocols and suggests potential directions for future work. Apparently, power management in ad hoc wireless networks and finding a distributed protocol appears to be a complex and challenging problem as well.

2.3.1 Relevant Routing Algorithms

Finn [F] proposed a greedy routing scheme that chooses the successor node that makes the best progress toward the destination. The distance to destination is used to measure the progress. The algorithm fails when no node is closer to the destination than the current node that holds the message. The author proposes a special based flooding mechanism, targeted at n-hop neighbors. The goal is to reach the neighbor that is closer to the destination than the node that currently holds the message (at has no neighbor that is closer to the destination).

A similar protocol, a variant of greedy based algorithm, called GEDIR, is presented in [SL1]. The algorithm stops if the best choice for the current node is to return the message to the previous node.

Several other existing location-based routing algorithms are reviewed in [SL1]. Two of them are compared with the *GEDIR* algorithm performance. One of methods, called *MFR* (most forward with progress), introduced in [TK], was comparable to *GEDIR* (it chooses the same route as *GEDIR* in over 90% of cases). However, the algorithm is conceptually more sophisticated and requires more power, as proved in [SL]. The other algorithm, based on direction of edges [BCSW, KV, and KSU], is not loop-free [SL1] and does not perform better in terms of success rates or hop counts, as proved in [SL].

2.3.2 GRA – DFS Alike Routing Algorithm

[JPS] proposed, independently of our research, a routing strategy called *geographic routing algorithm (GRA)*. Network nodes are required to partially store routes toward certain destinations in routing tables. The *GRA* routing algorithm applies a greedy strategy while the message is forwarded. This independent work and its contributions are similar to the thesis proposed routing algorithm, Depth First Search (DFS).

When the message reaches node C that it is closer to the destination D than any of its neighbors, the routing logic switches from a greedy mode to a recovery mode. The packet is 'stuck' at node C and the greedy routing logic has to change in order to resolve the problem. Under this condition, it starts the route discovery protocol. The route discovery discovers a path from C to D. It updates the routing tables toward D at any node on the path with this information. Upon successful completion of the route, the packet, currently stopped at node C, can be routed along newly discovered route from C to D.

The authors propose two route discovery strategies: *breadth first search* and *depth first search (DFS)*. The breadth first search is based on flooding mechanism. In contrast, the *DFS* approach generates a single acyclic path from S to D. Each node puts its name and address on the route discovery packet p. Then it forwards p to a neighbor who has not seen p before. This neighbor is one of node's neighbors which minimizes

$$d(S,y)+d(y,D)$$

where $d(x,y)$ is Euclidean distance between nodes x and y.

The packet will be returned to the node from which it originally received if a node is not able to forward the packet. This node's information, the address and the name, are not going to be preserved at the packet. If a same packet is received for the second time, it is immediately rejected. Route discovery packets are required to be kept for some time at network nodes to enable the the node to reject the same packet. The DFS protocol, as presented in [JPS], strives to build and maintain the routing table embedded within the packet. The authors investigate routing table sizes and present methods for taking into account positional errors, node failures and mobility.

2.4 Dominating Sets and Internal Nodes

Routing based on the dominating set approach, as presented and defined in [SSZ, WL], offers an innovative and promising improvement for potential solutions to the ad hoc network's routing problem. A set of nodes, a subset of all network nodes, is considered to be a dominating set if all nodes in the network are either in the set or are direct neighbors of nodes in the set. Nodes that belong to the dominating sets are called internal nodes. Different definitions for dominating sets generate different set of internal nodes.

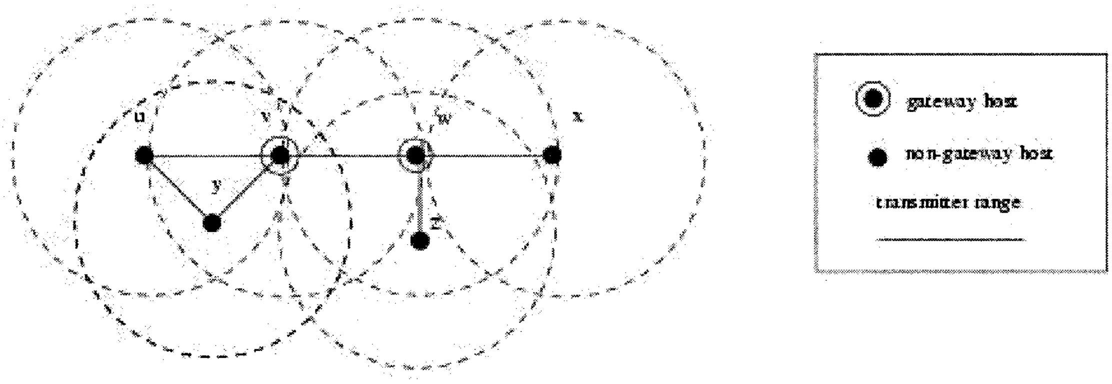


Figure 2-2 ***Ad hoc wireless network dominating set and its internal nodes***

The substance of this approach is to exclusively use internal nodes for routing of messages throughout the ad hoc network. The consequence is that the search space for a route is reduced to corresponding internal nodes. If a source is not an internal node, it forwards the message to one of its neighboring internal nodes. This internal node represents a new source point of the routing problem that happens only along a reduced graph based on internal nodes of the network. As routing gets to the end of the route, the message reaches the internal node that is either the destination node or a direct neighbor of the destination node. In a later case, an additional forward toward the destination is required.

A direct result is a significant reduction of the communication overhead. A dominating set, as defined in [SSZ], was able to generate the reduction in a communication overhead of ≤ 53% on random unit graphs with 100 nodes for most values of average degrees d. The reduction falls to ≤48% when $d \leq 3$. For values of d that fall in a range [10, 20] the reduction is ≤40%.

[SSZ] authors proved with their experiments that routing based on dominating sets is better than non-dominating set routing in all analyzed aspects: reliability, rebroadcast savings, and maintenance communication overhead. In addition, it has been shown that these dominating set based algorithms are simple, efficient and distributed (localized).

[WL] proposed a simple and efficient distributed algorithm for calculating connected dominating set in ad hoc wireless networks. The latest contributions to the way dominating sets are calculated and formed are examined in 2.4.1, New Definitions of Dominating Sets, where [SSZ, S1] introduce a major breakthrough and improvement in a way these dominating sets are calculated and formed.

[WL] discussion introduced the concept of an intermediate node. A node A is an intermediate node if there exist two neighbors B and C of A that are not direct neighbors themselves. As presented on Figure 2-3 Intermediate, intergateway and gateway nodes as proposed by [WL] Figure 2-3, nodes *C* and *K* are not intermediate nodes, while other presented nodes can be classified as intermediate.

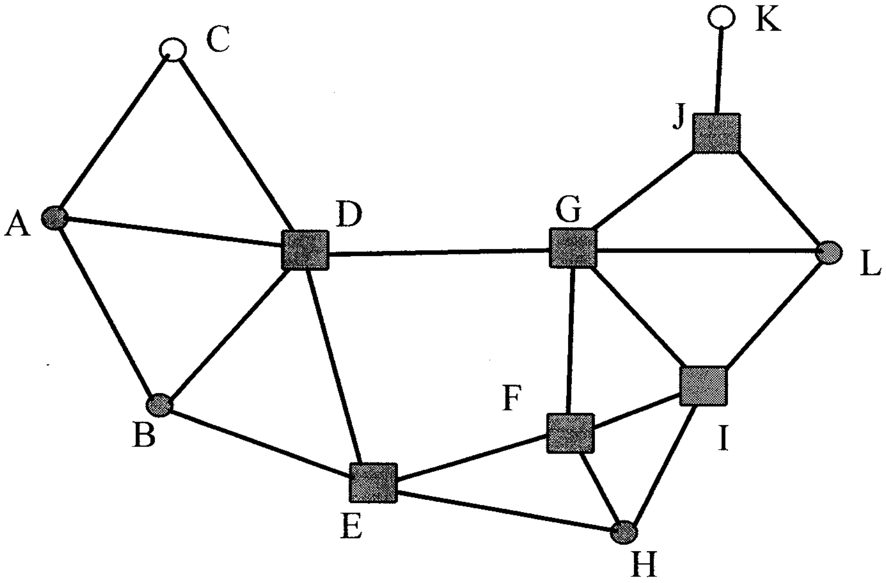


Figure 2-3 ***Intermediate, intergateway and gateway nodes as proposed by [WL]***

[WL] Rule 1 defines intergateway nodes. *N[a]* = *N(a) U {a}* defines closed set neighbor set of node *a* while *N(a)* denotes an open set of node *a*. Each node is assigned an unique number, *id(a)*. One possible option is to adopt the location coordinates, *x* and *y*. When two nodes *id* numbers are compared, we first compare *x* coordinates. If they are equal, we then compare the *y* coordinates.

Let us consider two intermediate nodes *v* and *u*. If $N[v] \subseteq N[u]$ in *G* and $id(v) < id(u)$, then node *v* is not an intergateway node. As presented on Figure 2-3, nodes *A* and *B* are covered by node *D*, node *H* is covered by node *F*, and node *L* is covered by *G*. The remaining six nodes are inter-gateway nodes.

[WL] Rule 2 defines gateway nodes. Look at u and w that are two inter-gateway neighbors of a inter-gateway node *v*. If $N(v) \subseteq N(u) \cup N(w)$ in *G* and $id(v) = \min \{id(v), id(u), id(w)\}$, then node *v* is declared a non-gateway node. As presented on Figure 2-3 Intermediate, intergateway and gateway nodes as proposed by [WL] Figure 2-3, all inter-gateway nodes remain gateway nodes. Node *E* is 'covered' by *D* and *F*, but *D* and *F* are not connected themselves. Although all neighbors of node I are neighbors of either *F* or *G*, it does not have lowest *id* (in this example, x coordinate serves as *id*).

2.4.1 New Definitions of Dominating Sets

[S1] paper is the latest correspondence and the extension of the work and ideas outlined in [SSZ]. It includes more details and relevant improvements to the dominating set approach outlined in [SSZ]. This thesis and concurrent experiments were based on the dominating set definition, as proposed and defined in these two associated papers.

[SSZ] proposed improving [WL] rules for building dominating sets by replacing node ids with a record: $key = (degree, x, y)$

where *degree* is the number of neighbors of a node and *x* and *y* are its two coordinates. In both [WL] rules, it is suggested that both [WL] rules be modified so nodes would first compare their degrees and node with higher degree would have greater changes of remaining an internal node. In a case of ties, x-coordinates are used to resolve it or y-coordinates (in a case of x-coordinates ties).

Initial definition of dominating sets, outlined in [WL], required that each node acquires distance-2 neighbors knowledge in order to determine its dominating set status, i.e. if the node belongs to the dominating set or not. [SSZ, S1] propose a major breakthrough in the way dominating sets are created. Each node is able to determine its 'status', that could be one of the three possible: intermediate, intergateway and gateway, based only on location information of itself and its immediate neighbors. The location information of a node's neighbors is gathered through usual 'hello' message exchanges, when the network is initialized for the first time.

What is a new in [SSZ, S1] proposal is that process of determining dominating set status of the ad hoc network nodes requires zero communication overhead. This is a significant improvement over previous ways of building dominating sets. As presented in [S1], the following is the logic that each node would execute in order to determine its dominating set status:

- A node is marked as an intermediate node if it has two unconnected neighbors.
- A node *A* is covered by neighboring node *B* if each neighbor of *A* is also neighbor of *B*, and $key(A) < key(B)$. Nodes not covered by any neighbor are defined as intergateway nodes.
- A node *A* is covered by two connected neighboring nodes *B* and *C* if each neighbor of *A* is also neighbor of either *B* or *C* (or both), $key(A) < key(B)$, and $key(A) < key(C)$. An intermediate node not covered by any neighbor becomes an intergateway node. An intergateway node not covered by any pair of connected neighboring nodes becomes a gateway node.

Upon completion, intergateway and gateway nodes of the ad hoc network form a dominating set as proven in [S1]. It is interesting and important to note that a dominating set generated by [SSZ, S1] is identical to a dominating set generated by [WL].

This is the actual procedure, used by a network node, to determine its internal dominating set status:

```
Procedure internal-status(v);
    intermediate(v) = intergateway(v) = gateway(v) = false

    for each neighbor u of v, u != v do
        for each neighbor w of v, w != v do
            if d(u;w) > R then intermediate(v) = true
    if intermediate(v) then {
        intergateway(v) = true;
        for each neighbor u of v, u != v do {
            covered = true;
            for each neighbor w of v, w != v;w != u do
                if d(u;w) > R or key(v) > key(u) then
                    covered = false;
        if covered then intergateway(v) = false }
    if intergateway(v) then { gateway(v) = true;
        for each neighbor u of v, u != v do
            for each common neighbor w of v and u do {
                covered = true;
                for each neighbor z of v do {
                    if (d(z; u) > R and d(z;w) > R) or
```

```
            key(v) > key(u) or
            key(v) > key(w)
          then covered = false;
    if covered then gateway(v) = false }}
```

2.4.2 Fixed Dominating Sets and Mobility

Another important issue related to the dominating sets is the fixed selection of internal nodes. Stationary selection of internal nodes for solving routing problems in ad hoc networks would easily create a special scenario. In this case, only selected network nodes, belonging to dominating sets, will retransmit the message, which will quickly reduce their battery power levels.

This thesis does not consider mobility of nodes. However, this is one of the rare cases in the ad hoc wireless routing problem sphere, where mobility would actually help in the potential routing solution, in terms of prolonging nodes and network's life. Any movement of nodes will request that the internal node status is recalculated. As a direct consequence, the network would select new nodes as its internal nodes and power savings would happen at nodes that lost such status.

An interesting proposal for solving this problem, caused by stationary selection of internal nodes, was presented in [WDGS]. The authors designed new algorithms for selecting dominating sets, which take node battery power into consideration as well. A node power level, in addition to its degree, determines if the node will be a part of dominating set. Nodes having more remaining power will be more likely to be in the dominating sets, and the status may change at some threshold values. The change is therefore triggered by change of the power status, not by the node mobility. The proposed scheme proved to balance the overall energy consumption in the network and generate a relatively small connected dominating set.

2.5 QoS Routing in Wireless Networks

Various QoS issues in wireless network routing were discussed by Wang and Crowcroft [WC] with respect to support of multimedia applications such as digital video and audio. They examined the basic problem of QoS routing, namely, finding a path that satisfies multiple constraints, and its implications on routing metric selection. The authors presented three path computation algorithms for source routing and hop-by-hop routing.

Delay and bandwidth were identified as two most important QoS routing metrics [WC]. In addition, loss probability, cost, and delay jitter were discussed and presented. [WC] defines the bandwidth of a path as the minimum of the residual bandwidth of all links on the path or the *bottleneck bandwidth.*

As per [WC], the delay is made up of two basic components: queuing delay and propagation delay. Queuing delay is determined by bottleneck bandwidth and traffic characteristics. Since queuing delay is already reflected in the bandwidth metric, we only need to consider propagation delay in the delay metrics [WC]. This provides that these two metrics are not interdependent. Bottleneck bandwidth and propagation delay can be viewed as the width and length of a path. The problem of QoS routing is then to find a path in the network, between source-destination nodes, that will satisfy given QoS constraints: delay and bandwidth, i.e. the path length and width.

2.6 Power and Cost Aware Localized Routing Algorithms

A good overview of available power and cost-aware proposed solutions to power and cost aware routing problems in wireless ad hoc networks is presented in [LSR]. In this chapter we present the most relevant work that marks this research area and is directly used in this research.

2.6.1 The Power Aware Problem

The problem of managing and understanding the limitations of the power of various mobile devices is a relatively simple one for a basic type of users (i.e. standalone mobile users). These devices are typically equipped with lithium-ion, rechargeable batteries. Their lifetime is known and predictable, with a current lifetime of a few hours of active workload and about several days of idle time. However, this problem has a completely different substance when the same device becomes a part of a wireless ad hoc network .

This section discusses a group of routing algorithms that is designed and analyzed with different metrics in mind. Most routing protocols are compared based on their hop count and one of the key routing algorithm goals is to minimize this hop count (or the delay).

However, in certain networks, described common power model might not be accepted. Examples would include non-homogenous networks, where network nodes are dispersed in the space with big gaps between certain clusters (or individual nodes). Figure 2-4 displays the example and explains how distance between remote nodes forces us to redesign the legacy approach where the transmit power level was constant at all network nodes.

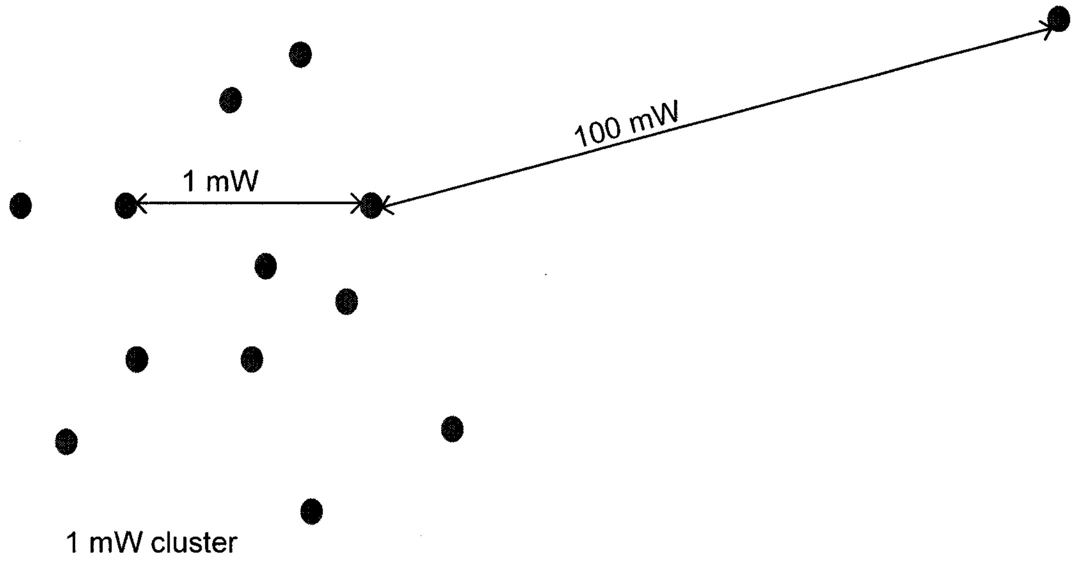


Figure 2-4 ***A common vs. adjusted power level in ad hoc wireless networks***

The approach of controlling and adjusting transmit power levels is beneficial even in ad hoc networks that have more homogeny distribution. It is obvious that multiple hops will generate a saving in the consumed power over the case where the power is adjusted to reach the most distant neighbor (or the destination). The power is kept down due to lowering the link length. With this approach, nodes are allowed to adjust their transmission power to just reach a desired receiver node, in order to preserve the energy. This is feasible based on a help of other network nodes, who are willing to forward other network nodes traffic.

