The Pennsylvania State University
The Graduate School

PURPOSEFUL MOBILITY AND CAPACITY ISSUES IN SENSOR NETWORKS

A Thesis in
Electrical Engineering
by
Rajesh N. Rao

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2007

The thesis of Rajesh N. Rao was reviewed and approved* by the following:

George Kesidis
Professor of Electrical Engineering and Computer Science and Engineering
Thesis Advisor, Chair of Committee

Catherine M. Harmonosky
Associate Professor of Industrial and Manufacturing Engineering

John J. Metzner
Professor of Electrical Engineering and Computer Science and Engineering

David J. Miller
Associate Professor of Electrical Engineering

W. Kenneth Jenkins
Professor of Electrical Engineering
Head of Department of Electrical Engineering

*Signatures are on file in the Graduate School.

Abstract

In this thesis methods to exploit “purposeful” mobility to improve the efficiency and performance of a sensor network are presented. For example a mobile node experiencing local deep fades and shadowing can move to a different location for better channel conditions. Transmission power required to transmit over a distance d is here on assumed to be given by Kd^α where K is a constant and $\alpha \geq 2$ is the transmission attenuation factor. Mobility can reduce transmission power by reducing the transmission distance d. The thesis presents an algorithm to move sensor nodes to reduce transmission distance and hence transmission energy, spending less energy for motion compared to the energy saved in transmission over time.

In our sensor network, data generated by sensor nodes is aggregated at local sinks and forwarded to a central node. “Capacity” is defined as the number of sensor nodes that a sink can support. The number of data flows (each emanating from a sensor node) that a sensor node can relay is limited by a variety of factors such as channel conditions (including interference, attenuation, fading and ambient noise) and internal hardware and energy resources of the node. Assuming that the one-hop neighbors of a sink form the most significant communication relaying bottleneck, an analytical result for the fraction of sensor nodes that are unable to connect to their sink, i.e., the outage probability is presented.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my parents, Nagaraja Rao and Hema Rao for their support and encouragement during my studies. Without their support this would not be possible.

I would also like to thank my advisor Dr. George Kesidis for his guidance and support during the course of my graduate research.

I would also like to thank my committee members Dr. David J. Miller, Dr. Catherine M. Harmonosky and Dr. John J. Metzner for their feedback and suggestions that helped me to improve on my work.

Chapter 1

Introduction

An ad hoc network is a collection of autonomous and independent devices that can communicate with one another over a wireless channel. Ad hoc networks have become pervasive due to the availability of cheap wireless devices. They offer a convenient alternative to wired networks. Typical uses for ad hoc networks can be found in sensor networks, where devices may be rapidly deployed in a, possibly, hostile environment, disaster relief operation where existing infrastructure is no longer available, and general temporary networks where the short lifetime of a wired network does not justify the cost and time for setting it up. Due to the untethered nature of these devices, referred here on as *nodes*, they can be mobile and in communication as long as they are in radio range of each other. The current focus in this thesis is on sensor networks rather than mobile ad hoc networks (MANETs).

A sensor network is a collection of wireless devices, henceforth referred to as *sensor nodes*, that are deployed (possibly densely) in a geographic area to interact with or observe their environment. Typically, compared to an ad hoc network, a sensor network has a larger number of nodes that are more densely deployed. Sensor nodes are usually small, low-powered, inexpensive devices. The range of communication of a sensor node is typically limited compared to a node in an ad hoc network. As they are low-powered devices they cannot perform computationally intensive tasks. However it is not uncommon to have a fewer number of expensive sensor nodes deployed together with large number of inexpensive sensor nodes.

Some of the application areas of sensor networks are military, health, industrial sensing, and habitat and environmental monitoring [3] [17]. In all the above applications, data accumulation is the primary task of the sensor network. Sensor nodes are equipped with transducers to convert external stimuli to sensor data. Data generated by source sensor nodes, is forwarded by relay sensor nodes to sink sensor nodes. The source sensor nodes, relay sensor nodes and sink sensor nodes are henceforth simply referred as *sources*, *relays* and *sinks*, respectively.

Data dissemination methods vary in sensor networks, from the direct routing of data to the final destination to the publish-subscribe (or push-pull) methodology [34]. Three methods based on the location of the sensed data are considered.

In the first method, data gathered by sources is stored locally and queried through the sinks when required. Sinks query data by flooding the network [34] or query sources in return to the advertisement flooded by sources on availability of data [50]. This method is optimal if only a small fraction of sensed data is needed.

In the second method, data is stored at specific sensor nodes that act as local repositories. The data from co-located sources are aggregated, compressed and stored for future queries. Sinks query the local repositories for the required information. Unlike the first method where sinks can query co-located sensor nodes for redundant data, the local repository reduces querying for redundant data by aggregating and compressing data gathered from co-located sensor nodes. Data compression also helps in reduction of bandwidth required for communication with sinks. Locations of the local repositories can affect energy consumption of sensor network. Local repositories located closer to sources reduce the energy consumption due to data transfer from sources to local repositories, but increases the querying distance from sinks [63]. Local repositories can be uniquely assigned for specific data types. For example, data related to tank activity can be stored in one repository while data related to troop location in another.

In the first two methods, the sources and sinks discover one another either with advertisement or through flooding. Alternately the authors in [78] have proposed an index based search method. The sensed data is stored at the sources or in a local repository and the location information of the sources are stored in *index sensor nodes*. The queries are routed to the index nodes. This is particularly suited for a large sensor network with low data request.

In the third type, data sensed by the sources are forwarded to the nearest sink. This is ideal when all the data sensed by the sensor network is necessary for processing. The sensed data can be processed and located at sinks for future queries or forwarded to a central base station after some processing. This is the sensor network architecture considered in the current thesis. It is assumed that the sinks are more expensive hardware compared to relays or sources as they have the additional task of pre-processing and/or temporary storage of data. Figure 1.1 shows an example of sensor network.

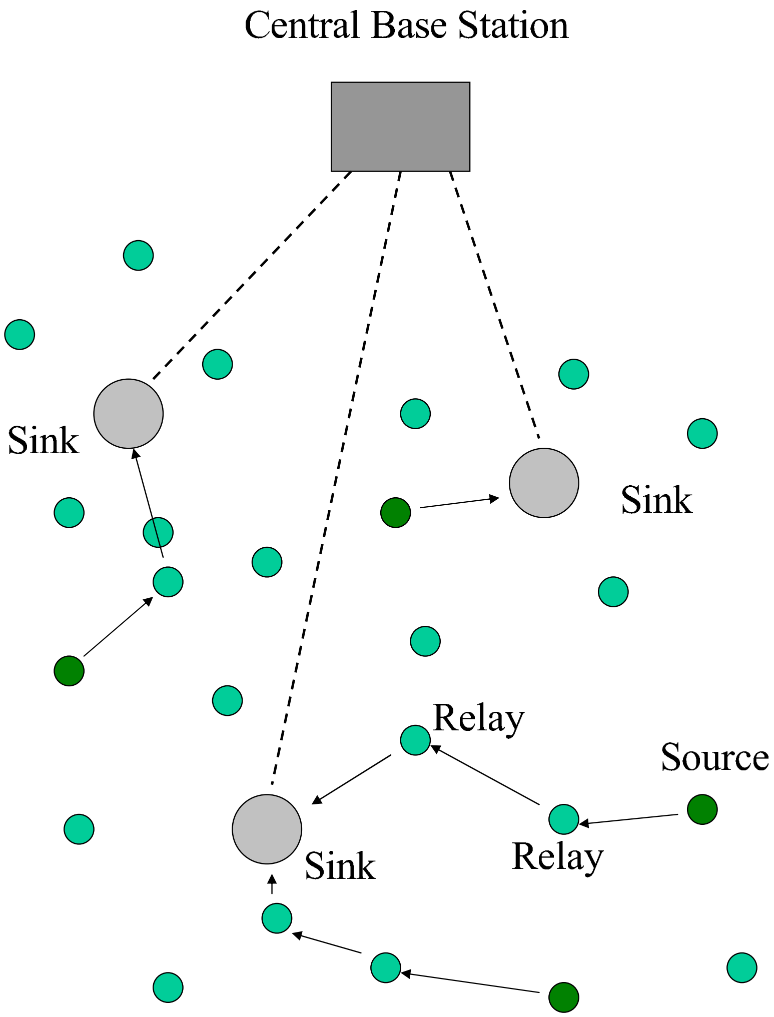


Figure 1.1. A typical sensor network

Data generated by sources can be separated into latency critical and latency non-critical data. Latency critical data, like real-time video feed are constrained by timely delivery to the central base station. Delay guarantees are limited by the amount of traffic a sensor network can handle. The relation between capacity and delay is examined in [29, 27, 8, 19].