Evidently, at the same time, network nodes are asked to keep forwarding other people's traffic. This keeps draining their battery power along the way. All of this has complicated power aware routing problem. Apparently, there is a need to make routing protocols power aware which is going to minimize the total energy needed to route messages throughout the network.

In addition, the design of power- and cost-aware routing protocols is complicated. The issue of control messages in the network needs to be analyzed, since these messages will drain the battery levels as well.

Power-aware routing algorithms are designed with the goal of minimizing power per routing task. These algorithms tend to select well positioned neighbors that will minimize the consumed power for the ongoing routing task.

Cost aware routing algorithms maximize the lifetime of the network, i.e. the network is considered to be available as long as all nodes are present (and their battery levels are not exhausted). Within this routing problem, nodes with more remaining power are preferred to other network nodes.

Power-cost aware routing algorithms are the combination of proposed power and cost aware schemes. In a nutshell, these algorithms will choose power efficient paths among cost optimal ones.

2.6.2 Power Aware Algorithm

[RM] suggested a model for the power consumption between two nodes at distance r to be:

$$u(r) = r^{\alpha} + c$$

where α ($2<\alpha<5$) and c are some constants. In their experiments they used the modified model:

$$u(r) = r^4 + 2*10^8$$

This model is referred to as the RM Model. The power-aware routing algorithm which they outlined runs in two phases. In the first phase, each node searches for its neighbors and selects these neighbors for which direct transmission requires less power than if an intermediate node is used to retransmit the message. This defines so-called enclosure graph. In the second phase, each possible destination runs distributed loop-free variant of (non-localized) Bellman-Ford shortest path algorithm and computes shortest path for each possible source.

[HCB] proposed a slightly different model. The total power for transmission and reception, for nodes at distance r, is set to be:

$$u(r) = r^2 + 2*1000$$

Their model is referred to as HCB Model. Signal attenuation was used to design an energy efficient routing protocol for wireless microsensor networks, where the destination is fixed and known to all nodes. They propose to utilize a 2-level hierarchy of forwarding nodes, where sensors form clusters and elect a random cluster head. The cluster head forwards transmissions from each sensor within its own cluster. It is shown that this scheme saves energy under some conditions. However, this clustering approach needs significant communication overhead, the routing algorithm is not localized, and the destination is not necessarily fixed.

[SL] introduced and discussed localized power aware algorithms. They are at the center of the thesis work – a variant of these algorithms is expanded and modified. A source node, or an intermediate node, that holds a message selects one of its neighbors with the goal of reducing the total power needed for the message transmission toward the destination. This is how the authors model the power consumption for the routing task. Let us assume that node A is a neighbor of node B and adopt the following:

$r=|AB|$, $d=|BD|$, and $s=|AD|$

The power needed for transmission from B to A, based on RM model:

$$u(r) = ar^{\alpha} + c$$

and the assumed power needed to route the message from node *A* to a destination *D*, is assumed to be:

$$v(s) = sc(a(\alpha - 1)/c)^{1/\alpha} + sa(a(\alpha - 1)/c)^{(1-\alpha)/\alpha}$$

For α=2, this power becomes:

$$v(s) = 2s(ac)^{1/2}$$

The proposed algorithms are looking for nodes in the network that are closest to the optimal desirable position. In this particular case, node *B* will select one of its neighbors, node *A*, which will minimize:

$$p(B, A) = u(r) + v(s)$$

Conducted experiments, as per [SL], confirmed the validity of the approach. This is the actual localized power aware routing algorithm:

```
Power-routing(S,D);
  B:=S;
  Repeat
    B:=A;                    (* do not set the first time *)
    Let A be neighbor of B that minimizes
      p(B,A)=u(r)+ tv(s);
    Send meesage to A
  until A=D (* destination reached *) or A=B (* delivery failed *);
```

The optimal power-saving algorithm, that minimizes the total energy per routing task for a message, can be calculated by applying Dijkstra's single-source shortest weighted path algorithm. Each edge has the weight $u(d) = ad^{\alpha} + c$, where d is the length of the edge. We refer to this algorithm as SP Power algorithm (shortest path) and it is mainly used to compare the performance of various new localized algorithms.

2.6.3 Cost Aware Algorithm

[SWR] proposed the usage of a function $f(A)$ that would indicate a node's reluctance to forward the packet, i.e. to be a router point for the specific network route. This is suggested reluctance function:

$$f(A) = 1/g(A)$$

where $g(A)$ represents the remaining life of the node. $g(A)$ is normalized to be in the [0,1] interval and as a consequence the corresponding reluctance will grow significantly when node battery lifetime approaches 0.

This is how cost efficiant protocols were born. [SL] proposed localized, position based, cost aware protocol that would select one of its neighors that minimizes the cost function, *c(A)*, for node *A*:

$$c(A) = f(A)(1 + s/R)$$

where s is the distance from node A to the destination node and R is the node radius. The produced cost aware paths minimize the sum of $f(A)$ for nodes on the path.

Node *B*, that holds the message, selects one of its neighbors A which minimizes $c(A)$. The algorithm proceeds until the destination is reached, if possible, or until a node selects the neighbor the message came from as its best option to forward message. This is the localized cost aware routing algorithm:

```
Cost-routing(S,D);
  B:=S;
  Repeat
    B:=A;                    (* do not set the first time *)
      Let A be neighbor of B that minimizes
        c(A);
      Send message to A
  until A=D (* destination reached *) or A=B (* delivery failed *);
```

An algorithm that is used as a reference point for cost aware protocols is referred to as SP Cost algorithm. In this case, where information about all network nodes battery levels is known to each node, the optimal cost aware algorithm, that will maximize the number of routing tasks that network can handle, can be calculated by applying Dijkstra single source shortest weighted path algorithm. The reluctance $f(A)$ is used as a weight in a shortest weighted path algorithm.

2.6.4 Power-Cost Aware Algorithm

[SL] proposed to merge existing power and cost considerations into a single routing algorithm. Authors discuss power-cost metrics that is considered when a message is sent from node A to node B and propose several power-cost metrics and associated routing algorithms. These localized algorithms select power-efficient paths among cost optimal ones.

One power-cost metric is based on product of reluctance function for node $A, f(A)$, and the power needed for transmission a message between nodes B and A, with distance r,

$u(r)$, as defined in 2.6.3 and 2.6.2, respectively. Power-cost routing algorithm will try to minimize this product metric:

$$power\cos t(B,A) = f(A) * u(r)$$

Alternative mertic, based on the sum would look like this:

$$power\cos t(B,A) = \beta f(A) + \alpha u(r)$$

where α and β values are to be selected (this was done through experiments).

In addition, so-called SP Power-cost algorithm would find the optimal power-cost by applying a single source shortest weighted path Dijkstra's algorithm (assigned node cost is equal to the edge cost leading to the node). As previously noted, this SP algorithm defines the baseline for comparison of proposed power-cost routing algorithms.

2.6.5 Progress Based Power and Cost Aware Routing Algorithms

[KNS3] proposed progress based localized power and cost aware routing algorithms. In addition to these, [KNS3] authors propose other groups of routing algorithms, the projection power-based, iterative power progress-based and iterative projection power progress ones.

The improvements of so-called progress-based algorithms are done on algorithms from [SL]. These new algorithms choose one of its neighbors that are closer to destination than itself. In addition, a selected neighbor is required to minimize the ratio of power and/or cost to reach that neighbor, and the progress made, measured as the reduction in distance to destination (or projection along the line to destination). The new power and cost localized schemes are shown to be conceptually simpler than algorithms presented in [SL] and have similar or better performance.

Let us assume that the destination node is D. Node S currently holds the message and one of its neighbors is node A. Let $r=|SA|$, $d=|SD|$, and $x=|AD|$.

[KNS3] authors look at measure of the proportional progress and the actual power used to make that portion of the progress. As previously discussed, the power spent on transmission of the message from node S to node A is $(r^{\alpha}+c)$. The portion of the progress made with this power is $(d-x)$. With similar advancing through the network, there would be $d/(d-x)$ steps, and the total power cost would be $(r^{\alpha}+c)*d/(d-x)$. Therefore, this new progress based routing algorithm will select a neighbor that will minimize:

$$(r^{\alpha}+c)/(d-x).$$

This is the neighbor that minimizes the power spent per unit of progress made, in terms of getting closer to the destination.

Power metrics is in a similar way replaced with a cost or power-cost metric in order to define cost or power-cost per unit of progress made. This introduces the Cost Progress and Power-Cost Progress routing algorithms. Cost Progress algorithm will select a neighbor that minimizes: $f(A)/(d-x)$

Consequently, Power-Cost Progress algorithm will select a neighbor that minimizes:

$$f(A)(r^{\alpha}+c)/(d-x)$$

where $f(A)$ is a reluctance function for node A, as discussed in 2.6.5:

$$f(A)=1/g(A)$$

and $r=|SA|$, $d=|SD|$, and $x=|AD|$. Node S currently holds the message, A is the neighbor of S and D is the destination point.

Chapter 3

Depth First Search Routing Algorithm

This chapter presents the first centerpoint of the thesis contribution work, the DFS routing algorithm. The DFS internal design is discussed and outlined with more details. The several proposed improvements are observed and how they affect the DFS routing performance. The direct impact on a way DFS memorize information about past traffic is discussed and analyzed as well.

The DFS routing algorithm is characterized with a single path strategy. It is the first localized position-based routing scheme that guarantees delivery even in a case where destination node location information is not updated, i.e. the sender node knowledge about node D coordinates is not correct. As with other localized, position-based, routing algorithms, a node makes DFS routing decision based only on the location of itself, its neighbors, and the destination.

This research work analyzes wireless networks with nodes that do not move (or have insignificant movement patterns). DFS is suggested as potential localized routing strategy for a routing problem in these types wireless networks, with routing performance that is extremely close to the best possible (shortest path, non-localized routing scheme). Due to its ability to guarantee delivery, DFS routing strategy will be interesting to applications that are sensitive to the loss of the traffic.

A significant segment of these applications would be built on top of wireless sensor networks that are formed in ad hoc manner, deployed quickly and quite often requested to host certain intelligence and process exchanged data in a real time. They can be used for military purposes (enemy movements), environmental (monitoring of forest, oceans, airplanes), for the purposes of biological and chemical wars (detection and monitoring), security (schools, residential and business buildings), etc.

3.1 DFS Routing Algorithm

Depth first search, DFS, based algorithm is similar to the routing strategy that has been independently discussed in [JPS]. However, DFS algorithm does not use routing tables like the one proposed in [JPS]. In DFS routing approach the message follows the whole depth first search routing process from a node C to a node D. As explained in 2.3.2 GRA – DFS Alike Routing Algorithm, the node C would be the one where the message stopped, since it is closer to the destination D than any of its neighbors.

A node C that holds a message selects one of its neighbors, a node S that minimizes:

$$d(S,D)$$

where $d(x,y)$ is Euclidean distance between nodes x and y.

This DFS algorithm routes messages in a greedy approach whenever it is possible, i.e. a node exists that is closer to D than S. For dense graphs most of the paths generated by this method are the same as the paths obtained by the greedy method. When the routing process reaches the node C, the routing switches from 'greedy' to 'recovery' mode. In 'recovery' mode, the DFS algorithm will help the routing process by getting back one or more steps away from the node C. After each recovery step, the DFS will try to switch to the 'greedy' mode again, if it is possible, i.e. there is a neighbor that is closer to the destination node D than the node that holds the message. Otherwise, the DFS recovery mode forces routing to return message further back along the route the message previously crossed.

This routing algorithm guarantees delivery for a connected wireless ad hoc network even in a case when the destination location is not correct. The recovery phase of the routing algorithm makes sure that delivery happens all the time. The DFS-based routing algorithm performs DFS search in a distributed way. It builds a path in the graph without relying on nodes that are not direct neighbors of a node that is executing the DFS processing.

In DFS-based algorithm, nodes are marked as white or gray. This marking is specific for the message that is going to be routed throughout the ad hoc network. Initially all nodes are white. The process of visiting nodes coincides with sending messages between nodes. There is always only one copy of the message in the graph.

Figure 3-1 defines a possible routing problem, where *S* is the source node that holds the message and *D* is the destination node. The start node *S* begins routing by changing its status to 'gray'. This means that a node has been visited by this message (that is, it received the message at least once). Each message that is sent from a node *B* to a node *A* has one bit that indicates whether the message is forwarded or returned. Node *A* receiving the message then acts according to that bit.

White node *A*, upon receiving forwarded message for the first time, changes its color to gray. Node neighbors are sorted according to distance to the destination. Neighbors which are closer to destination are preferred and are selected the first. The only exception is that node *B*, that sent message to *A*, is ignored by the algorithm that is executing at node *A*.

Node A is required to memorize, together with the message *id*, the neighbor *B* that forwarded that message. The message is then forwarded to the first choice *C* among node *A* neighbors. If there is no choice, message is returned to *B*.

Gray node *A*, upon receiving forwarded message from any node *B*, will reject the message immediately. That is, the message will be immediately returned to *B*. Gray node *A*, upon receiving a returned message from node *C*, will forward the message to the next choice E in its sorted list of neighbors, if such a neighbor exists. If *A* has no more neighbors in its list, message will be returned to the neighbor *B* which sent the message to *A* (and which was memorized for that purpose).

Figure 3-1 gives an example of a DFS routing path: ***SABC***_A_***C***_SCB_***A***_CA_***S***_C_***SEFGHID***, where bold and regular letters correspond to forwarded and returned/rejected messages, respectively (sender node is marked).

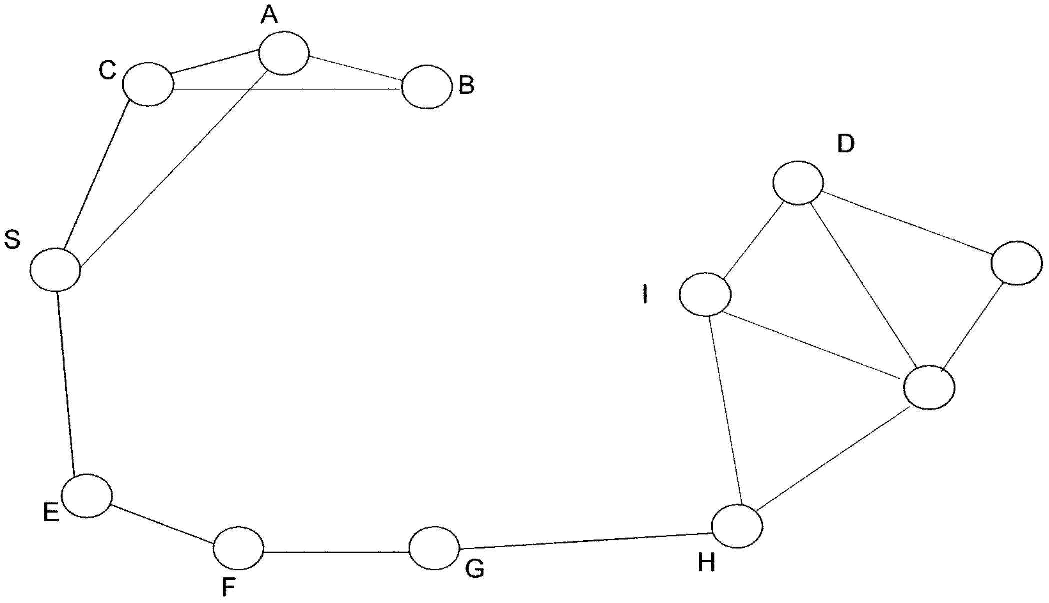


Figure 3-1 ***DFS routing path SABCACSCBACASCSEFGHID***

The pseudo code of *DFS* based routing algorithm:

```
//A is the neighbor of B that is closest to D
Procedure DFS-forward(B, A, D);
```

```
  B forwards message to A;
  If A=D
  then
    message delivered
  else
  {
    If A is white node
    then
    {
      A is colored as gray node;
      A memorizes B;
      A sorts all neighbors (except B) according to distance from D as C1, C2, ..., Ck;
      If k>0
      then
        //C1 is the neighbor of A that is closest to D
        DFS-forward(A, C1, D)
      else DFS-return(A, B, D)
    }
    else DFS-return(A,B,D);
  }

Procedure DFS-return(C,A,D);
  C returns message to A;
  A sorts all neighbors (except memorized neighbor B) according to distance from D as C1, C2, ..., Ck;
  Let L be the index such that C=CL;
```

```
If L < k
then
  DFS-forward(A, CL+1, D)
else
{
  If A=S
  then
    D not connected to S
  else
    DFS-return(A,B,D);}

Program DFS-routing(S,D);
  Let C be the neighbor of S which is closest to D;
  DFS-forward(S, C, D);
```

3.2 Proposed DFS Improvements

One of the focuses of the research is the improvement of the basic DFS routing algorithm. The DFS scheme can be improved in various ways – this section will present several improvement proposals. In this section, we only present the essence of these improvements. Later, in Chapter 5 results obtained from experiments will be used to demonstrate improvements and the performance of various proposed changes and design ideas.

The first one is based on concept of dominating sets and internal nodes. Three types of internal nodes are used, intermediate, intergateway and gateway. The goal here, as described in 2.4 Dominating Sets and Internal Nodes, is to reduce the number of nodes that are used for routing messages throughout the network.

Another proposal for a change and enhancement of DFS routing protocol is the request for memorizing some of the past traffic at network nodes – actually, nodes will memorize traffic of their neighbors. It has been noticed that the basic DFS algorithm considers all neighbors at each node. DFS routing in Figure 3-1 shows that a message from node *C* will be send to node *A* even though the same message was previously sent to node *A,* by some other network node. Apparently, previous attempts to reach the destination node through node *A* have not succeeded and node *C* should exclude node *A* from its list of DFS eligible neighbors (only for this message). As per the DFS algorithm, node *A* was marked on the initial visit of the message as the 'gray' node (for this specific message). As a consequence, when node *C* forwards the message to node *A*, it is immediately rejected by node *A*. DFS hop count includes both of these transmissions.

The proposed modification is to eliminate such communication based on additional memory requirement imposed on network nodes. Nodes will also memorize, in addition to one neighbor (the one message came from for the first time as required by *DFS* algorithm), any neighboring node that sends the message to another node, which is then consequently rejected.

Network nodes need to memorize and manage this data for each individual message that travels throughout the network. DFS routing bases its operation on this knowledge and information that is stored in the required memory structures. These are discussed in more details in 3.3 DFS Memory Requirements. It has to be noted that a DFS-enabled network is capable of routing several messages in an independent manner and in parallel. The unique message information that enables parallel DFS routing is the message id number.