The node capacity of a static ad hoc network, consisting of n nodes, increases as $O(\frac{1}{\sqrt{n}})$ [31]. This is applicable to a random source-destination pair of nodes in a static ad hoc network. As the number of nodes, n, increases in a static ad hoc network, the number of concurrent one-hop transmission increases as $O(n)$, assuming a constant node density and communication radius. If communication is between a random source-destination pair of nodes, the mean linear distance between them increases as $O(\sqrt{n})$: the spatial diameter of the sensor network. The capacity of the end-to-end communication link increases approximately as $O(n/\sqrt{n}) = O(\sqrt{n})$ and capacity of each node increases as $O(1/\sqrt{n})$ [47]. For a mobile ad hoc network, where the nodes move independently, and the maximum number of relays for a source-destination pair communication is limited to 1, the capacity increases as $O(1)$ [29]. The source forwards data to all its neighboring nodes and the data is relayed to the destination when one of the neighbors or the source moves in communication range of the destination. The delay experienced by a source-destination pair is large, that is, of the order of time taken for node contact by mobility.

A compromise between capacity and delay can be achieved by considering limited mobility such that there is acceptable delay bounds [19] or by considering geographic based routing algorithms in mobile ad hoc network that forward packets to nodes that are closer to the destination and in turn reduce delay [8].

Mobility, thus far, has been considered as an uncontrollable event, that the sensor network is designed to deal with. For example the sensor nodes in smart dust [39] do not have control over their motion. However, if mobility of sensor nodes can be controlled, mobility can be considered as a design parameter.

The current thesis examines methods to leverage “purposeful” mobility to improve the efficiency and performance of a sensor network. For example a mobile sensor node experiencing local deep fades and shadowing can move to a different location for better channel conditions. Another benefit from mobility can be reduced energy consumption for transmission. The transmission power required to transmit over a distance d is here on assumed to be given by Kd^{α} where K is a constant and $\alpha \geq 2$ is the transmission attenuation factor. Mobility can reduce transmission power by reducing the transmission distance d.

The cost of mobility is, typically, expensive compared to communication or computation cost. To justify mobility, there may need to be a significant resulting savings in communication cost over time. That is, the cost to move could be amortized against the savings in communication energy. Another motivation to move is significant improvement in channel conditions which reduces communication delays, network bottlenecks and improves the overall network condition.

Mobility might be required in some situations regardless of the energy cost. For example, consider a sensor network which has to perform surveillance over a given region. If the sensor node density is low (this can happen when sensor nodes die over time), the only way sensor nodes can cover the entire region would be by being mobile. Another example is when a sensor node tracking a target begins following it: the supporting relays that forward traffic from the target tracking sensor nodes may also need to move to ensure that information from target tracking sensor node is forwarded to the sinks.

Due to the distributed nature of sensor networks, the purposeful mobility algorithms are developed in the context of a distributed system. Sensor nodes use only local information to execute the algorithm thereby allowing the algorithm to scale to large networks. The algorithm is not computationally intensive which works well for sensor nodes with limited computation capacity.

In the current thesis, the Distributed Simulated Annealing algorithm based on the Simulated Annealing paradigm is presented. Relays implementing the DSA algorithm consider moves that reduce the total transmission power in its neighborhood. The DSA algorithm considers the transmission power of the relay and the neighboring sensor nodes transmitting to the relay when deciding on a move. An annealing based algorithm is considered to overcome local optima and find the global optima. Since there are mutliple sinks in the sensor network, an extension of DSA algorithm to include the *anycasting* routing paradigm is also considered [54, 52, 69, 74]. A unicast based extension to create an anycast routing algorithm that routes data from the sources to the “nearest” sink is considered.

Mobility affects the different layers of the network stack starting from the physical layer up to the application layer. For example, as discussed above, mobility can make or break communication links. Mobility can affect the quality of a link by moving sensor nodes out of shadow regions and/or change channel characteristics.

From a network layer perspective, mobility can affect the stability of a route. If a sensor node moves very often, then routes that depend on it tend to break. This leads to the routing algorithm frequently learning paths between sensor nodes. The routing algorithm can factor in mobility if a *mobility metric* is available, while forming routes, i.e., the routing algorithm can ignore nodes that have higher mobility metric. For purposeful mobility, the mobility metric can simply be the probability of motion in the next time slot and for uncontrolled mobility it can be an estimate of the probability of motion in the next time slot based on the previous N time slots.

The application layer can also affect or be affected by mobility. For example, surveillance application of sensor network would require mobility to monitor regions that are not covered by the initial deployment of sensor nodes and require mobility to visit the regions periodically. Target tracking requires mobility if the tracked target is mobile and moving out of sensing range of all sensor nodes.

Figure 1.2 summarizes the interaction of mobility with the different layers of networking stack.

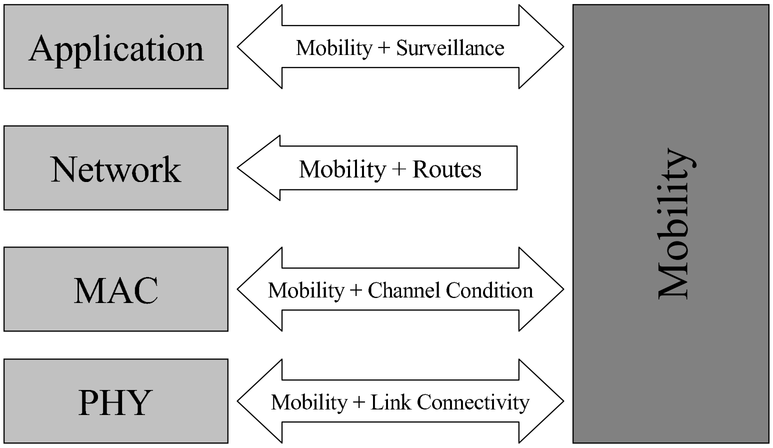


Figure 1.2. Influence of mobility on wireless stack

As discussed previously, mobility affects the capacity of a sensor network. Capacity of a mobile sensor network scales better compared to a static sensor network, but at the expense of delay. In this thesis capacity issues related to static sensor networks are studied. Unlike the network in [31], where the authors study the network capacity for random source-destination pair, this thesis considers a sensor network architecture where data from the sources are forwarded to the nearest sink by relays and from the sinks to a central base station. The capacity of a sensor network is quantified by the number of sources a sink can support. Since the relays forward the data from the sources to the sink, the relays closest to the sink forward maximum amount of data, thus creating a "funnel effect" near the sink. Now, the capacity of the sensor network is limited by:

- The forwarding capacity of a sensor node. Since the relays closest to the sink forward maximum amount of data, they are likely to reach saturation first and limit capacity increase.

- The sensor node density or communication radius. The number of sensor nodes contending for the channel, which influences the capacity of a sensor network, depends on the sensor node density and the communication radius of each sensor node.

- Capacity of the sink. Greater capacity of the sink means that a greater number of sources can be supported by it.

- Capacity of the link connecting the sink to the central base station.

The outage probability, i.e., fraction of sensor nodes unable to communicate with the sink, is presented and its relation to sensor node density, sink density and communication radius of sensor nodes is studied. The detail are presented in chapter 7.

1.1 Thesis Outline

The remainder of this thesis is divided as follows. Chapter 2 is the literature survey. Chapter 3 presents the tactical sensor network considered for the thesis and the theoretical framework for purposeful mobility. The affect of mobility strategies on surveillance in a tactical sensor networks is studied in chapter 4. The simulation setup and results for purposeful mobility are presented in chapter 5. Chapter 6 presents the extension of the DSA algorithm for the case of multiple sinks. In chapter 7, an analytical framework to measure “outages” due to relay-capacity constraints are developed. The simulation setup and results for measuring outages are presented in chapter 8. Finally, the conclusions and future work are discussed in 9.

Chapter 2

Literature Survey

This chapter reviews some of the previous work in sensor networks related to mobility and the interplay between mobility capacity and delay.

Sensor network are, typically, deployed for observation and/or interacting with their environment [3, 17, 4]. A typical sensor network consist of sensing nodes that collect sensing data and either store it temporarily or forward it to a central sink. Data is forwarded to the sink through sensor nodes that act as relays.

Data is temporarily stored in sinks or pre-processed and forwarded to central base station [3]. Alternately data can be stored at sinks and read from the sinks by mobile agents that periodically visit each sink [36]. An example of such a network can be a sensor network deployed in a remote region and an aerial vehicle flying over the region and collecting data from the sinks. The mobile agent can also be a person periodically visiting the sinks or even sensor node to collect data. In [36] the authors consider an automated mobile agent and develop algorithms to optimize the path taken by the mobile agent with respect to delay in gathering data (similar to the traveling salesman problem). The advantage of this architecture is that energy expenditure on mobility is restricted to few mobile agents, possibly even one mobile agent. Also energy reserves of mobile agents can be easily replenished since mobile agents collect data and return to "base". But due to the time scale involved, such an architecture is not suitable for real-time latency critical data.