The improved version of the DFS routing algorithm, when executed on a section of the network graph presented on Figure 3-1 will immediately demonstrate the strength and quality of the modifications. Let us assume that presented network nodes are internal nodes of (calculated) dominating sets. They are either intermediate, intergateway or gateway nodes. This is just to highlight that each node on Figure 3-1 is considered for DFS routing. The improved version of DFS applied in this section of the network would generate the following route:

SABCBASCSEFGHID

which is significant improvement, in terms of hop count, over the initial DFS algorithm proposed path:

SABCACSCBACASCSEFGHID

S-A: Node *S* is the start point for the presented routing problem, Figure 3-1. White node *S* changes its state to gray, meaning that the current message has visited the node and has been processed by DFS routing logic. As per DFS, all neighbors of *S* are sorted according to their distance to the destination point *D*. Node *A* is selected as next hop neighbor due to the fact that is the closest to the *D*. The proposed improvement of DFS requires that all neighbors of *S*, in this case nodes *E* and *C*, update their *ignoreNeighbor* list with the recipient of the message, node *A*.

A-B: In a similar way, node *A*, upon receiving the message (this is the first visit of this message), changes its color to gray, sorts all neighbors according to their distance do node *D*. Node *B* gets selected as a next hop selection. Similarly, the proposed improvement of DFS requires that all neighbors of *A*, in this case nodes *C* and *S*, update their *ignoreNeighbor* list with the recipient of the message, node *B*.

B-C: Node *B* DFS localized routing algorithm, for the next hop neighbor is not going to select node *A*, even though this node *B* neighbor is the closest to the destination *D*. As per DFS, the node *B*, that sent the message to node *A*, is ignored by DFS algorithm. Therefore, the next neighbor of node *B*, whose location is the closest to node *D* is node *C*. As required, *ignoreNeighbor* list for node *B*'s neighbors, node *A*, is updated with the recipient of the message, node *C*.

C-B: Node *C* DFS localized routing algorithm, sorts all its neighbors according to their distance to the destination. Node *B* is rejected, since this node sent the message to node *C*. Node *A* is also rejected due to the fact that it is within node *B* *ignoreNeighbor* list. Node *C* rejects the source of the message as well. Since no other neighbors are present, node *C*, as per DFS algorithm, will switch to the recovery mode, i.e. the message will be returned to the node that the initial sending (toward the node *C*). Therefore, it is returned to node *B*.

B-A: Node *B* receives this return message and looks for next DFS eligible neighbor. Node *B* in its internal memory structure keeps information for this message about its neighbor that initially forwarded that message. Therefore, node *A* is again rejected as possible next hop neighbor. Since node *B* has no more neighbors, except *A* and *C*, node *B* decides to return the message to the neighbor that initially forwarded the message to the node *B*. Node *A* is selected, by DFS routing algorithm, as next hop neighbor.

A-S: Similarly, node *A* looks at this return message and searches for the next DFS eligible neighbor. Node *B* was already attempted by DFS. Node *C* is within *ignoreNeighbor* list, therefore, it is rejected as valid choice as well. Since no other neighbors exist, node *A* returns the message to the node that initially sent it; it goes back to node *S*.

S-C: Node *S* searches for the next DFS eligible neighbor. Node *C* is the next one closest to the destination and messages is forwarded to node *C*. Other node *S* neighbors, *A* and *E*, update their *ignoreNeighbor* list with this information (i.e. node *C* is added into their *ignoreNeighbor* lists).

C-S: Node *C* searches through its list of neighbors. It concludes both of its neighbors, nodes A and B, are inside its *ignoreNeighbor* list (see S-A and A-B DFS steps and how the node *C* *ignoreNeighbor* list was updated). Since no neighbor is available, node *C* will, as per DFS, to return the message to node *S*.

S-E: Node *S* keeps running in DFS greedy mode and selects the next DFS eligible neighbor, node *E*. Nodes *C* and *A* update their *ignoreNeighbor* lists with this information (i.e. node *E* is added to their *ignoreNeighbor* lists).

E-F: Node *E* executes DFS routing logic, which runs in greedy mode and selects node's neighbor that is closest to the destination, node *F*, as its next hop selection. Node *E* neighbors, the only node *S*, updates its *ignoreNeighbor* lists as well.

The following steps for this execution are clear and they include F-G, G-H, H-I, and I-D.

3.3 DFS Memory Requirements

As explained in 3.1, DFS Routing Algorithm, DFS is not a stateless routing scheme that does not require memorization about past traffic through the network. Each node has to comply with several requests with respect to remembering details about past traffic.

One potential solution to the memory problem could be to design and maintain (as per DFS routing), at each network node, a memory structure, implemented as a table. Keys in the table would be unique message identification numbers. Corresponding values would be a special data structure that holds all necessary data required by DFS algorithm.

The initial version of the DFS algorithm required two following information to be remembered, per each message:

- The node status – possible values include white or gray (as defined per DFS). The localized DFS algorithm for node *A* reads the corresponding value of each received message from its memory, the DFS memory table. This value determines the DFS algorithm logic, as explained above, in Chapter 3, Depth First Search Routing Algorithm.

  If the message is received for the first time, i.e. there is no entry in the memory table, the status of node *A*, for that specific message, is considered to be 'white'. DFS requires that the status of node *A*, for this message, is changed to 'grey'. The 'gray' status points out that this specific message has previously visited the node *A*.

  If the message was formerly received by node *A*, the node *A* status, for the message is already set to gray. DFS routing algorithm will process this message, as explained above, in Chapter 3, Depth First Search Routing Algorithm. The status of the node *A*, for the message, stays unchanged – 'grey'.

- The initiating node – when node *A* receives the forwarded message for the first time, it is going to remember the node that did the initial transmission of this message (toward node *A*).

The improved version of the DFS algorithm, as presented within our research, establishes additional memory requirement:

- *The node's ignoreNeighbor list* – as presented in 3.2, Proposed DFS Improvements, nodes will need to update their ignoreNeighbor list. A node *C* is required to update its ignoreNeighbor list whenever the message is sent and heard by the node (and

> node *C* is not intended recipient). Node *C* will put the intended recipient in this 'ignore' list. As discussed, the improvement of DFS routing algorithm requests that this list is consulted prior to sending a node. Node *C* neighbors, present in this list, will be excluded as potential DFS candidates for the next forward. The processing will chose the next neighbor, if available, or if no other neighbors are present, the message is returned to the node that sent the message to node *C*.

It should be noted that DFS memory requirements are not connected with network size, i.e. the number of nodes that form the network. DFS is fully scalable with respect to this argument (the network size). As with other localized routing algorithms, a node makes a DFS routing decision based only on the location of itself, its neighbors, and the destination. The network size does not influence this process.

The only information about the traffic that node *A* needs to maintain is strictly tied to messages that travel through the network and get routed through the node. As explained above, everything is associated with the message. The maintenance of a node memory structure is not considered complex.

After a timeout, defined and set on a network level, each node can remove old data structures. These unused pieces of information are tied to message identification numbers that have not been accessed or used for some time. This memory cleanup process can be done independently in the background and in a localized manner, Node's core node functionality would not need to be disturbed (receiving, processing and sending message) in any way. Other network nodes or neighbors do not need to be part of this memory cleanup procedure.

In addition, the total memory usage, at a network node, is not going to be significant. Primarily due to the fact that nodes need only to memorize message and node id numbers that are inherently small. It is going to have a linear relation with an amount of the traffic that travels throughout the node due to the network ability to process several messages in parallel. A node's assigned memory will dynamically grow, as new messages are received and processed by the node, and diminish whenever the memory cleanup procedure is executed.

3.4 DFS Based QoS Routing

The framework of the DFS routing protocol can be extended in order to support QoS routing requirements in multihop ad hoc wireless networks. This is the first proposed localized QoS routing algorithm for wireless ad hoc networks. In addition, this is the first routing strategy that tried to address QoS routing problem by using GPS information in its routing decisions.

The proposed DFS QoS routing informs the source of the bandwidth and quality of service available for required source-destination path. The knowledge obtained by DFS QoS routing is the basis for establishment of connections that meet required QoS in wireless ad hoc networks. This, consequently, enables a whole set of real-time and multimedia applications. An example of these would include a packet radio network, streaming multimedia (voice or data), IP telephone, and others.

Extended DFS QoS routing is modeled after the bottleneck bandwidth model [WL], as previously outlined in 2.5 QoS Routing in Wireless Networks. Propagation delay, as discussed in [WL], is considered as independent delay requirement. In wireless networks, this delay is directly proportional to a hop count between two nodes. The basic strategy with DFS QoS routing is to improve hop count performance as much as possible in order to decrease this delay and increase the associated bandwidth of the network, for a specific source-destination routing path.

DFS-based QoS routing is implemented into three distinctive phases. In the research and analysis of QoS problem in wireless ad hoc networks, the mobility of nodes is not taken into account.

a) Destination search;

A source node, that holds a message, initiates a search for a destination node. It creates a DFS QoS message and sends this message toward the destination node.

This phase depends on destination search approach, which is not clearly defined by DFS QoS. A possible candidate for this inquiry phase can be one of the DFS routing algorithms as well, since it guarantees destination discovery in connected wireless networks. Quite important cases where the destination location is not fully corrected are also resolved by guaranteed delivery of DFS framework.

b) Building of QoS path;

The destination node, when found by the search process, recognizes the DFS QoS message and learns about location of the source node and associated QoS requirements. It is responsible for a construction of the QoS path, which is built by executing the second phase of DFS QoS routing algorithm.

The destination node returns this special message, marked as DFS QoS, toward the source node. In this phase of the DFS QoS routing scheme, the destination node is the DFS start node and the source node is considered as the DFS stop node.

As per DFS QoS, each network node is required to remove nodes that return or reject the message from the QoS path. As the second phase of DFS QoS routing is executed, the DFS QoS path gets constructed.

An example of constructed a QoS path, after a search for DFS QoS path has been executed on the section of the wireless network, is presented in Figure 3-1 *DFS routing path SABCACSCBACASCSEFGHID* would become:

SEFGHID

This path is constructed upon applying the bottleneck bandwidth model, as explained in [WL], on the initially constructed DFS routing path:

SABCACSCBACASCSEFGHID

c) Source initiates QoS routing;

In this final phase of DFS QoS routing, the source node, upon receiving a specially marked DFS QoS message, sent by the destination node, will learn the DFS QoS path toward the destination. The unique message identification number is used to associate the received DFS QoS message with the initial message that needs to be sent along QoS path.

The initial message is now ready to be sent toward the destination node, along the created QoS path. In this phase, network nodes will recognize the message that travels within the network has a predefined DFS QoS path. Each hop of the message will follow the route stored within the message that travels.

Chapter 4

DFS and Power Aware Routing Algorithms

This chapter focuses on the introduction of a new set of power and cost-aware routing algorithms. A progress-based set of power-aware algorithms are analyzed in more detail in 2.6.5 Progress Based Power and Cost Aware Routing Algorithms. They are the basis for the second thesis contribution, the proposal of a new set of power-aware algorithms. The section presents main features and characteristics of these newly proposed algorithms, the way they operate and execute.

The last section outlines assumptions and limitations that have been adopted within the research. Their analysis and the impact on the conducted experiments are discussed as well.

4.1 Power Progress DFS Routing Algorithms

These new algorithms are generated by integrating the improved version of the DFS algorithm (presented in section 3.2) with progress-based power-aware algorithms (presented in section 2.6.5). The essence of the intent is to combine latest versions of power and cost aware algorithms, presented in 2.6.5, Progress Based Power and Cost Aware Routing Algorithms, with the improved version of the DFS algorithm, discussed in 3.2, Proposed DFS Improvements. The improved version of DFS scheme includes both modifications to DFS, i.e. one that uses dominating set principles and a modification that requires nodes to memorize past traffic for messages.

The goal is to extend three power-aware routing algorithms that do not guaranty the delivery of the message with the DFS model. The message delivery, in this proposal, would be guaranteed by the depth first search (DFS) algorithm.

The DFS routing algorithm sorts all neighbors (edges) at each visited node according to the one of progress-based power, cost, or power-cost criteria, as discussed in 2.6.5, Progress Based Power and Cost Aware Routing Algorithms. The framework of the execution for progress-based power and cost-aware algorithms are defined by the DFS routing protocol rules and the DFS routing framework.

This group of routing algorithms is referred to as Power Progress DFS routing algorithms. As previously discussed in 3.1 and 3.2, the greedy mode of DFS will enable this set of routing algorithm to minimize the distance toward the destination D as long as there is a node closer to D than node S that currently holds a message. When such a neighbor is non-existent, DFS recovery mode kicks in, returns the message and tries to switch to greedy mode, if possible. If it is not possible, the recovery mode will continue to search for a neighbor that will enable the switch to greedy mode.

Apparently, the DFS framework is the delivery vehicle in this combined mode, providing that routing never halts and securing that the destination node is eventually reached – regardless of the cost (hop counts or the energy spent along the route search phase). The generated path, i.e. neighbors selected by routing logic, will either create routes that minimize the power per message route or routes that will maximize the number of routing that network can perform.

4.2 A Routing Example

The previously discussed routing problem, as depicted on Figure 3-1, based on the proposed localized power progress-aware algorithm, DFS Power Progress, would potentially create the following route solution:

SCBABCSCSEFGHID

which is different than the routing solution, described in section 3.2:

SABCBASCSEFGHID

S-C: Node *S* is again the start point for the routing problem, Figure 3-1. As discussed in section 3.2, all neighbors are selected according to a certain criteria. In this case, where routing is controlled with a new DFS-based power-aware logic, neighbors are sorted according to associated power-aware values, calculated by following equation:

$$(r^{\alpha} + c)/(d - x)$$

as proposed in section 2.6.5. Let's assume that Figure 3-1 nodes are positioned in such way, so that node *C* minimizes the above calculated value (amongst all other node *S* neighbors). DFS Power Progress algorithm selects node *C* as its first choice. All other DFS routing framework required steps are executed as well. The node changes its status, from 'white' to 'gray'. When the message is transmitted from *S* to *C*, all node *S* neighbors update their *ignoreNeighbor* lists (in this case, nodes *A* and *E*).

C-B: The DFS Power Progress logic at node *C*, might chose node *B* (apparently, this is dependent on real distances between presented nodes) which minimizes the referenced power progress value. As per DFS, the node changes its status to 'gray' and transmits the message. All other DFS required maintenance steps will be executed as well.

B-A: Node *B*, in the absence of any 'forward' nodes, will select one of its neighbors, nodes *A* or *C*, that minimize the power progress equation. Node *A* might be the node selected by the routing algorithm and the message gets sent to this node.

The following steps for the execution of the proposed DFS-based power-progress algorithm are clear and they include A-B, B-C, C-S, S-E, E-F, F-G, G-H, H-I, and I-D.

The resulting DFS Power Progress path, generated as a possible solution to the routing problem, as presented on Figure 3-1, will not be equal to the path created by DFS routing algorithm (as discussed in section 3.2). However, this DFS-based power-aware localized routing is expected to generate a route whose power consumption will be smaller than the consumption of other DFS-generated routes.

4.3 Assumptions

The following is the list of the assumptions that our work and conducted adjacent experiments adopted.

- Mobility of network nodes – As discussed in 2.3 Position Based Routing Algorithms, the problem of finding a satisfactory location update scheme has proved to be quite hard to resolve. The focus of this work is on finding an algorithm that satisfies a routing problem in an ad hoc wireless network where nodes do not move or have small and insignificant moving patterns. If mobility is high and changes of neighbor information across several network nodes are significant, the ability of the DFS routing framework to guarantee message delivery might be negatively impacted.

- Activity status of nodes – In order to preserve the energy in the ad hoc wireless networks, a distributed wake-up model should be implemented in order to simulate a real routing problem. When nodes are in the 'sleep' mode, the network topology changes and that influences the routing as well. As discussed in 2.3 Position Based Routing Algorithms, the currently available solution to the power save mode in ad hoc wireless networks do not provide adequate scheme. In our experiments, we assume that network nodes are present and awake all the time. They receive, transmit packets or are in the idle mode (i.e. listening mode). Apparently, this simplification will help us to model the proposed improvements and a new set of routing algorithms. Simplifications are desired as well due to the fact that the goal is to the comparison with the set of power and cost-aware algorithms that also used analogue simplification as well.

- Communication overhead – The need to communicate information with neighboring nodes or nodes within the same group makes the simulation efforts more complex. The goal of any routing algorithm is to minimize the communication overhead, due to the limited battery power, in order to maximize the number of routing tasks, i.e. the lifetime of the network node.

  Node mobility and the activity status of nodes require extensive communication overhead. As outlined above, these two issues are not taken into the account in this research. Therefore, the communication overhead is not a factor in our experiments. In addition, the communication overhead initiated in the startup phase, when network nodes are powered up and learn about their immediate neighbors, is not taken into consideration either.

- Unique message identification numbers – The research and simulation efforts didn't look into the problem of how this unique message identification number is created and how this knowledge is distributed throughout the network. This knowledge is treated as a primary key that localized DFS running algorithms use for processing and DFS routing messages that they receive.

  This unique identification number is the base of the network node ability to route and process several messages in parallel.

- Dominating set status – We consider the established network dominating sets correct and static due to the absence of mobility of nodes while the routing happens.
- Timeout value – As per 3.3 DFS Memory Requirements, various memory structures are created, at the node level, for each routed message. Since the network is capable of routing several messages at the same time, the associated memory structures, at the node level, will grow as messages are processed and retransmitted. DFS relies on a timeout mechanism that is executed in a localized manner, at each network node, which is needed to prevent the exhaustion of the memory resources. This research and associated simulation effort did not analyze and monitor the memory consumption level at network nodes, as the DFS algorithm executes. Since the required memory structures are small and the size of the modern memory components have significantly increased, a simple cleanup procedure, executed in a localized manner, with a constant, globally set timeout values, will prevent any problems being created around this issue.
- Security – This is an interesting topic for wireless ad hoc networks. As the traffic starts to flow between network nodes in a seemingly uncontrolled manner, the need

for security and authentication arises. This thesis research did not look into these issues nor have they been implemented in the simulation experiments.

Two key simplifications, the mobility and activity status of network nodes, do not reduce the significance of proposed power-aware DFS-based routing algorithms. In a first case, the case where the routing is done in a mobile wireless networks, one possible research strategy is to combine DFS routing with independently executed location update scheme. The goal here would be accomplished in two phases. In a first phase, the source node would initiate a search for latest correct destination information using adequate location update scheme. The network response to the search packet would provide correct and valid destination coordinates. The second phase would be based on DFS routing algorithm which would then route the message toward the destination node.

It is to be confirmed how efficient DFS would be when the destination nodes moves and it location information becomes invalid. Small movement patterns or a temporary change of location events would be successfully resolved by DFS framework ability to deliver even when destination location is not correct. Other moving patterns, especially the networks with significant moving events, would need more research and probably a new approach for addressing this complex issue.