Applications like surveillance, infrastructure monitoring and health care require real-time data transfer. Data can be forwarded, for e.g., in a battlefield from sinks to a central base station through a satellite link [3] or in case of video surveillance through wired connection to a central security office [14]. Note, a sensor network can carry both latency critical and latency non-critical data at the same time [77, 15]. To increase the lifetime of the sensor network, routing algorithm try to find optimal routes in term of energy and delay constraints [21]. Latency critical traffic are routed on least delay routes, while latency non-critical traffic are routed on energy rich routes. The authors in [37] use pheromones to build multipath routes with acceptable delay and energy constraints. The current thesis considers the Distributed Bellman Ford with transmission power as route metric.

2.1 Mobility, Connectivity and Coverage in Sensor MANETs

The current thesis focuses on tactical sensor networks deployed for surveillance and target tracking. Issues of surveillance, coverage and communication connectivity for random static sensors are explored in [30] and [62]. In [30] the authors look at the minimum transmission power required to create a connected network (i.e. no isolated node) when nodes are deployed randomly. The authors have derived results for the minimum power needed to keep the network connected. In [62], the nodes are active with certain probability with a given sensing and communication radius. The authors derive results to prove connectivity and coverage is maintained even if the probability of a node being active and communication radius are low. This may give the communication radius necessary for coverage and connectivity but only for a static network. It is not possible to extend this framework to include mobility as the nodes are assumed to be uniformly arranged. In [30] and [62] the authors do not consider mobility for coverage and connectivity. Mobility can be used to assist the sensor nodes during deployment to maintain connectivity and coverage. Mobility for deployment is explored in [72], [79] and [33]. In [72], the authors consider a scenario where mobility assists the sensor nodes in deployment. The goal is to maximize coverage while not compromising connectivity by intuitively enacting local repulsion of nodes (to minimize redundancy of coverage) along with long-range attraction of nodes so as not to compromise network connectivity. Authors in [79] and [33] also considered virtual forces between the nodes for sensor deployment.

Again, the mobility assists only in the deployment of a static network. A variant to the deployment of static network is considered in [73], where the authors look at sensor node redeployment. Here the mobility algorithm is executed whenever the network topology changes significantly due to sensor node loss from battery failure or external events. The mobility algorithm "reconfigures" the topology to find optimal position for coverage and connectivity.

Clearly, a basic motivation of node mobility is that the nodes are not deployed with sufficient density in the region under surveillance to make mobility of the nodes unnecessary. Such a situation can arise when a large population of sensor nodes die over time, reducing the density of deployment and thus making it necessary to move. Here, unlike [73], its is necessary to consider continuous mobility for periodic monitoring of all regions.

In [41], the authors evaluate the distribution of the time until detection of a point-target under purely random (diffusion) mobility per node. Given an associated Bessel process describing the distance between any two given nodes, there also exist expressions for the distribution of the time between successive contact of any two nodes assuming each of their communication ranges is bounded, see p. 297 of [11]. Dynamic surveillance with mobility is also explored in [45] where sensor nodes are continually moving. Similar to [41] the authors discuss and derive results for distribution of the detection time of a randomly placed target. The authors in [45] also consider a game theoretic approach to study best and worst case scenario for target detection time. In [61] the authors look at limited mobility to assist in coverage improvement. Here the mobility is motivated by dead nodes that create coverage holes and are filled by limited mobility of neighboring nodes with minimum energy expenditure for mobility.

A contemporary attempt at applying controlled mobility for energy conservation due to communication is considered in [28]. Here the authors look at moving sensor nodes to reduce communication distance and hence communication energy. Similar to the work in this thesis, the authors look at a distributed algorithm where sensor nodes move with only local information, i.e., information from their immediate neighbors. However, unlike the current thesis, the authors do not consider the role of routing. The authors consider a location based routing algorithm and execute the controlled mobility algorithm. Though the mobility algorithm gives an optimal position of nodes for that route, it does not consider moves that could create alternate routes which offers a lesser communication energy expenditure than the current route. Unlike the DSA algorithm they do not consider the tradeoff of expenditure due to mobility against savings for communication. Also they look at variants of their mobility algorithm for different traffic patterns, for e.g. sink unicast flow, multiple unicast flows and *concast* flows (flows from multiple sources to a sink).

In [49], the effect of mobility on a position detection algorithm was considered. The authors suggest the use of hop-counts from reference nodes to find relative positions of all nodes. Reference nodes are moved to neighborhoods where accurate information is not available. Here mobility helps to increase accuracy of information regarding node positions but does not help to find best position for coverage or connectivity.

Flocking properties of platoons of UAVs were studied in [67, 35]. Specifically, they explore local mobility laws that keep formation (velocity and heading).

In [48], the authors look at navigating a vehicle across a sensor grid. The sensor nodes interact with the vehicle giving it local information about the terrain and helping it decide where to move next. Similarly in [60] the authors look an interaction of static and mobile sensor nodes to achieve a common goal. Static nodes assist in location determination while the mobile nodes with controlled mobility fill gaps in communication. The goal of mobile nodes is to assist in communication, and not to find optimal position for energy efficient communication.

An interesting idea of separating mobility from sensing is discussed in [44]. Here the authors look at sensor nodes attaching on to mobile agents that transport the sensor nodes. In such a system, practically all sensor nodes can be moved and the impact of mobility cost can be significantly reduced, since the mobile agent can be built specifically for mobility with enhanced and even renewable energy source.

In [40] the authors discuss applications of controlled mobility to improve a number of issues, including topology adaptivity, capacity, energy capacity and data fidelity.

Finally, simulated annealing mechanisms have been proposed in the past for other networking purposes. For example, in [70], a (centralized) simulated annealing algorithm is used for clusterhead selection based on weights assigned to nodes.

2.2 Capacity in Sensor MANETs

Kumar and Gupta [31] show that the capacity of static ad hoc network scales as $O(\frac{1}{\sqrt{n}})$ where n is the number of nodes. This capacity is computed for communication between random pairs of nodes. The current thesis, however, considers a sensor network wherein sensor nodes communicate only with their local sink.

In [38], the authors consider the funnel effect in a mesh network and its impact on capacity. They simplify the network to a single dimensional chain of nodes and consider interference caused by member nodes of the chain. Instead the current thesis considers a two dimensional distribution of sensor nodes. Li et al. [47] consider the capacity of a wireless ad hoc network and the impact of IEEE 802.11 MAC layer on it. The authors study the dependence of capacity on the average communication distance, and conclude that as it increases, the capacity of the network decreases. One way to decrease the communication distance is to break up the network into clusters where communication is limited to the cluster-head. Liu et al. [20] compare the performance of a flat network with a hierarchical network and prove that a hierarchical network scales better. There have been attempts to improve the capacity of an ad hoc network by modifying the MAC or PHY layers. In [68], the authors analyze and suggest changes to the MAC layer to increase capacity of WLAN networks. In [1], modifications to the MAC layer are proposed to increase number of simultaneous transmissions in a WLAN network. Similar modifications are discussed in [46, 75, 7].

The affect of mobility on capacity is studied in [29]. Unlike static ad hoc networks, capacity in mobile ad hoc network scales as $O(1)$. Capacity increase is at the cost of transmission delay, that is, in the worst case, on the order of the time required for sensor node contact by mobility. The authors in [19] consider limited mobility for a static network to increase capacity with acceptable delay bounds. Authors in [8] study application of a routing algorithm for mobile sensor nodes to achieve capacity increases with acceptable delay bounds. The authors in [27] examine delay capacity relation in [31] and [29] and propose scheme to achieve optimal order of delays for given mobility with changes to number of hops for communication, transmission range and degree of mobility of a sensor node.

In [26, 6], the authors consider hierarchical networks of wireless nodes and have developed models for the same. The authors consider each hierarchy of nodes as a spatial Poisson deployment connecting to a single node (sink) and the sinks in turn forming a spatial Poisson process that connect to a base station.

The sensor node and sink distribution are modeled as a two-dimensional spatial Poisson process [26, 65]. The node density of sensor nodes is assumed to be much greater than that of sinks. In [5], the authors study a “radial spanning tree” by which all nodes communicate to a single sink at the origin. Asymptotic limits of various performance metrics of the radial spanning tree are derived. The authors do not, however, consider constraints on the number of flows a sensor node can support/relay or on the communication range of a sensor node.

Chapter 3

Purposeful mobility

This chapter details the sensor network architecture and applications considered for the thesis. Also the network model and mathematical formulation for the purposeful mobility algorithm are presented.

3.1 Purposeful mobility in sensor networks

The current thesis examines the application of purposeful mobility to tactical sensor networks. Sensor networks are deployed randomly over a geographic area and typically left unattended. Due to this sensor network algorithms and protocols are robust to change and possess self-organizing capabilities. The current thesis looks at the application of tactical sensor network in a military scenario. More specifically, a tactical sensor network that is deployed behind enemy lines to monitor enemy activities is considered.

The sensor network considered consists of fewer number of expensive, mobile, GPS-enabled sensor nodes with larger number of inexpensive, passive, immobile sensor node deployed in the battlefield. To motivate mobility, it is assumed that the inexpensive sensor nodes have died over time (due to limited battery capacity) and the remaining mobile sensor nodes are performing the given tasks.