Similarly, the ad hoc networks need to preserve the power level of network nodes will impose a certain power management scheme upon all network nodes. Again, as in a previous case, it is fair to assume that certain subset of network nodes is 'awake' and that our routing algorithms execution will be based on present nodes. The DFS routing strengths will enable that a message gets delivered to the destination node as long as the network stays connected and the destination node is ‘awake’. Nodes that are in the sleep mode will not break the guaranteed delivery.

The goal of listed simplifications is to, in addition to the simplification of a complex routing problem, to properly evaluate and understand a true routing performance of the DFS routing algorithms in this specific case (network nodes do not move).

Chapter 5

Performance Evaluation

In this chapter, experimental results in both areas of research are presented, a set of DFS routing algorithms and a new set of power-aware DFS-based routing algorithms. The first section of this chapter explains the simulation environment and its main features.

The second section compares hop count performances of DFS algorithms and how various improvements affect the performance of the initial DFS routing algorithm. Only a subset of experiment results is presented – network sizes with 40, 100, and 250 network nodes with selected network density values (the average number of neighbors). The best improvement scenarios are chosen and explained. The resulting improved DFS routing algorithm, the best DFS candidate, is used as proposed DFS component for a new set of routing algorithms. The third section looks into the performance numbers of DFS QoS routing algorithm.

The fourth and fifth sections monitor the performance of a new set of proposed routing algorithms (DFS power and cost aware schemes). Obtained simulation results demonstrate improvements to the power and cost metrics of DFS scheme. At the same time, they manifest the ability of the DFS framework to guarantee delivery for power progress aware algorithms that do not warrant the delivery all the time.

5.1 Simulation Environment

In order to evaluate the thesis contributions to existing the DFS algorithm and to understand the performance of the newly proposed routing algorithms, extensive simulation experiments were executed within the framework of a specially constructed network simulator. This custom network simulator was designed and implemented in Java programming language and used to construct ad hoc wireless networks and required routing of messages through randomly formed networks.

The source code, as explained in Appendix C, is available for download. It gives one the ability to easily reproduce all presented simulation results. The same simulation platform has been used for all required variations of experiments. Few simulator relevant configuration parameters are required to be adjusted for two separate test scenarios (execution environments):

- environment setup to simulate various DFS improvement schemes and
- environment setup for testing a new set of progress power and cost aware routing algorithms combined with DFS scheme

Ad hoc networks have been modeled by random unit graphs. Experiments have been executed only on connected graphs – disconnected generated graphs were rejected immediately. Each of n network nodes was chosen by selecting its x and y coordinates at random in the interval $[0, 100)$.

In order to control the average node degree k (the average number of neighbors per node), all potential $n*(n-1)/2$ edges in the network were sorted by their length, in increasing order. Selected node radius R that corresponds to chosen value of k was equal to the length of $nk/2$-th edge in the sorted order.

5.1.1 Implementation Details

The simulation was written in Java programming language. Eclipse, version 3.1.0, was used as integrated development environment (IDE). The included code presents the integrated network simulator, extended and modified to support the DFS routing algorithm and various modifications. It is available for download from the following web site:

http://home.mycybernet.net/~bosko/DFS-Power

The zip file has to be extracted to a new folder and the simulator will run from a DOS prompt or from a UNIX command line. Included redme.txt file includes more detailed instructions for simulation running. In addition, various configuration parameters are presented and discussed. They have to be properly set and changed, as different simulation goals are established. In addition, it is possible to run a full suite of experiments with only one command.

The code has a number of test cases that were implemented to support the experiment requirements. The central method for executed experiments is inside GraphP.test(). This class contains the simulation main() routine. The experiment code, written in Java, will run on any other platform that has properly installed and configured JRE (Java Runtime Environment).

5.2 Performance Evaluation of DFS Based Routing Protocols

In order to evaluate the DFS-based routing protocols and proposed improvements, as they were outlined in 3.2, Proposed DFS Improvements, the experiments were structured as follows.

The basic DFS routing algorithm was used to route messages in simulated ad hoc wireless networks. The first set of experiments were designed to test the initial version of DFS algorithm. This basic version of DFS was then improved with the concept of internal nodes.

Three different types of internal node – intermediate, intergateway and gateway, as discussed in section 2.4 and introduced by [SSZ. WL], were used to restructure the network topology. A separate suite of experiments was conducted for each one of them. In addition, the proposed improvement of the DFS algorithm, based on memorizing the past traffic, as outlined in 3.2, was combined with DFS routing over internal nodes (intergateway and gateway nodes).

The properly designed and implemented simulation environment has given us the ability to test all of these cases of incremental improvements separately and with minimal changes to the simulation platform configuration parameters. The effect of randomness on results was eliminated by running the required experiments over 100 times. Each rerun was completely independent from previous executions and happened within a new, randomly created, ad hoc wireless network.

Each experiment execution included following types of simulations:

- DFS (initial):

  In this scenario, all network nodes are considered when the DFS routing algorithm routes a message through the network. Therefore, the number of internal nodes, used for routing, is equal to the number of nodes. The DFS algorithm will use all network nodes for routing messages throughout the network.

- DFS-Int:

The basic DFS routing is improved by introducing intermediate nodes (as introduced in 2.4 and 2.4.1). The network nodes will set their status to intermediate if they satisfy intermediate requirement. The DFS algorithm will use only these intermediate nodes for routing messages throughout the network.

- DFS-IGW:

  Another incremental improvement of the DFS routing scheme is tested with the introduction of intergateway nodes (as defined and introduced in 2.4 and 2.4.1). Nodes that satisfy intergateway requirement will form a new dominating set and the routing will be based only on these nodes. Intermediate nodes are not considered or used.

- DFS-IGW-I:

  DFS routing is additionally modified and adjusted, as discussed in 3.2, Proposed DFS Improvements – based on a proposed request for DFS to memorize past traffic at each network node. In this scenario, the dominating set, formed by intergateway nodes, is not changed. The only modification is done to the DFS routing scheme. The DFS algorithm will, as in DFS-IGW mode, be based only on intergateway nodes.

- DFS-GW:

  In this case, the dominating set of the network is recalculated. Network nodes that satisfy the gateway definition, as defined in 2.4 and 2.4.1, define new internal nodes of the network. This experiment is based on the initial the DFS scheme (no memorizing of the past traffic is done, as proposed by the DFS improvement request). The DFS algorithm will use only these gateway nodes for routing messages throughout the network. Intergateway or intermediate nodes are not used.

- DFS-GW-I:

  The previous DFS routing mode is changed, the DFS scheme is improved to memorize past traffic (as discussed in 3.2). As in DFS-GW case, the routing is based only on gateway nodes. Intergateway or intermediate nodes are not used.

- SP:

  This corresponds to the non-localized shortest path routing algorithm used in this chapter to find the best possible 'hop count' value. In this case, the performance results of various DFS based routing algorithms are compared with Dijkstra shortest path routing algorithm. This non-localized algorithm possesses global knowledge about the network and defines the best possible routing for established routing scenario (routing between nodes of randomly created connected ad hoc wireless networks). Since Dijkstra has a global knowledge, the routing solution is equivalent to the shortest path problem.

Every conducted simulation required that a message is sent from each network node to all other network nodes. Therefore, for each test of proposed routing algorithms, as stated above, the network routed $n*(n-1)$ messages. Experiments were executed on three different network sizes, i.e. with different number of network nodes:

- Table 5-1 presents results for networks with n=40 nodes.
- Table 5-2 presents results for networks with n=100 nodes.
- Table 5-3 presents results for networks with n=250 nodes.

In addition, within the framework of each network size, all of these experiments were rerun on newly randomly generated networks with different average numbers of neighbors per each node (the network density). Each column in listed tables Table 5-1, Table 5-2, and Table 5-3, is another and independent execution of the experiment.

All presented results are average values of 100 executions over randomly created graphs. Unconnected graphs were immediately rejected. Each table cell includes the following four digits:

- Average hop count (includes reject and return messages)
- Average reject count (as per DFS algorithm)
- Average return count (as per DFS algorithm)
- Average number of 'internal' nodes (nodes that formed dominating set of the network)

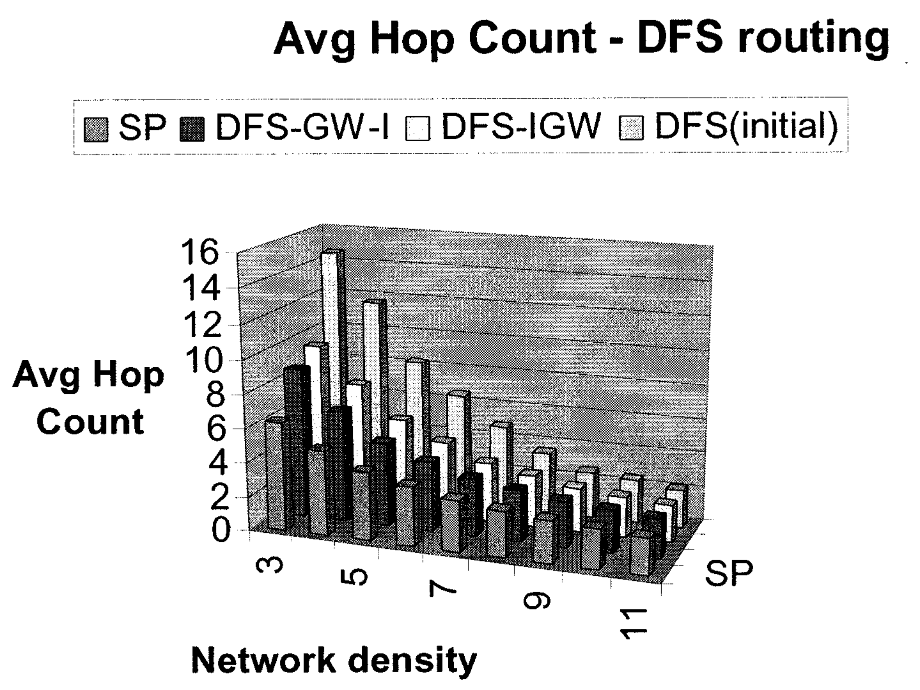
For illustration purposes, the number of messages that had to be routed throughout the network needed for one table data population is:

$$54*100*n*(n-1)$$

where n is number of network nodes, 100 is number of reruns needed to average results of the experiment and 54 is the number of table cell (shortest path information did not need a separate reruns). The population of one table cell from Table 5-3 (test case with n=250 nodes) with simulation data required 6225000 messages to be sent and routed throughout the simulated and randomly formed ad hoc wireless network.

Degree (*d*)	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**labels**
DFS (initial)	14.626	11.789	8.32	6.542	4.879	3.553	2.681	2.505	2.199	Avg. hop
	2.117	2.199	1.512	1.156	0.731	0.323	0.062	0.066	0.005	Rej. count
	1.727	0.839	0.35	0.177	0.096	0.024	0.003	0.0042	~0	Ret. count
	40	40	40	40	40	40	40	40	40	Internal nodes
DFS-Int	11.239	9.55	7.091	5.863	4.413	3.387	2.635	2.448	2.195	Avg. hop
	1.196	1.477	1.081	0.906	0.55	0.262	0.046	0.044	0.004	Rej. count
	1.054	0.524	0.222	0.122	0.065	0.014	0.0007	0.0016	~0	Ret. count
	28	31	32	34	34	35	36	36	37	Internal nodes
DFS-IGW	9.55	7.44	5.523	4.448	3.457	3.032	2.561	2.374	2.184	Avg. hop
	0.543	0.635	0.435	0.31	0.143	0.108	0.017	0.015	0.001	Rej. count
	0.925	0.414	0.167	0.085	0.035	0.012	0.0012	0.001	~0	Ret. count
	25	24	24	24	24	24	24	24	23	Internal nodes
DFS-IGW-I	8.922	6.715	5.029	4.083	3.288	2.924	2.548	2.359	2.183	Avg. hop
	0.229	0.273	0.188	0.128	0.059	0.053	0.010	0.007	0.0009	Rej. count
	0.925	0.414	0.167	0.085	0.035	0.012	0.0012	0.001	~0	Ret. count
	25	24	24	24	24	24	24	24	23	Internal nodes
DFS-GW	9.17	6.905	5.16	4.215	3.470	3.044	2.693	2.482	2.325	Avg. hop
	0.359	0.351	0.214	0.138	0.069	0.045	0.008	0.006	0.004	Rej. count
	0.914	0.42	0.177	0.094	0.046	0.02	0.0065	0.0044	0.0019	Ret. count
	25	22	21	19	18	17	15	15	14	Internal nodes
DFS-GW-I	8.755	6.513	4.919	4.064	3.389	3	2.684	2.476	2.322	Avg. hop
	0.151	0.155	0.094	0.063	0.029	0.023	0.004	0.003	0.002	Rej. count
	0.914	0.42	0.177	0.094	0.046	0.02	0.0065	0.0044	0.0019	Ret. count
	25	22	21	19	18	17	15	15	14	Internal nodes
SP	6.33	4.955	4.012	3.445	2.994	2.708	2.488	2.318	2.171	

Table 5-1 Average hop counts (include reject and return messages) for DFS - n=40 nodes

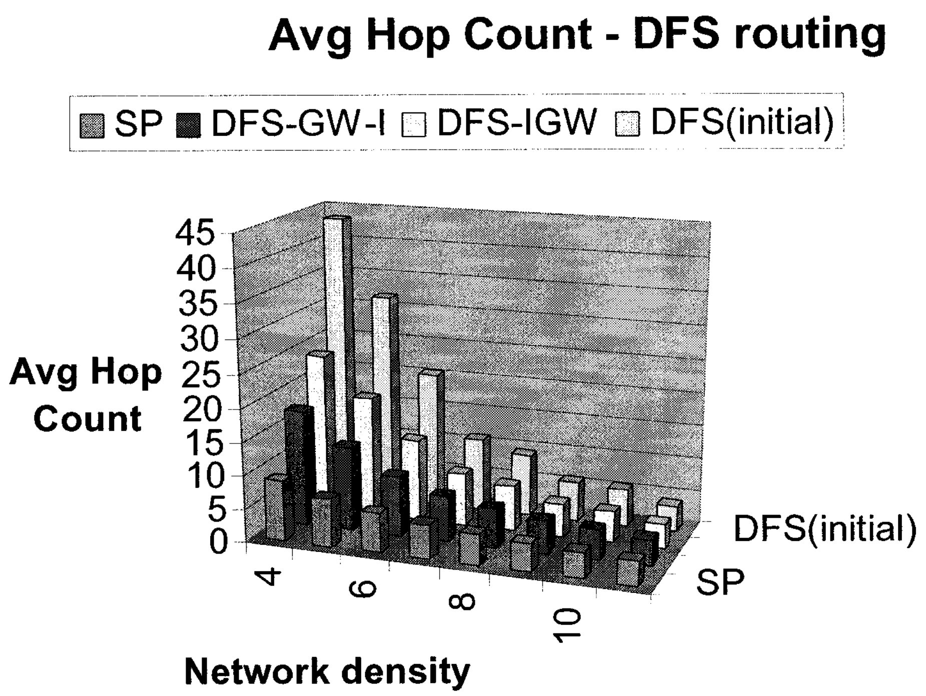


	3	4	5	6	7	8	9	10	11
▨ SP	6.3	5	4	3.4	3	2.7	2.5	2.3	2.2
■ DFS-GW-I	8.8	6.5	4.9	4.1	3.4	3	2.7	2.5	2.3
□ DFS-IGW	9.6	7.4	5.5	4.4	3.5	3	2.6	2.4	2.2
▨ DFS(initial)	15	12	8.3	6.5	4.9	3.6	2.7	2.5	2.2

Figure 5-1 ***Graphical representation of data from Table 5-1***

Degree (*d*)	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**labels**
DFS (initial)	Connected graphs could not be generated	43.421	31.939	20.489	11.118	9.348	5.917	5.642	3.939	Avg. hop
		11.16	8.896	5.507	2.245	1.847	0.648	0.708	0.11	Rej. count
		4.647	2.193	0.877	0.247	0.153	0.036	0.045	0.003	Ret. count
		100	100	100	100	100	100	100	100	Internal nodes
DFS-Int		34.151	26.929	18.013	10.242	8.544	5.696	5.407	3.856	Avg. hop
		8.118	7.142	4.6	1.935	1.535	0.571	0.615	0.078	Rej. count
		3.285	1.611	0.653	0.178	0.109	0.021	0.036	0.0006	Ret. count
		79	84	87	90	91	93	94	94	Internal nodes
DFS-IGW		23.979	18.184	12.329	8.008	6.895	5	4.737	3.721	Avg. hop
		3.915	3.406	2.134	0.973	0.82	0.272	0.321	0.025	Rej. count
		2.723	1.306	0.503	0.155	0.091	0.016	0.033	0.0008	Ret. count
		65	66	66	68	68	69	70	70	Internal nodes
DFS-IGW-I		19.168	13.986	9.729	6.825	5.883	4.689	4.317	3.695	Avg. hop
		1.509	1.307	0.834	0.382	0.314	0.117	0.111	0.012	Rej. count
		2.723	1.306	0.503	0.155	0.091	0.016	0.033	0.0008	Ret. count
		65	66	66	68	68	69	70	70	Internal nodes
DFS-GW		20.127	14.852	10.358	7.539	6.374	5.151	4.619	4.041	Avg. hop
		2.178	1.805	1.085	0.561	0.423	0.194	0.132	0.049	Rej. count
		2.593	1.312	0.58	0.238	0.146	0.051	0.038	0.01	Ret. count
		60	58	54	52	50	47	45	43	Internal nodes
DFS-GW-I		17.532	12.706	9.065	6.866	5.866	4.928	4.469	3.986	Avg. hop
		0.88	0.733	0.439	0.225	0.169	0.083	0.057	0.021	Rej. count
		2.593	1.312	0.58	0.238	0.146	0.051	0.038	0.01	Ret. count
		60	58	54	52	50	47	45	43	Internal nodes
SP		9.152	7.273	5.954	5.14	4.626	4.178	3.863	3.593	

Table 5-2 Average hop counts (include reject and return messages) for DFS - n=100 nodes

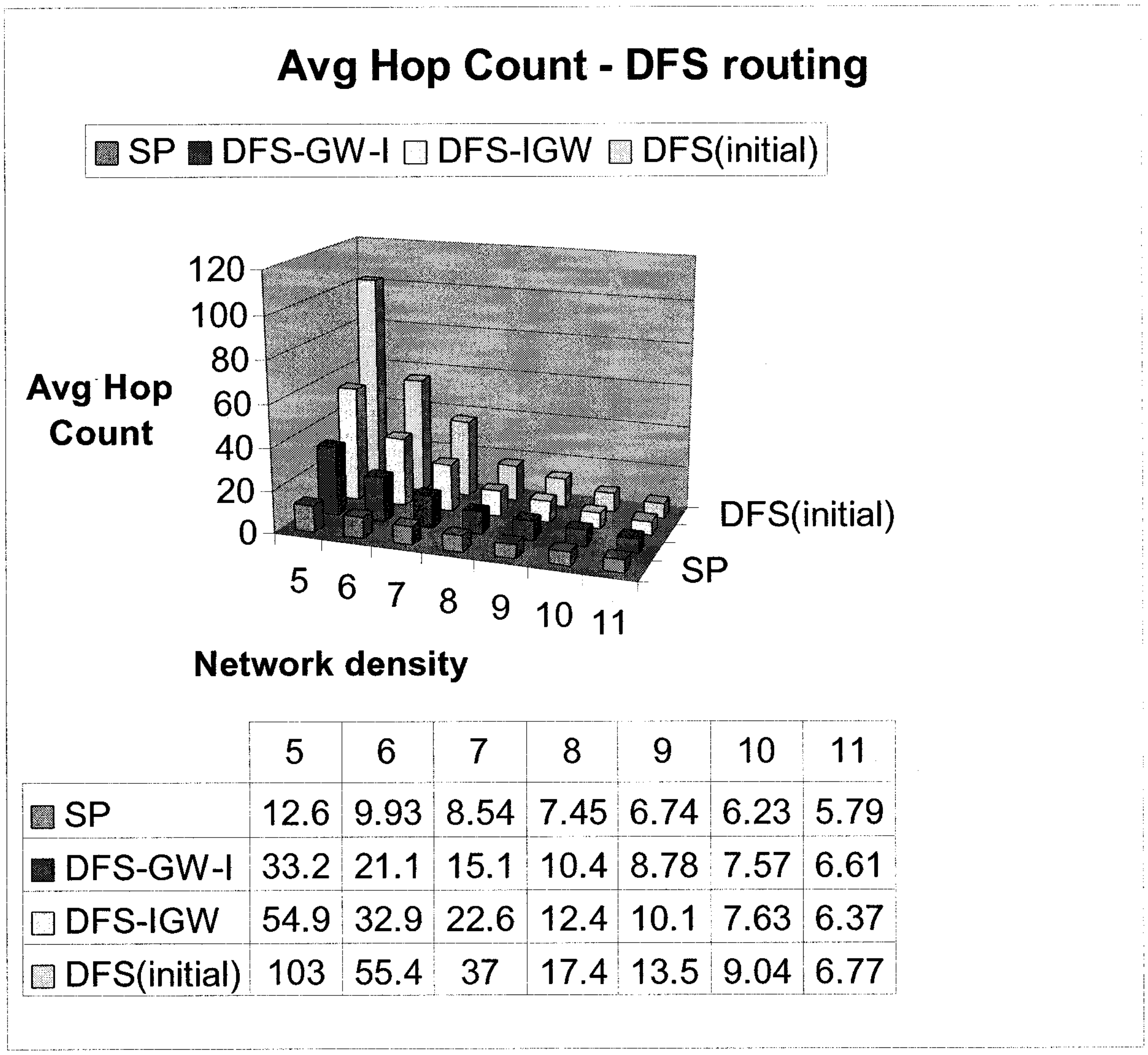


	4	5	6	7	8	9	10	11
▨ SP	9.15	7.27	5.95	5.14	4.63	4.18	3.86	3.59
■ DFS-GW-I	17.5	12.7	9.07	6.87	5.87	4.93	4.47	3.99
□ DFS-IGW	24	18.2	12.3	8.01	6.9	5	4.74	3.72
▨ DFS(initial)	43.4	31.9	20.5	11.1	9.35	5.92	5.64	3.94

Figure 5-2 ***Graphical representation of data from Table 5-2***

Degree (*d*)	3	4	5	6	7	8	9	10	11	labels
DFS (initial)	Connected graphs could not be generated		102.61	55.412	37.012	17.428	13.456	9.044	6.766	Avg. hop
			32.695	17.228	11.215	3.849	2.657	1.053	0.313	Rej. count
			8.764	2.939	1.294	0.262	0.127	0.038	0.004	Ret. count
			250	250	250	250	250	250	250	Internal nodes
DFS-Int			86.438	48.692	33.499	16.317	12.817	8.783	6.662	Avg. hop
			27.017	14.781	9.897	3.436	2.412	0.956	0.276	Rej. count
			6.762	2.298	1.007	0.188	0.098	0.028	0.002	Ret. count
			214	222	230	235	238	240	242	Internal nodes
DFS-IGW			54.946	32.889	22.563	12.384	10.105	7.632	6.367	Avg. hop
			13.436	7.841	5.051	1.717	1.215	0.463	0.156	Rej. count
			5.467	1.935	0.79	0.145	0.078	0.021	0.002	Ret. count
			169	173	177	181	184	187	189	Internal nodes
DFS-IGW-I			37.745	22.869	16.11	10.278	8.585	7.069	6.189	Avg. hop
			4.836	2.832	1.825	0.664	0.455	0.181	0.066	Rej. count
			5.467	1.935	0.79	0.145	0.078	0.021	0.002	Ret. count
			169	173	177	181	184	187	189	Internal nodes
DFS-GW			42.266	26.613	18.441	11.641	9.636	7.986	6.753	Avg. hop
			7.352	4.439	2.696	1.031	0.711	0.36	0.126	Rej. count
			5.524	2.308	1.057	0.29311	0.154	0.062	0.018	Ret. count
			147	142	137	33	129	123	120	Internal nodes
DFS-GW-I			33.168	21.147	15.119	10.417	8.784	7.566	6.608	Avg. hop
			2.803	1.706	1.035	0.419	0.285	0.15	0.053	Rej. count
			5.524	2.308	1.057	0.293	0.154	0.062	0.018	Ret. count
			147	142	137	133	129	123	120	Internal nodes
SP			12.57	9.927	8.541	7.447	6.736	6.227	5.79	

Table 5-3 Average hop counts (include reject and return messages) for DFS - n=250 nodes



	5	6	7	8	9	10	11
▨ SP	12.6	9.93	8.54	7.45	6.74	6.23	5.79
■ DFS-GW-I	33.2	21.1	15.1	10.4	8.78	7.57	6.61
□ DFS-IGW	54.9	32.9	22.6	12.4	10.1	7.63	6.37
░ DFS(initial)	103	55.4	37	17.4	13.5	9.04	6.77

Figure 5-3 ***Graphical representation of data from Table 5-3***

The available simulation results, as presented in Table 5-1, Table 5-2, and Table 5-3, offer lots of data and input for analysis. The following is the list of several key messages coming out from these three tables:

- It is interesting to monitor how the size of dominating set affects the hop count of the DFS routing algorithms. For example, Table 5-1 includes *n*=40 and *degree*=5 network configuration. The size of dominating set (the number of internal nodes used for routing of the message), for this network (40/5) changes from 40 (initial) to 32 (intermediate nodes), then to 24 (intergateway nodes) and finally to 21 (gateway nodes) internal nodes. Even though the number of internal nodes kept decreasing, the hop count was not punished, i.e. it kept improving as well.

  It would be normal to expect the number of hop counts to increase, as the number of internal nodes decreases (intergateway vs. gateway nodes). This would be based on the fact that network nodes have fewer 'core nodes' available for their routing requests. However, this is not the case all of the time. For example, Table 5-1 data shows that for *d*=[3, 6], the hop count decreases even though the number of internal nodes decreases. The turning point, for network size *n*=40 appears to be *d*=7. For this and greater values of network density parameter, the simulation data demonstrate the increase of the hop count as the number of internal nodes decreases (intergateway vs. gateway based dominating sets).

  Other network sizes appear to have a similar turning point. Data from testing network sizes with *n*=100, Table 5-2, confirm *d*=9, while networks with *n*=250, Table 5-3, show that value to be *d*=10. In this last case, *n*=250, if intergateway nodes with DFS improvement is compared with gateway nodes with DFS improvement, the turning point for this value appears to be *d*=8.

- The size of connected dominating sets (either based on intermediate, intergateway or gateway nodes) and their dependency on network density parameter can be analyzed. As a general rule, it appears that the number of intermediate and intergateway nodes keeps growing as network density increases. However, at the same time, the number of gateway nodes decreases as the network density increases. A small exception to this rule is the case of intergateway nodes (for n=40 networks) whose numbers mostly stay constant, or slightly decrease, as network density increases. Table rows (with intermediate, intergateway, and gateway nodes) demonstrate these facts.

  Apparently, the number of network internal nodes is inversely proportional with the hop count numbers. This is consistent even for the exception case presented above (intergateway nodes for n=40 networks).

- Improvements to the DFS routing algorithm hop count numbers are evident as different dominating set notions are applied. The lesser network density, the improvements are bigger.

  For example, n=40 and d=3, the basic DFS algorithm is improved ~40% with the best DFS routing scheme (which is, in this scenario, improved version of DFS with gateway nodes). This improvement is only ~1% for n=40 and d=11 and ~6% for n=40 and d=10.

  For a bigger network, these improvement numbers are much higher. Table 5-3 presents the improvement to be ~68% for n=250 and d=5. A bigger density, for the same network size, gives ~9% (n=250 and d=11).

- It is important to note that small improvements are delivered in those network scenarios where the basic DFS routing algorithm performance is quite similar to the best possible routing solutions. For example, the improvement of ~1% (n=40 and d=11) is delivered for basic DFS performance which is only ~1.3% worse from shortest path hop count.

However, the key message, coming out of the available data, is that the improved versions of the DFS algorithm greatly help the hop count performance to converge to the ideal solution. This gap is dramatically narrowed for those network scenarios where basic DFS routing algorithm performance is far from established (shortest path hop count) baseline – networks with small network density values. At the same time, these improved algorithms deliver their benefits even in those network cases where basic DFS performance is quite close to the ideal one – networks with higher network density. This performance has been consistent across different network sizes (n=40, n=100, and n=250) and monitored network density values.

- The presented simulation data prove that the second proposed DFS improvement, as discussed in 3.2, benefits analyzed DFS routing schemes. Improvements to hop counts are presented in both used cases, for DFS intergateway routing and for DFS gateway routing. Similarly to improvements to the DFS dominating set, the benefits are more visible in cases where deviations of basic DFS metrics from ideal routing numbers are higher.

  For example, in this case, degree number d=4 and n=40, the first improvement (to the DFS intergateway routing) is ~10% and the second one (to the DFS gateway routing) is ~6%. Or, for n=100 and d=4, the first improvement is ~20% and the second one is ~13%.

  On the other side of included data, these improvements are lower (since the performance numbers are closer to the ideal anyway). For network size n=250 and d=11, the first improvement is ~3% and the second one is ~2%. Evidently, the improvement pattern and the actual contribution follow the pattern of dominating set improvements to DFS.

- An important conclusion, coming from these tables, is also the fact that improved versions of DFS help the routing problem on two issues, independently. The improvement in the hop count metric is delivered jointly with the improvement to the number of internal nodes (this number is decreased). This reduced number of internal nodes will make the introduction and future integration of DFS routing schemes with location update protocols and wake-up power based management schemes much easier.

- Interesting conclusion can be drawn from section 5.2.1. The hop count performance of the best DFS routing candidates is compared with hop count of FACE and GFG routing algorithms (introduced and simulated in another paper, [SD-th]). It appears that DFS routing scheme performance is not as good as the best routing candidate from that group, GFG-IG. However, it is quite close. At the same time, DFS routing offers better performance for networks that are not well connected (*d*=3 and *d*=4).
- The best possible DFS routing candidate appears to be either improved DFS with intergateway internal nodes or improved DFS with gateway internal nodes. The network density value determines the turning point for the selection of the best DFS candidate. For networks *n*=40, turning point appears to be *d*=7. This point for networks *n*=100 is *d*=9 and for networks *n*=250 turning density value is *d*=8.

  This routing scheme has hop count performance very close to the ideal, for networks that are well connected (*d*>10). The best values are gained for smaller networks. For example, the best routing solution differs from shortest path metrics only ~6.5% (*n*=250, *d*=11), ~2.7% (*n*=100, *d*=11) and almost identical to the shortest path baseline, ~0.5% difference for *n*=40 and *d*=11.

5.2.1 FACE and GFG Routing Performance

This section presents more simulation data and enables additional comparison of the hop count performance of the best DFS routing candidate(s) with hop count performances of algorithms discussed in [SD-th]. These [SD-th] algorithms are created in similar fashion to DFS power aware routing algorithms – the approach was to extend the guarantee delivery of FACE based algorithms with power and cost aware routing.

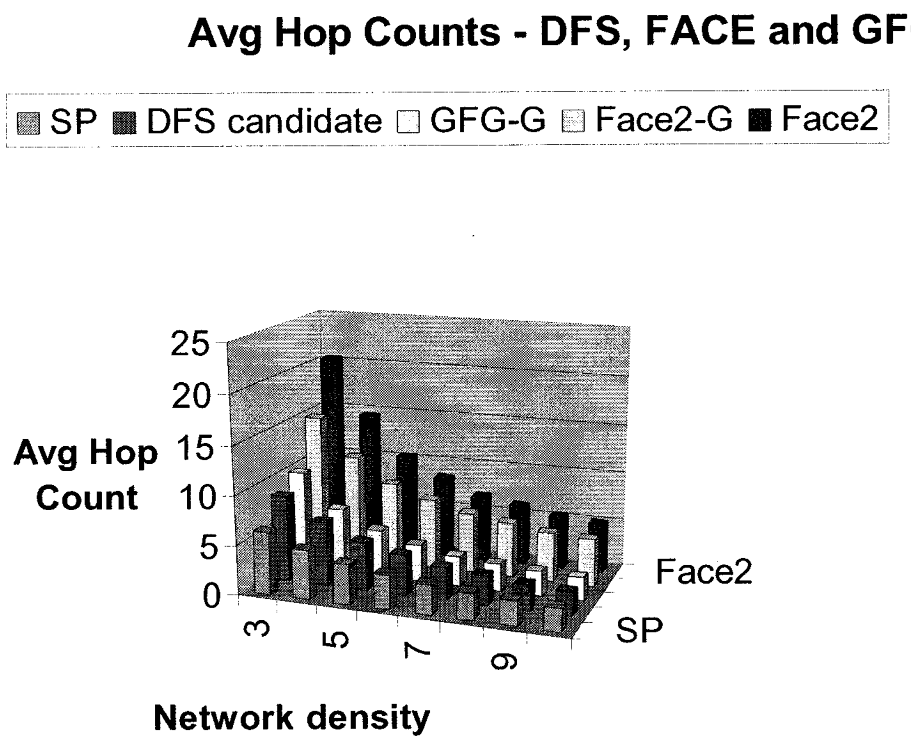
FACE is a localized routing scheme introduced by [BMSU] which guarantees the message delivery. The improved version of FACE, called GFG, is also described in the same paper ([BMSU]). GFG (Greedy-FACE-Greedy) keeps switching its routing mode (between Greedy and FACE modes) and by doing that improves the FACE performance. Suffixes 'IG' and 'G' indicate intergateway and gateway node definitions of dominating sets.

[SD-th] work improved the performance of FACE and GFG algorithms (their hop count) by applying Internal nodes and Shortcuts methods.

Included simulation data for FACE and GFG experiments in Table 5-4 and Table 5-5 are not verified and are assumed to be correct. These two tables include hop count performance of the best DFS routing candidates as well as the SP (shortest path) hop count results – as presented in Table 5-1 and Table 5-2.

Degree (*d*)	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Face2	20.404	14.555	10.391	8.476	6.81	6.236	5.485	5.166
Face2-I	15.612	11.78	9.143	7.885	6.504	5.93	5.304	5.034
Face2-IG	14.915	11.054	8.593	7.329	6.207	5.65	5.072	4.798
Face2-G	14.915	11.008	8.58	7.249	6.206	5.667	5.057	4.853
GFG	12.115	7.924	5.303	3.975	3.1	2.794	2.473	2.303
GFG-I	10.306	6.832	4.917	3.87	3.051	2.748	2.457	2.298
GFG-IG	10.083	6.607	4.8	3.792	3.019	2.72	2.438	2.295
GFG-G	10.083	6.584	4.851	3.779	3.018	2.721	2.44	2.299
DFS-IGW-I or DFS-GW-I	8.755	6.513	4.919	4.064	3.288	2.924	2.548	2.359
SP	6.33	4.955	4.012	3.445	2.994	2.708	2.488	2.318

Table 5-4 Average hop counts – *n*=40 nodes (FACE and GFG variations)

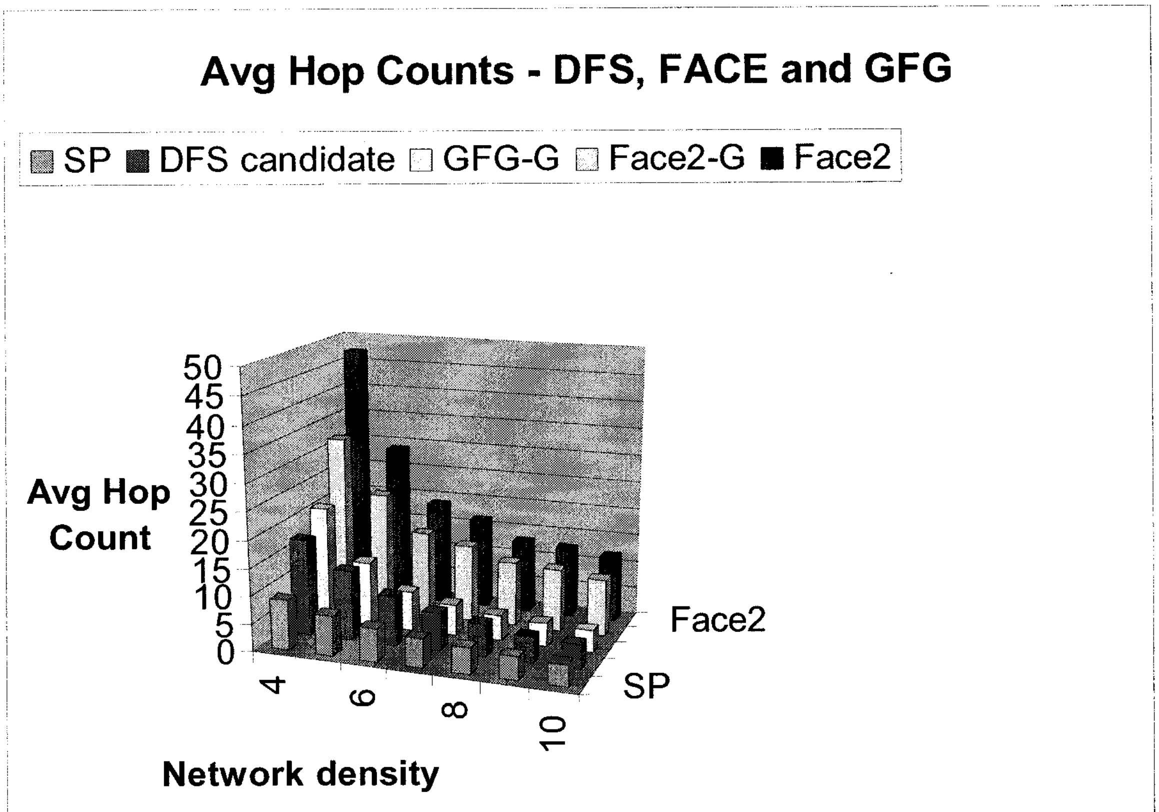


	3	4	5	6	7	8	9	10
▨ SP	6.3	5	4	3.4	3	2.7	2.5	2.3
■ DFS candidate	8.8	6.5	4.9	4.1	3.3	2.9	2.5	2.4
☐ GFG-G	10	6.6	4.9	3.8	3	2.7	2.4	2.3
▨ Face2-G	15	11	8.6	7.2	6.2	5.7	5.1	4.9
■ Face2	20	15	10	8.5	6.8	6.2	5.5	5.2