The goal of a tactical sensor network is to monitor military activities on the field and track enemy targets. Tactical sensor networks must have a long lifetime as there cannot be any maintenance for sensor nodes deployed behind enemy lines and frequent redeployment might not be possible due to risk factors.

Purposeful mobility can be used for deployment, surveillance, target tracking and transmission power conservation.

3.1.1 Objectives of tactical sensor network

The objective of a tactical sensor network is to collect and transfer information from the battlefield to a command-and-control center. The command-and-control center can process the information and further instruct the sensor network on future actions.

The tasks performed by a tactical sensor network are:

1. Surveillance: Performing surveillance of the local environment to scan for enemy activities and transmit this information to a command-and-control center. The surveillance task generates low volume data that is low priority and non-latency critical.

2. Target Tracking: Actively track targets that have been identified as hostile and send real-time information to the command-and-control center. The tracking task generates high volume real-time data (e.g. a video feed) that is of high priority and latency critical.

3. Logistical Support: Sources must be able to convey the information they gather from their environment to a command-and-control center. Some of the sensor nodes act as relays to forward information from sources to the command-and-control center.

3.1.2 Surveillance task

Sensor network deployment in unfriendly regions could lead to methods for deployment where there is no control over the final placement of sensor nodes. The initial deployment could be suboptimal for the surveillance task. There could be unequal sensor node density through the field, leaving some areas better monitored than others. Limited purposeful mobility can move sensor nodes to better positions for improved performance of surveillance task. If the number of nodes are sufficient then no more mobility is required for surveillance. However it is possible that the coverage area of all the sensor nodes is less than the area that needs to be observed. This could happen over time as sensor nodes die out. In such situation, mobility is the only way to ensure that the entire area is under observation periodically.

3.1.3 Tracking task

A sensor node assigned the task of tracking a target monitors the activities of the target and sends this information to the command-and-control center in real-time. Sensor nodes involved in tracking tasks can be static or mobile. A target is usually tracked by multiple sources in the same neighborhood to increase accuracy.

Sensor nodes performing tracking tasks are equipped with special sensors to monitor target activities. Note, mobility for tracking task is not considered in this thesis. It is assumed that sensor nodes perform tracking and generate information that needs to be transferred to the command-and-control center.

3.1.4 Relaying task

Data is transferred from the sources to the sink by relays. A routing algorithm (for example distributed Bellman Ford [18]) determines the routes from the sources to the sink. The relays forward data to the next relay in the route to the sink.

There is a communication cost to forward data from a source to a sink. It is possible to reduce the cost if the communication distance between the relay nodes are reduced. Purposeful mobility is used to move relay nodes such that communication distances are reduced.

The formulation of a cost function to keep track of communication and mobility cost and application of a “greedy” and distributed annealing algorithm is explained in sections 3.3 and 3.4.

For the tactical sensor network considered, it is assumed that the volume of data generated by surveillance tasks is negligible compared to the volume generated by tracking tasks. Further the data from tracking task is of higher priority compared to the data from surveillance, as the target is tracked in real-time.

3.2 Mobility for relay networking

In this section it is assumed that the relays can perform only one task, i.e. just relaying. For example, in response to target detection, certain sensor nodes (proximal to the target) are assigned logistical target-tracking tasks while others are assigned relaying tasks. The focus herein is on mobility for the latter category of sensor nodes and, for those sensor nodes, the goal is *mobility* to maximize the lifetime of the sensor network and, at the same time, perform the required relaying tasks satisfactorily. Specifically, the objective is to incrementally find the node positions that minimize the total required transmission power for all the active flows in the sensor network while suitably "penalizing" for the energy cost of motion in order to find these positions.

For a sensor node to move from one position to another, there must be a significant resulting reduction in communication power compared to the power consumed for motion. This would vary significantly depending on the environment in which the moving vehicle operates. For example, the relative cost for UAVs (Unmanned Air Vehicles) will be significantly less than UUVs (Unmanned Underwater Vehicles) which will be significantly less that terrestrial vehicles. In the following, it is assumed that sensor nodes make such decisions based only on local information (traffic and neighbor positions) as appropriate for a highly decentralized and distributed sensor network. A distributed mobility strategy based on the simulated annealing algorithm is devised, see, e.g., [42] [25]. The randomness introduced in the strategy allows the sensor network to avoid positions that are suboptimal local minima of its objective.