Figure 5-4 ***Graphical representation of data from Table 5-4***

Degree (*d*)	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Face2	47.353	29.025	19.007	16.717	13.324	12.787	12.037
Face2-I	35.012	24.534	17.185	15.415	12.701	12.299	11.671
Face2-IG	32.175	22.313	15.682	14.123	11.803	11.393	10.477
Face2-G	32.205	22.194	15.759	14.035	11.819	11.307	10.427
GFG	29.263	13.551	7.876	5.976	4.674	4.318	3.857
GFG-I	22.608	12.573	7.526	5.869	4.649	4.229	3.859
GFG-IG	20.994	11.71	7.155	5.679	4.612	4.201	3.837
GFG-G	21.057	11.721	7.18	5.64	4.627	4.206	3.841
DFS-IGW-I or DFS-GW-I	17.532	12.706	9.065	6.866	5.866	4.689	4.317
SP	9.152	7.273	5.954	5.14	4.626	4.178	3.863

Table 5-5 Average hop counts – *n*=100 nodes (FACE and GFG variations)



	4	5	6	7	8	9	10
▨ SP	9.15	7.27	5.95	5.14	4.63	4.18	3.86
■ DFS candidate	17.5	12.7	9.07	6.87	5.87	4.69	4.32
☐ GFG-G	21.1	11.7	7.18	5.64	4.63	4.21	3.84
▨ Face2-G	32.2	22.2	15.8	14	11.8	11.3	10.4
■ Face2	47.4	29	19	16.7	13.3	12.8	12

Figure 5-5 ***Graphical representation of data from Table 5-5***

As per available data (Table 5-4 and Table 5-5), best DFS candidates (DFS-IGW-I and DFS-GW-I), prove to be better routing solutions than the best GFG candidates (GFG-IG and GFG-IG) for $d<5$. For values $d>4$, their hop count performance is not as good as GFG, however, it is quite close. This demonstrates that DFS best routing candidates are comparable to the hop count performance of best GFG routing candidates.

5.3 Performance Evaluation of DFS QoS Routing

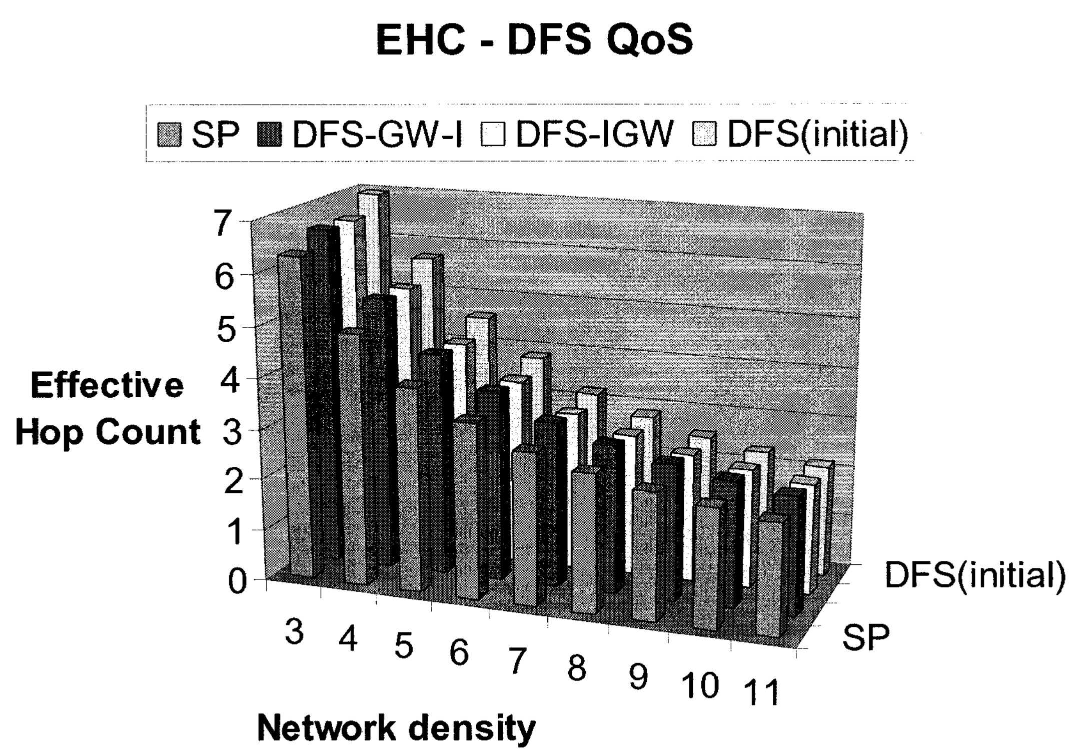
This section evaluates the performance of the DFS QoS routing algorithm. The DFS-based routing framework is reused for analysis of DFS QoS routing. QoS routing requirements, when applied on the DFS algorithm, result in some edges being removed from the used DFS resulting paths. The DFS QoS analysis did not require a separate set of QoS routing experiments to be executed since the remaining subgraph is also represented by pair (n, d) (d is the average degree of the remaining subgraph).

Instead of a regular hop count, a so-called effective hop count (EHC) was measured. In a nutshell, this is the DFS path without edges that were marked as those that rejected or returned messages. As discussed in 3.4, DFS Based QoS Routing, the path created by eliminating these unnecessary hops is required QoS path. The EHC of resulting QoS path can be calculated as:

$$EHC = hopCount - 2*rejectCount - 2*returnCount$$

Degree (*d*)	3	4	5	6	7	8	9	10	11
DFS (initial)	6.938	5.713	4.596	3.876	3.225	2.859	2.551	2.356	2.189
DFS-Int	6.739	5.548	4.485	3.807	3.183	2.835	2.542	2.356	2.187
DFS-IGW	6.614	5.342	4.319	3.658	3.101	2.792	2.524	2.342	2.182
DFS-IGW-I	6.614	5.341	4.319	3.657	3.1	2.794	2.525	2.343	2.181
DFS-GW	6.624	5.363	4.378	3.751	3.24	2.914	2.661	2.461	2.309
DFS-GW-I	6.625	5.363	4.377	3.75	3.239	2.914	2.663	2.461	2.31
SP	6.33	4.955	4.012	3.445	2.994	2.708	2.488	2.318	2.171

Table 5-6 Effective hop counts (EHC) for DFS QoS routing (*n*=40)



	3	4	5	6	7	8	9	10	11
▨ SP	6.33	4.96	4.01	3.45	2.99	2.71	2.49	2.32	2.17
■ DFS-GW-I	6.63	5.36	4.38	3.75	3.24	2.91	2.66	2.46	2.31
☐ DFS-IGW	6.61	5.34	4.32	3.66	3.1	2.79	2.52	2.34	2.18
▨ DFS(initial)	6.94	5.71	4.6	3.88	3.23	2.86	2.55	2.36	2.19

Figure 5-6 ***Graphical representation of data from Table 5-6***

As per Table 5-6 Effective hop counts (EHC) for DFS QoS routing it is evident that EHC results are significantly improved over those reported Table 5-1. This is evident for the various DFS-routing algorithms. DFS QoS routing performance improvement is greatest for those DFS routing protocols that are characterized by the most return and reject messages. The best DFS routing protocols, DFS-GW and DFS-GW-I, as expected, offer better EHC performance.

The Table 5-6 results demonstrate other interesting characteristics of the DFS QoS routing scheme. QoS performance of all listed DFS algorithm variations fall in the same range (for a certain network configuration (*n,d*)), i.e. QoS performance of the worst DFS performer gets quite close to QoS performance of the best DFS performer (hop count performance). It should be noted that these data prove that DFS QoS routing offers almost the shortest path route for QoS routing in localized manner. Excellent results are present for low degree networks. Well connected networks (higher factor *d*) offered equally good results (quite close to SP performance).

5.4 Performance Evaluation of DFS Progress Based Power Aware Routing Algorithm

In this section, a new set of routing algorithms, as introduced in 4.1, is evaluated through various set of experiments. More specifically, this section data, presented in Table 5-7 and Table 5-8, deal only with DFS Power Progress, power aware routing algorithm. Section 4.1 describes used power-aware metrics, required by routing algorithms for a neighbor selection process.

For this set of routing experiments, DFS-GW-I variant of DFS algorithm was selected and merged with Power Progress scheme. As discussed in Chapter 5.2, this is the best DFS candidate for $d<7$ (both analyzed network sizes, n=40 and n=100). Based on this data, it was decided to execute a simulation of the Power Progress DFS algorithm on the improved version of DFS routing that executes over dominating set based on gateway internal nodes.

Two power models, HCB and RM (as presented in Chapter 2.6), were simulated. Only HCB simulation data are included due to the fact that RM demonstrated similar patterns. As per the [HCB] power model, the energy needed for one transmit and receive event was:

$$u(r) = r^2 + 2*1000$$

(for nodes at distance r). This energy was split between the sender node $(r^2 + 1000)$ and the fixed power for reception is charged to the receiving node (1000). Initial energy levels for network nodes were set to be 200000. Two set of data is included (two different network sizes):

- Table 5-7 presents results for networks with n=40 nodes.
- Table 5-8 presents results for networks with n=100 nodes.

Each one of these set of data include a set of network density values (average number of neighbors per network node): d=[4,10].

As in Chapter 5.2, an identical simulation and execution environment was used. All presented results are average values of 100 executions over randomly created graphs. Unconnected graphs were immediately rejected. In addition to the new power-aware algorithm, DFS Power Progress, power consumptions of various variations of DFS routing algorithm are included as well.

Included DFS algorithm variations are explained with more details in Chapter 5.2. Table 5-7 and Table 5-8 present the same set of DFS routing experiments, this time with focus on the power aware aspects of these algorithms (not the hop count).

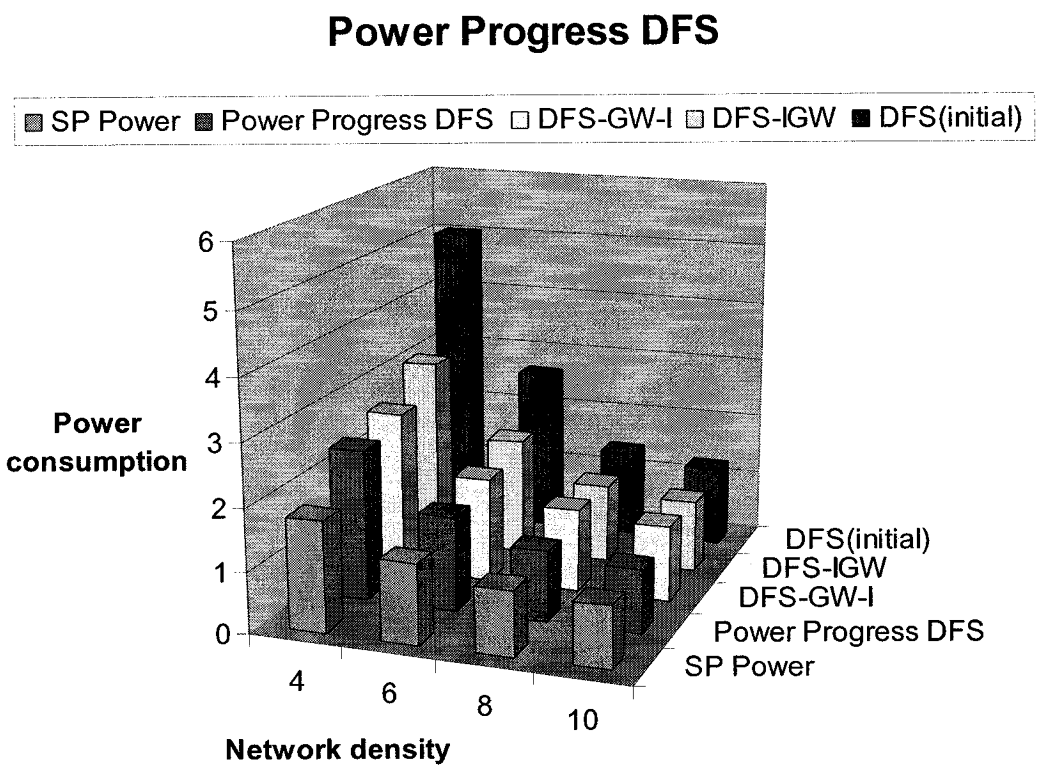
This is the short description for these shortcuts:

- DFS-Int: DFS routing based on intermediate nodes.
- DFS-IGW: DFS routing based on intergateway nodes.
- DFS-IGW-I: DFS-IGW with DFS improvement (as discussed in 3.2)
- DFS-GW: DFS routing based on gateway nodes.
- DFS-GW-I: DFS-GW with DFS improvement (as discussed in 3.2)
- DFS Power Progress: This is a new routing algorithm, based on the merge between latest power aware routing scheme, so called power progress, and DFS algorithm (DFS-GW-I candidate).
- SP Power: This is non-localized best possible power-aware routing within established network scenarios. It possesses a global knowledge about the network and is capable of routing a message across the network with minimal energy consumption. The SP Power routing performance is the actual baseline used for comparison and understanding power consumptions of all included routing algorithms.

Degree (*d*)	4	6	8	10
DFS (initial)	5.045	2.666	1.411	1.269
DFS-Int	4.099	2.376	1.347	1.247
DFS-IGW	3.112	1.893	1.288	1.168
DFS-IGW-I	2.746	1.716	1.264	1.157
DFS-GW	2.831	1.756	1.336	1.221
DFS-GW-I	2.627	1.686	1.322	1.218
Power Progress DFS	2.443	1.516	1.15	1.025
SP Power	1.771	1.282	1.049	0. 9485

Table 5-7 Power consumption for all routing tasks for *n*=40

All numbers in **Table 5-7** are ten to the power of seven.



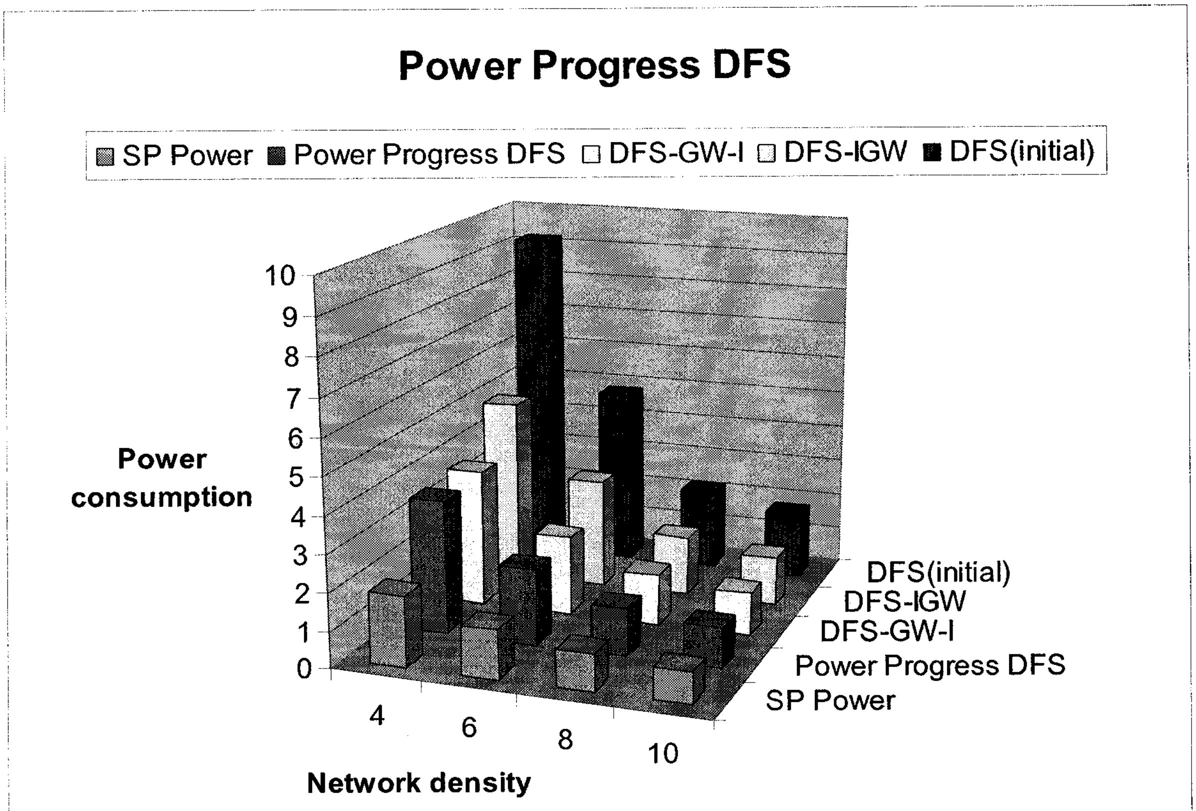
	4	6	8	10
▨ SP Power	1.771	1.282	1.049	1
■ Power Progress DFS	2.443	1.516	1.15	1.025
□ DFS-GW-I	2.627	1.686	1.322	1.218
▨ DFS-IGW	3.112	1.893	1.288	1.168
■ DFS(initial)	5.045	2.666	1.411	1.269

Figure 5-7 ***Graphical representation of data from Table 5-7***

Degree (*d*)	**4**	**6**	**8**	**10**
DFS (initial)	9.2	4.845	2.199	1.791
DFS-Int	7.23	4.224	2.073	1.7
DFS-IGW	4.965	2.976	1.625	1.312
DFS-IGW-I	4.005	2.293	1.376	1.135
DFS-GW	4.277	2.45	1.503	1.216
DFS-GW-I	3.725	2.117	1.367	1.15
Power Progress DFS	3.615	2.029	1.294	1.077
SP Power	1.884	1.308	0.993	0.867

Table 5-8 Power consumption for all routing tasks for *n*=100

All numbers in **Table 5-8** are ten to the power of eight.



	4	6	8	10
▨ SP Power	1.884	1.308	0.993	0.867
■ Power Progress DFS	3.615	2.029	1.294	1.077
□ DFS-GW-I	3.725	2.117	1.367	1.15
□ DFS-IGW	4.965	2.976	1.625	1.312
■ DFS(initial)	9.2	4.845	2.199	1.791

Figure 5-8 ***Graphical representation of data from Table 5-8***

Presented data in Table 5-7 and Table 5-8 shows the strength of newly proposed DFS routing algorithms. It appears that the contribution to the saving of power spending appears to be greater for networks with higher density. This is the case for both tested scenarios, *n*=40 and *n*=100.

In addition, as expected, various DFS algorithm power consumptions seem to be in direct link with the number of hop counts that characterize these schemes. Associated hop count simulation data of these algorithms are in Table 5-1 and Table 5-2.

DFS Power Progress scheme spends ~15% less energy than DFS-GW-I algorithm for *n*=40 and *d*=10. This improvement is not as good for *n*=40 and *d*=4, which is close to ~7%. Similarly, the saving appears to be ~14% for *n*=100 and *d*=10 and much less for *n*=100 and *d*=4, only ~3%.

The total power consumption is quite close to the ideal spending for smaller networks and higher values of network density. The deviation of DFS Power Progress is only ~7% for *n*=40 and *d*=10 and ~9% for *d*=9. These numbers are higher for the second simulated case, with *n*=100 network nodes. The deviation for *d*=10 is ~20% and ~23% for *d*=8.

5.5 Performance Evaluation of DFS Progress Based Cost Aware Routing Algorithms

In this section, a new set of routing algorithms, as introduced in section 4.1, is evaluated through various set of experiments. More specifically, presented data in Table 5-9 and Table 5-10, deal only with DFS cost aware progress based routing algorithms, DFS Cost Progress and DFS PowerCost Progress schemes. Section 4.1 describes used cost and power-cost aware metrics, required by routing algorithms for a neighbor selection process.

Similarly to section 5.4, for this set of routing experiments, DFS-GW-I variant of DFS algorithm was selected and merged with cost aware progress schemes. As discussed in Chapter 5.2, this is the best DFS candidate for d<7 (both analyzed network sizes, n=40 and n=100). Therefore, the simulation of DFS Cost Progress and DFS PowerCost Progress algorithms is based on the improved version of DFS routing that executes over dominating set based on gateway internal nodes.

As in Chapter 5.4, we present only data for the HCB power model. All explained conditions and the simulation details are reused for this set of experiments as well. As in Chapter 5.4, two sets of data are included (two different network sizes):

- Table 5-9 presents results for networks with n=40 nodes.
- Table 5-10 presents results for networks with n=100 nodes.

Besides known DFS algorithm variations, two new, cost-aware DFS based routing schemes are added to the list of monitored algorithms:

- DFS Cost Progress: This is the resulting algorithm, generated by integrating the latest version of cost aware algorithm, so-called cost progress (discussed in Chapter 2.6.5), and DFS routing framework.
- DFS PowerCost Progress: This is the resulting algorithm, generated by integrating the latest version of power-cost aware algorithm (discussed in Chapter 2.6.5), so called power-cost progress, and DFS routing framework.
- SP Cost: This is non-localized best possible cost-aware routing within established network scenarios. It posses a global knowledge about the network and is capable of routing as many iterations as possible by the network before one network node drains its battery. The SP Cost routing performance is the actual baseline used for

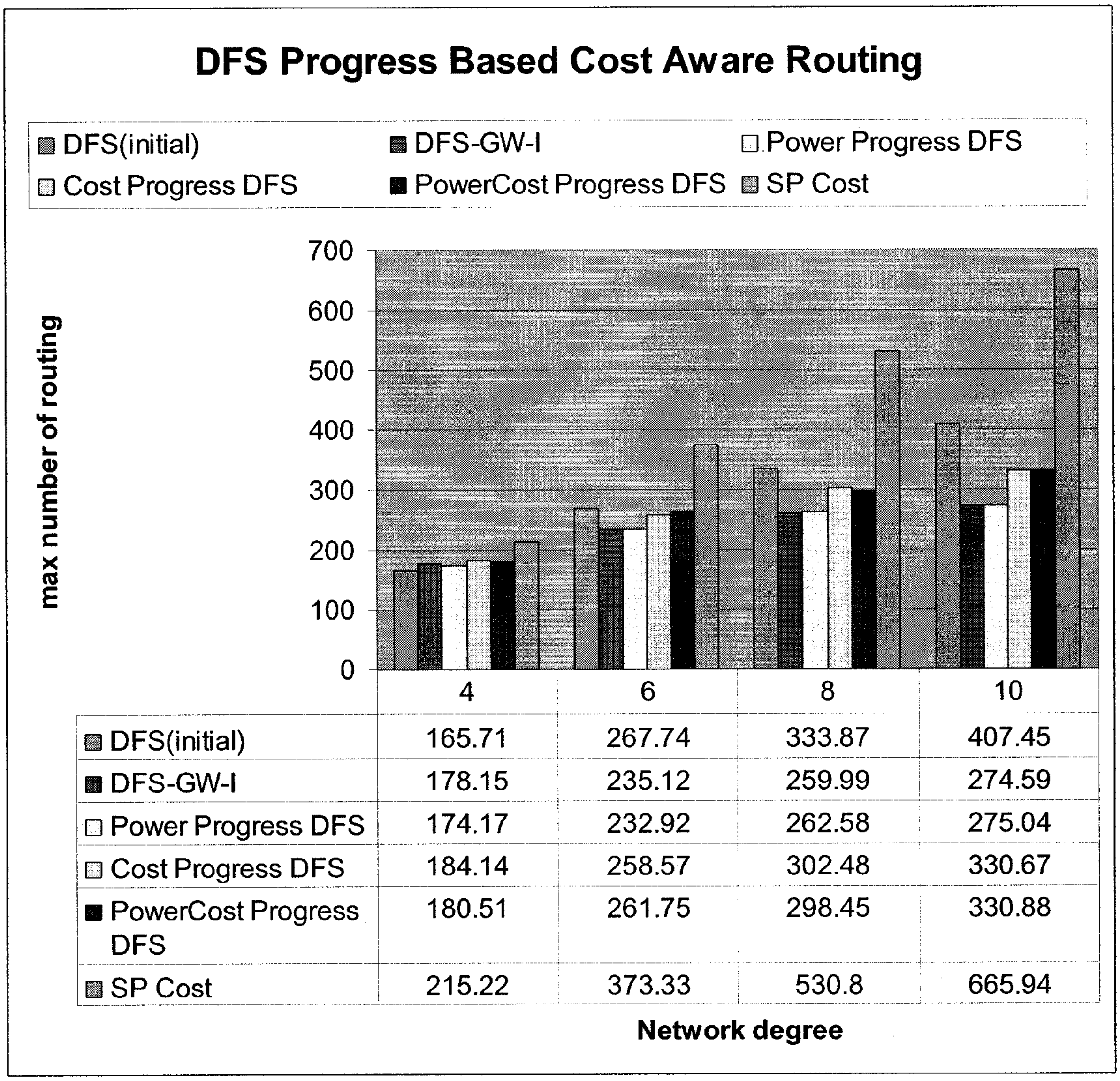
comparison and understanding various power consumption patterns of analyzed routing algorithms. It simply states the biggest number of messages that network can route before one of its nodes drains all energy (and becomes unavailable).

Presented data, in Table 5-9 and Table 5-10, define the number of iterations for specified routing algorithm, before one of network nodes becomes unavailable. This was set as the stoppage criteria – no further testing was conducted at this moment to confirm if the new network topology became disconnected or not (due to the event of node disappearing).

As currently constructed, the simulation randomly chooses source and destination points and routes the message throughout the network. Along the way, and according to HCB power model, network nodes that route messages update their battery levels to reflect executed routing requests. Upon the delivery of the message, a new source-destination pair is randomly selected and routing through the network repeats. As the message travels, the energy level of network nodes that route the message is monitored. As soon as one of nodes energy level reaches zero level, the simulation stops and exits.

Degree (*d*)	**4**	**6**	**8**	**10**
DFS (initial)	165.71	267.74	333.87	407.45
DFS-Int	166.5	269.36	336.53	408.9
DFS-IGW	174.9	267.22	330.65	383.38
DFS-IGW-I	180.92	266.27	329.43	380.69
DFS-GW	173.08	236.74	260.47	272.01
DFS-GW-I	178.15	235.12	259.99	274.59
DFS Power Progress	174.17	232.92	262.58	275.04
DFS Cost Progress	184.14	258.57	302.48	330.67
DFS PowerCost Progress	180.51	261.75	298.45	330.88
SP Cost	215.22	373.33	530.8	665.94

Table 5-9 Number of iterations for *n*=40 (before one network node drains battery)

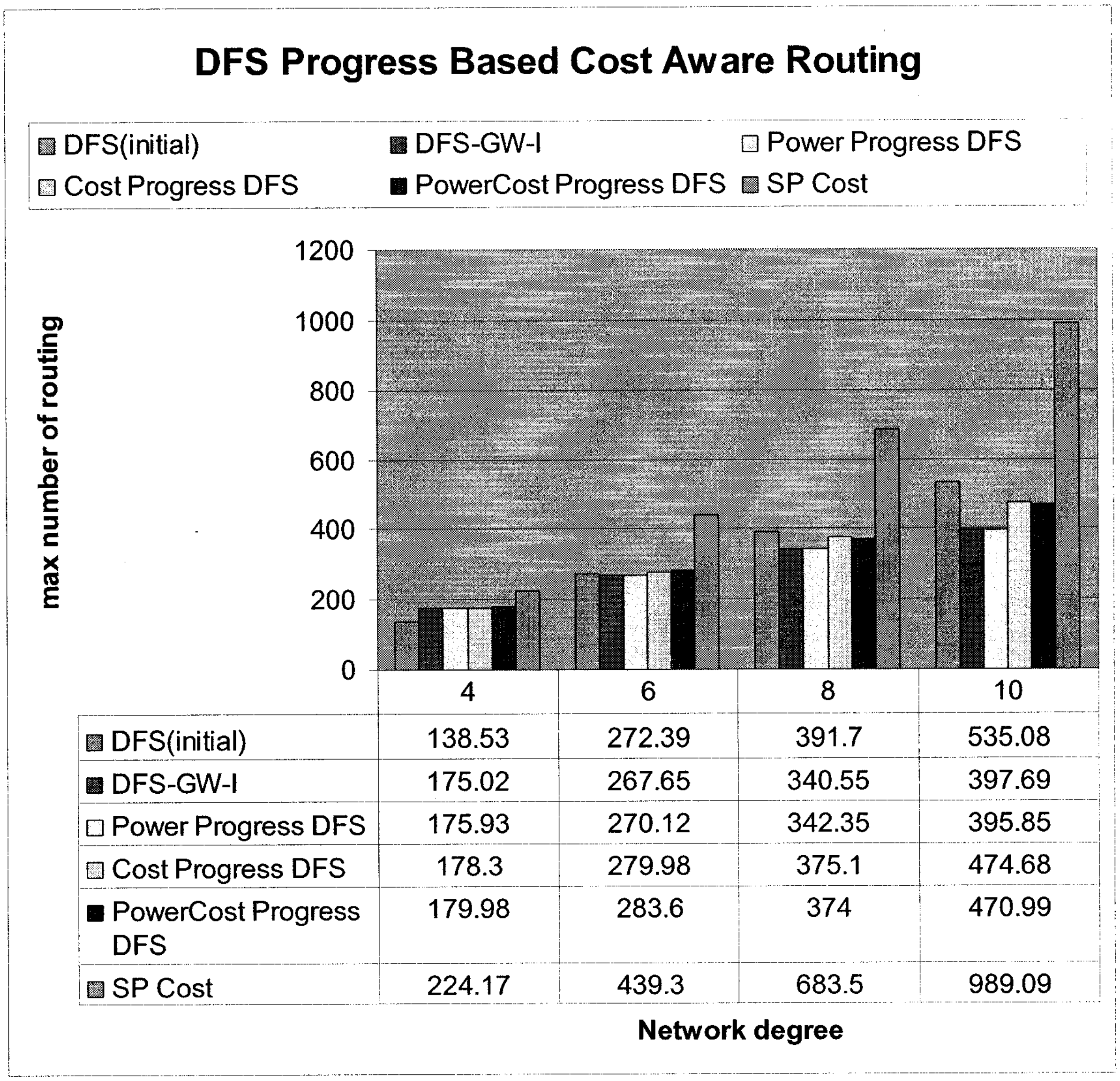


	4	6	8	10
▨ DFS(initial)	165.71	267.74	333.87	407.45
■ DFS-GW-I	178.15	235.12	259.99	274.59
☐ Power Progress DFS	174.17	232.92	262.58	275.04
▨ Cost Progress DFS	184.14	258.57	302.48	330.67
■ PowerCost Progress DFS	180.51	261.75	298.45	330.88
▨ SP Cost	215.22	373.33	530.8	665.94

Figure 5-9 ***Graphical representation of data from Table 5-9***

Degree (*d*)	4	6	8	10
DFS (initial)	138.53	272.39	391.7	535.08
DFS-Int	147.85	276.25	392.5	533.03
DFS-IGW	162.4	282.16	390.1	521.76
DFS-IGW-I	179.69	289.71	394.85	525.17
DFS-GW	165.16	260.73	332.3	397.21
DFS-GW-I	175.02	267.65	340.55	397.69
DFS Power Progress	175.93	270.12	342.35	395.85
DFS Cost Progress	178.3	279.98	375.1	474.68
DFS PowerCost Progress	179.98	283.6	374	470.99
SP Cost	224.17	439.3	683.5	989.09

Table 5-10 Number of iterations for *n*=100 (before one network node drains battery)



Network degree	4	6	8	10
▨ DFS(initial)	138.53	272.39	391.7	535.08
■ DFS-GW-I	175.02	267.65	340.55	397.69
□ Power Progress DFS	175.93	270.12	342.35	395.85
▨ Cost Progress DFS	178.3	279.98	375.1	474.68
■ PowerCost Progress DFS	179.98	283.6	374	470.99
▨ SP Cost	224.17	439.3	683.5	989.09

Figure 5-10 ***Graphical representation of data from Table 5-10***

The focus of the simulation efforts in this section is on two DFS cost-aware algorithms, DFS Cost Progress and DFS PowerCost Progress. These two routing schemes will maximize the number of routing tasks the network can perform before one of its nodes exhausts the energy. DFS PowerCost Progress will try to select power efficient routes among the cost optimal ones.

It should be no surprise that initial DFS cost- and power-aware performance is best due to the fact that all network nodes are used. For example, Table 5-3 shows that initial DFS routing is based on 250 network nodes (all of them), while DFS-GW is using only a subset of these, 147 nodes. DFS power- and cost-aware candidates are improved versions of DFS-GW (or DFS-IGW), therefore, these routing schemes will also use only a subset of network nodes. Consequently, the power- and cost-aware performance of these will be lower than achieved numbers of initial DFS scheme.

Presented data in Table 5-9 and Table 5-10 underline the similarity between cost aware routing performances in both network sizes. Other relevant conclusions coming out of this data include:

- There are three groups of routing protocols, each one of them with quite similar performances.

  <u>First one</u> is defined with first four algorithms (initial DFS, DFS-Int, DFS-IWG, DFS-IWG-I). Interesting to note is that the introduction of dominating sets, based on intergateway nodes, did not reduce the number of successful iterations, even though only the subset of total network nodes is used for the actual routing and delivery of the messages. This behavior seems to be consistent across both network sizes (n=40 and n=100) and used network density parameter values (d=4,6,8,10).

  <u>Second group</u> include DFS variations based on dominating sets that use gateway nodes, DFS-GW and DFS-GW-I. The proposed DFS Power Progress algorithm has similar performance as well. It proved to be successful (as discussed in Chapter 5.4) in reducing the total power needed for routing tasks. However, it is not sensitive to nodes with critical

energy levels encountered along routes. Therefore, its performance is as good as other two DFS gateway-based routing schemes. Again, this proved to be consistent for all included parameters.

Third group include two new DFS cost-aware protocols, DFS Cost Progress and DFS PowerCost Progress. These two offer consistent improvements for included test cases (different network sizes and network density values) over DFS gateway based variations.

- The improvement rate of the third group of protocols over the cost aware routing performance of the second group of protocols seems to be higher for the network with less nodes (n=40). The cost-aware performance is improved ~6% (d=4), ~11% (d=6), and ~15% (d=8) vs. bigger network (n=100) improvements of ~1.5% (d=4), ~4% (d=6), and ~10% (d=8).

  In addition, it should be noted that this performance is obviously better for the networks with higher network density parameter value. This is consistent for both monitored networks.
- The performance of the listed third group in most cases reaches or get quite close to the cost aware performance of the discussed first group. This is an important achievement, since these DFS cost aware algorithms base their routing only on a reduced number of network nodes (gateway nodes). Therefore, they have proved to be able to keep the hop count and the total consumed power as low as possible; yet, the number of possible routing tasks is kept quite high (in some cases, equal or better that the best cost aware results of the first group of protocols).

Chapter 6

Conclusion and Future Work

This thesis focused on two goals. Foremost, the DFS routing algorithm was introduced and various improvements to the initially proposed DFS routing scheme were simulated. The best candidate(s) were identified and selected to be merged into a newly proposed set of DFS power- and cost-aware localized routing schemes. In addition, the problem of QoS routing in wireless ad hoc networks was addressed by extending the DFS routing scheme to improve the propagation delay requirement. Presented research was simplified by focusing only on static wireless networks with network nodes whose availability did not change.

Simulation data has helped to identify the best DFS routing candidate(s). This selection depended on the network size and selected network density values. A small subset of simulated test cases have shown improved DFS routing based on gateway nodes to offer the best hop count performance. These are mostly limited to density values within range d=[3,6] (n=40), d=[4,8] (n=100), and d=[5,7] (n=250). In all other simulated network types (network sizes and network density values), simulation data has proved the improved DFS routing based on intergateway nodes to be the ideal candidate.

Proposed improvements of initial DFS routing algorithm succeed in delivering significant benefits to the hop count performance. The best candidates' performances are extremely close to the performance of the shortest path metrics for smaller and well connected networks. This improvement success is significant as well for bigger and well connected networks (however not as good as the improvement for the smaller and well connected networks). For networks with smaller network density values, the hop count metrics of monitored DFS candidates is not as close to the ideal, shortest path. However, in these cases, the proposed best DFS candidates offer the best absolute improvements of the initial DFS routing scheme.

The best DFS routing candidate was merged with power aware schemes. A new set of power- and cost-aware DFS based algorithms appear to have several important features. The DFS framework extends its ability to guarantee delivery on these blended set of power- and cost-aware routing algorithms that, as expected in their original form, do not deliver all of the time. At the same time, the power- and cost-aware performance of the DFS routing candidate is also improved. Power-aware improvement seems to be larger for smaller well connected networks and not as significant for larger networks that are not well connected. The improvement to the cost-aware performance demonstrates identical pattern as is the case with the power-aware performance. The important thing to note is that power- and cost-aware improvements of newly proposed set of power and cost DFS-based routing protocols are present across different network sizes and for different network density values.

It was shown how DFS QoS generated paths offer performance that is very close to the QoS path generated by applying shortest path algorithm. This performance is obtained in a localized manner, while the shortest path algorithm requires a global knowledge of the network. The DFS QoS framework requires a certain overhead – needed to generate QoS path. This overhead is required in order to learn about return and rejected messages and is significantly reduced in improved version of DFS routing algorithms. The outlined DFS QoS routing performance data positions DFS based routing algorithm as an excellent candidate for QoS routing in wireless ad hoc networks.

The DFS strategy can make a major improvement in routing performance in wireless sensor networks that are formed in ad hoc manner. The absence of node movements in these applications, after the network is formed, is identical to the simulation conditions used in this research work.

DFS offers superb performance, guaranteed delivery (even when a destination location is not fully correct) and supports applications that are sensitive to data loss. Some of previously discussed applications, that would be hosted within these, sensor-based wireless networks, are going to need data continuity and guaranteed delivery capability of underlying routing framework. If QoS is required, then adapted DFS routing, DFS QoS, can be deployed in order to support this type of requests for application execution environment.

The exclusion of mobility from this thesis did not reduce the significance of published results in both areas of the thesis (the DFS routing algorithm and the introduction of a new set of routing protocols). It is fair to assume that selected best candidate DFS algorithm(s) will easily integrate with location update protocols. A source node which is about to start DFS routing can use available location update scheme(s) to determine the correct location of the destination node. Once the destination information is available, DFS routing would initiate the routing task independently of the location update component, and based on the data provided by these location update mechanisms.

Researched DFS routing algorithms prove to be ideal candidates with their guaranteed delivery capability, even for special cases, where location information is not fully correct. They achieve that by switching from greedy mode to recovery mode, as long as it is needed or the destination is reached. This is going to be very important when mobility is taken into the account. It is fair to assume that the latest and up to date destination location, obtained by recent location update request, might change due to the movement of the destination node, while the message travels through the network. However, DFS would make sure that even this case is resolved, i.e. the message reaches the required destination node.

In a same manner, introduction of wake-up power-based management schemes and integration with selected the DFS candidate, is expected to be relatively easy. It can be implemented on the node level, in a localized manner, as proposed by the power management algorithm and independent of the DFS routing framework. As expected, the only requirement would be for the wireless network to stay connected and for the destination node to be awake – in order for the delivery to succeed. The DFS algorithm, as already explained, will not be obstructed in its delivery capability. If one or few neighboring nodes are not available, i.e. it is in the 'sleep' mode, the greedy mode of DFS will select other available nodes and routing will continue to execute. If no more neighboring nodes are present, the node DFS routing logic would switch to the recovery mode, until greedy routing mode would become possible. The delivery is going to be guaranteed.

It is important to note that the new set of DFS power and cost aware routing algorithms prove to be an excellent candidate for future integration with the location update protocol and wake-up power-based management scheme. Its ability to deliver the message even if the location information of the destination is not fully correct might be a turning point in the selection process.

6.1 Future Work

There are several possible directions for associated research of potential improvements to introduced DFS-based power- and cost-aware routing algorithms. The following lists the major directions for the work:

- The hop count might be improved by applying Gabriel graph or RNG concepts, described in [BMSU, DSV]. These concepts could be used instead of a dominating set theory or in a conjunction with it.
- Potentially interesting and related future directions could be based on further improvements of used dominating set definitions. The inevitable problem of routing algorithms based on dominating sets is caused by the fact that dominating set nodes execute many more tasks than other nodes. [WDGS], as a solution to this problem, proposed to power aware dominating set definition. In this definition, each dominating set node has a key *(power level, degree, id)* for deciding its dominating set status. Thus nodes use their power levels as the primary criterion. Nodes with more power are preferred in the dominating set selection process. If power levels are same, degrees are used as secondary key, and finally node *ID* would be used to break ties. The further improvement is proposed in [STW], where the primary key is a linear combination of power level and degree, that is *a*power_level* + *b*degree,* where *a* and *b* are parameters whose best values need to be experimentally determined.
- Cost aware simulation efforts could be additionally improved. As it is currently designed, random selection of source-destination pairs is completely independent between different simulation phases (separate monitoring of cost aware performance of routing algorithms). The direct outcome is that cost aware performances of different algorithms are influenced by this randomness – the randomness in the selection of source-destination pairs. The comparison would be fairer and results would reflect the true cost aware ability of monitored algorithms if the same set of source-destination pairs was reused for all simulation phases.

- [SD] proposed another set of power- and cost-aware routing algorithms that guarantee the delivery. It would be interesting to compare simulation data of this thesis work with those from [SD] paper.
- Only one square size of wireless ad hoc networks was exclusively used in simulation efforts - [0,100). It would be interesting to confirm how the simulation data would be different, if any, when these established improvement patterns are reused for identical simulation environment parameters, but with different network square sizes.
- Used power models ([RM] and [HCB]) divide consumed power of one hop between sender and receiver nodes. As outlined in Chapter 5.4, the transmission power is charged to a sender node which depends on the distance between nodes. A fixed power for reception is charged to a receiving node.

  This model should be additionally expanded in order to reflect the fact that a message, sent to a particular node, is also heard by other neighboring nodes. However, not all neighboring nodes will be able to hear the message transmission event. Since the power is adjusted to the minimal necessary for successful reception of the message, only nodes that are closer to the sender node than the destination node will be able to receive it. The optimization function for power aware routing algorithm would need to take into account all power spent by a message transmission and receiving events. Similar adjustments can be made for cost and power-cost routing algorithms.

Chapter 7

References

[BCSW] S. Basagni, I. Chlamtac, V.R. Syrotiuk, B.A. Woodward: *A distance routing effect algorithm for mobility (DREAM).* Proc. MOBICOM, 1998, 76-84.

[BFNO] L. Barriere, P Fraigniaud, L. Narajanan, and J. Opatrny: *Robust position based routing in wireless ad hoc networks with unstable transmission ranges.* Proc. of 5th ACM Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications DIAL M01, 2001

[BMJHJ] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, J. Jetcheva: *A performance comparison of multi-hop wireless ad hoc network routing protocols.* Proc. MOBICOM, 1998, 85-97.

[BMSU] P. Bose, P. Morin, I. Stojmenovic and J. Urrutia: *Routing with guaranteed delivery in ad hoc wireless networks.* ACM Wireless Networks, 7, 6, November 2001, 609-616; Proc. of 3rd ACM Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications DIAL M99, Seattle, August 20, 1999, 48-55.

[CHH] S. Capkun, M. Hamdi, J.P. Hubaux: *GPS-free positioning in mobile ad-hoc networks.* Proc. Hawaii Int. Conf. on System Sciences, January 2001

[CN] S. Chen and K. Nahrstedt: *Distributed quality-of-service routing in ad hoc networks, IEEE J. Selected Areas in Communications.* 17, 8, August 1999, 1488-1505.

[DSV] S. Datta, I. Stojmenovic, J. Wu: *Internal node and shortcut based routing with guaranteed delivery in wireless networks.* Cluster Computing, 5, 2, April 2002, 169-178.

[EGHK] D. Estrin, R. Govindan, J. Heidemann, S. Kumar: *Next century challenges: Scalable coordination in sensor networks.* Proc. MOBICOM, 1999, Seattle, 263-270.

[F] G.G. Finn: *Routing and addressing problems in large metropolitan-scale internetworks.* ISI Research Report ISU/RR-87-180, March 1987.

[Fe] L.M. Feeney: An energy-consumption model for performance analysis of routing protocols for mobile ad hoc networks. Mobile Networks and Applications, to appear.

[FN] L.M. Feeney, M. Nilsson: Investigating the energy concumption of a wireless network interface in an ad hoc networking environment. Proc. IEEE INFOCOM, 2001.

[HCB] W.R Heinzelman, A. Chandrassekaran, H. Balakrishnan: *Energy efficient routing protocols for wireless microsensor networks.* Proceedings Hawaii International Conference on System Sciences, January 2000.

[JPS] R. Jain, A. Puri and R. Sengupta: *Geographical routing using partial information for wireless ad hoc networks.* IEEE Personal Communication, February 2001, 48-57.

[KFWM] W. Kieß, H. Füßler, J. Widmer, and M. Mauve: *Hierarchical location service for mobile ad-hoc networks.* ACM Mobile Computing and Communications Review (MC2R), Vol. 8, Issue 4, Oct 2004.

[KK] Vikas Kawadia, P. R. Kumar: *Power Control and Clustering in Ad Hoc Network.* IEEE Infocom 2003.

[KNS3] Johnson Kuruvila, Amiya Nayak, and Ivan Stojmenovic: *Progress based localized power and cost aware routing algorithms for ad hoc and sensor wireless networks.* Third Int. Conf. on AD-HOC Networks and Wireless ADHOC-NOW, Vancouver, BC, July 22-24, 2004; LNCS 3158, 294-299; International Journal of Distributed Sensor Networks , to appear.

[KSU] E. Kranakis, H. Singh and J. Urrutia: *Compass routing on geometric networks.* Proc. 11th Canadian Conference on Computational Geometry, Vancouver, August, 1999.

[KV] Y.B. Ko and N.H. Vaidya: *Location-aided routing (LAR) in mobile ad hoc networks.* MOBICOM, 1998, 66-75.

[LL] C. R. Lin and J.S. Liu: *QoS routing in ad hoc wireless networks*. IEEE J. Selected Areas in Communications, 17, 8, Aug. 1999, 1426-1438.

[LSR] S. Lindsey, K. Sivalingam and C.S. Raghavendra: *Power optimization in routing protocols for wireless and mobile networks.* Wireless Networks and Mobile Computing Handbook, Stojmenovic - I (ed.). John Wiley & Sons: 2002; 407–424.

[LTS] W.H. Liao, Y.C. Tseng, and J.P. Sheu: *GRID: a fully location-aware routing protocol for mobile ad hoc networks.* Telecommunication Systems, 18, 61-84, 2001.

[MC] J.P. Macker and M.S. Corson: *Mobile ad hoc networking and the IETF.* Mobile Computing and Communications Review. 2, 1, 1998, 9-14.

[MWH] M. Mauve, J. Widmer, and H. Hartenstein: *A Survey on Position-based Routing in Mobile Ad Hoc Networks*. IEEE Network, Vol. 15 No. 6, 2001.

[NBD] R. Naik, S. Biswas, and S. Datta: *Distributed Sleep-Scheduling Protocols for Energy Conservation in Wireless Networks*. Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05), Hawaii, 2005.

[RM] V. Rodoplu, T.H. Meng: *Minimum energy mobile wireless networks.* IEEE Journal on Selected Areas in Communications, Vol. 17, No. 8, August 1999, 1333-1344.

[RT] E.M. Royer and C.K. Toh: *A review of current routing protocols for ad hoc mobile wireless networks.* IEEE Personal Communications, April 1999, 46-55.

[S1] I. Stojmenovic: *Comments and Corrections to "Dominating Sets and Neighbor Elimination-Based Broadcasting Algorithms in Wireless Networks".* IEEE Transactions on parallel and distributed systems, Vol. 15, No. 11, November 2004.

[S-cm] I. Stojmenovic: *Position-Based Routing in Ad Hoc Networks.* IEEE Commmunications Magazine, Vol. 40, No. 7, July 2002, 128-134.

[S-ha] I. Stojmenovic: *Home agent based location update and destination search schemes in ad hoc wireless networks.* Advances in Information Science and Soft Computing (A. Zemliak and N.E. Mastorakis, eds.), WSEAS Press, 2002, 6-11.

[S-lu] I. Stojmenovic: *Location Updates for Efficient Routing in Ad Hoc Networks.* Handbook of Wireless Networks and Mobile Computing, Wiley, 2002, pp. 451–71.

[S-q] I. Stojmenovic*: A scalable quorum based location update scheme for routing in ad hoc wireless networks*. SITE, University of Ottawa, TR-99-09, September 1999.

[SD] I. Stojmenovic, S. Datta: *Power and cost aware localized routing with guaranteed delivery in wireless networks.* Wireless Communications and Mobile Computing, 4, 2, 2004, 175-188.

[SD-th] Susanta Datta: *Internal nodes and shortcut based, power and cost aware routing with guaranteed delivery in wireless networks.* Thesis paper (MSc in Computer Science) – University of Ottawa, SITE School, 2001.

[SL] I. Stojmenovic, Xu Lin: *Power-aware localized routing in wireless networks.* IEEE Int. Parallel and Distributed Processing Symp., Cancun, Mexico, May 1-5, 2000, 371-376; IEEE Transactions on Parallel and Distributed Systems, Vol. 12, No. 11, November 2001, 1122-1133.

[SL1] I. Stojmenovic and X. Lin: *Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks.* IEEE Transactions on Parallel and Distributed Systems, Vol. 12, No. 10, October 2001, 1023-1032.

[SR-TR] I.Stojmenovic and M.Russell: *Depth first search and location based localized routing and QoS routing in wireless networks.* TR-2000-01. January 2000.

[SRV1] I. Stojmenovic, M. Russell, and B. Vukojevic: *Depth first search and location based localized routing and QoS routing in wireless networks.* IEEE International Conference on Parallel Processing. August 21-24, 2000, Toronto, 173-180.

[SRV2] I. Stojmenovic, M. Russell, and B. Vukojevic: *Depth first search and location based localized routing and QoS routing in wireless networks.* Computers and Informatics, Vol. 21, No. 2, 2002, 149-165.

[SSS] M. Seddigh, J. Solano Gonzalez and I. Stojmenovic: *RNG and internal node based broadcasting algorithms for wireless one-to-one networks.* ACM Mobile Computing and Communications Review, 5, 2, April 2001, 37-44.

[SSZ] I. Stojmenovic, M. Seddigh, J. Zunic: *Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks.* IEEE Trans. on Parallel and Distributed Systems, Vol. 13, No. 1, January 2002, 14-25.

[SSSW] J. Shaikh, J. Solano, I. Stojmenovic and J. Wu: *New metrics for dominating set based energy efficient activity scheduling in ad hoc networks.* Proc. WLN Workshop at IEEE Conf. on Local Computer Networks, Bonn, Germany, Oct. 20-24, 2003.

[STW] I. Stojmenovic, Y.C. Tseng and J. Wu: *Sleep period sensor network schedules based on power aware dominating sets*. In preparation.

[SWR] S. Singh, M. Woo, C.S. Raghavendra: *Power-aware routing in mobile ad hoc networks.* Proc. MOBICOM, 1998, 181-190.

[T] A. Trevor: *Wake-up Based Power Management in Multi-hop Wireless Network.* http://www.eecg.toronto.edu/~trevor/Wakeup/survey.pdf

[TK] H. Takagi and L. Kleinrock: *Optimal transmission ranges for randomly distributed packet radio terminals.* IEEE Transactions on Communications, 32, 3, 1984, 246-257.

[WC] Z. Wang and J. Crowcroft: *Quality-of-service routing for supporting multimedia applications.* IEEE J. on Selected Areas in Communications, 14, 7, 1996, 1228-1234.

[WDGS] J. Wu, F. Dai, M. Gao, and I. Stojmenovic: *On calculating power-aware connected dominating set for efficient routing in ad hoc wireless networks.* IEEE/KICS Journal of Communications and Networks, 4(1):59–70, March 2002.

[WL] J. Wu, H. Li: *On calculating connected dominating set for efficient routing in ad hoc wireless networks.* Proc. DIAL M, Seattle, Aug. 1999, 7-14.

Appendix A

Acronyms or Glossary

AODV	Ad hoc On demand Distance Vector, a routing protocol for ad hoc networks (based on DSDV).
AP	Access Point, a base station or other place for connection to an available infrastructure.
DFS	Depth First Search routing algorithm.
DSDV	Destination Sequence Distance Vector, an early protocol for routing in ad hoc networks.
GRA	Geographic routing algorithm.
GPS	Global positioning system.
MANET	Mobile Ad hoc Network, a network that consists of mobile nodes that spontaneously can connect to each other without help from any preexisting infrastructure.
MH	Mobile Host, a mobile host in a network.
QoS	Quality of service.

Appendix B

DFS Routing Algorithm – Pseudo Code

The pseudo code of *DFS* based routing algorithm can be given as follows:

```
//A is the neighbor of B that is closest to D
Procedure DFS-forward(B, A, D);
    B forwards message to A;
    If A=D
    then
        message delivered
    else
    {
        If A is white node
        then
        {
            A is colored as gray node;
            A memorizes B;
            A sorts all neighbors (except B) according to distance from D as C1, C2, …, Ck;
            If k>0
            then
                //C1 is the neighbor of A that is closest to D
                DFS-forward(A, C1, D)
            else DFS-return(A, B, D)
        }
        else DFS-return(A,B,D);
    }
```

```
Procedure DFS-return(C,A,D);
  C returns message to A;
  A sorts all neighbors (except memorized neighbor B) according to distance from D as C1, C2, …, Ck;
  Let L be the index such that C=CL;
  If L < k
  then
    DFS-forward(A, CL+1, D)
  else
  {
    If A=S
    then
      D not connected to S
    else
      DFS-return(A,B,D);}

Program DFS-routing(S,D);
  Let C be the neighbor of S which is closest to D;
  DFS-forward(S, C, D);
```

Appendix C

Procedure internal-status(v)

As discussed in [S1], this is the correct procedure that a network node is required to use in order to determine its dominating set status. Once executed on all network nodes, their status flags will be set to one of possible values: intermediate, intergateway or gateway.

```
Procedure internal-status(v);
  intermediate(v) = intergateway(v) = gateway(v) = false

  for each neighbor u of v, u != v do
    for each neighbor w of v, w != v do
      if d(u;w) > R then intermediate(v) = true
  if intermediate(v) then {
    intergateway(v) = true;
    for each neighbor u of v, u != v do {
      covered = true;
      for each neighbor w of v, w != v;w != u do
        if d(u;w) > R or key(v) > key(u) then
          covered = false;
    if covered then intergateway(v) = false }
  if intergateway(v) then { gateway(v) = true;
    for each neighbor u of v, u != v do
      for each common neighbor w of v and u do {
        covered = true;
        for each neighbor z of v do {
          if (d(z; u) > R and d(z;w) > R) or
```

```
            key(v) > key(u) or
            key(v) > key(w)
          then covered = false;
    if covered then gateway(v) = false }}
```