3.3 Basic network model assumptions and a "greedy" mobility strategy

Definition of the following terms is necessary for the problem formulation:

- N is the number of intermediate relays;
- F is the number of flows each of constant rate λ packets/s (fixed length packets assumed herein);

- X is a vector of the positions of the intermediate relay nodes (so, in 3-dimensions, X is actually an $N \times 3$ matrix);

- r is the set of F routes (each assigned to one flow) where a route through the network is determined by a series of sensor nodes beginning with a source and ending with a sink;

- $V(X,r)$ is the total power required from the network to transmit the F flows using routes r when the relays are in positions X; the optimal choice of routes at position X is

  $$R(X) \equiv \arg \min_{r \in \mathcal{R}(X)} V(X,r)$$

  where $\mathcal{R}(X)$ is the set of feasible routes connecting those nodes when in positions X. Note, it is assumed that sources and sinks are in communication distance of each other, i.e. a source can transmit directly to sink by increasing its transmission energy. However the routing algorithm selects multihop routes due to the reduced commuication cost.

The quantity $R(X)$ is the objective of a distributed routing algorithm (like Bellman-Ford distance vector approach [18]) and its determination is assumed to occur on a much faster time-scale than that of the mobility of the nodes. Further assume that all nodes have an associated clock cycling every T seconds (clocks are not necessarily synchronized). Once every cycle, a node decides with probability p whether it should attempt to move. Under a deterministic "greedy" mobility strategy, node k at position x_k will move to position z that minimizes

$$\begin{aligned} & V((x_{-k};z), R(X)) - V(X, R(X)) + c|z - x_k|/T \\ \equiv \ & \Delta_k V(x,z) + c|z - x_k|/T \end{aligned} \tag{3.1}$$

where $(x_{-k};z)$ represents the vector X with x_k replaced by z, and $c|z - x_k|$ represents the amount of energy required for the move that has been amortized over a clock cycle-time (c is a fixed parameter of the assumed "constant" terrain). In equation (3.1), $V(x_{-k}; z, R(X))$ is the transmission power consumption at the "new" position and $V(X, R(X))$ is the transmission power consumption at the "old" position. The new position is favourable if $V(X, R(X)) - V(x_{-k}; z, R(X)) > 0$, i.e., transmission power consumption is reduced at the new position. The objective of the move is to maximize the reduction. The move must also minimize the cost of motion, i.e. minimize $c|z - x_k|/T$. Combining the two, i.e., maximize $V(X, R(X)) - V(x_{-k}; z, R(X))$ and minimize $c|z - x_k|/T$, results in minimize $V(x_{-k}; z, R(X)) - V(X, R(X)) + c|z - x_k|/T$. Note the movement according to equation (3.1) may be velocity v constrained, i.e.,

$$|z - x_k| \leq vT.$$

For a simple illustrative example, consider a relay k that forwards two flows from its tributary nodes i and j to node l as shown in Figure 3.1. The power required to transmit over distance d (again, at rate λ packets/s) is given by Kd^α Watts where K is a constant and $\alpha \geq 2$ is a transmission attenuation factor [29] [31] [58]. So, for this example,

$$\begin{aligned} \Delta_k V(x, z) = K[&|z - x_i|^\alpha + |z - x_j|^\alpha + 2|z - x_l|^\alpha \\ &-|x_k - x_i|^\alpha - |x_k - x_j|^\alpha - 2|x_k - x_l|^\alpha]. \end{aligned} \tag{3.2}$$

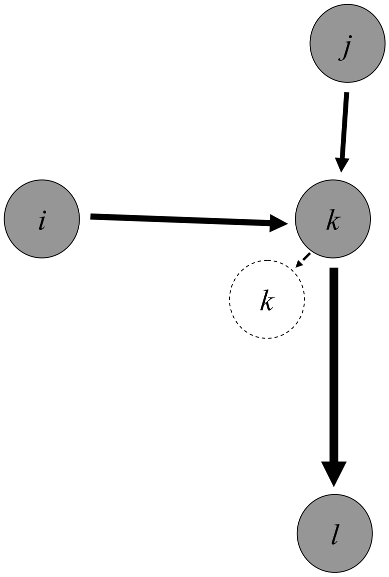


Figure 3.1. Greedy move to reduce power consumption

A basic assumption herein is that the quantity in (3.1) is computable by sensor node k requiring, in particular, knowledge of the location of its neighbors [51]. Note that what makes this "distributed" computation of $\Delta_k V(x,z)$ is the fact that V is an *additive* function of the transmission power required at each sensor node. The existing routes (at X) are used in the term $V((x_{-k};z),R(X))$ because, in this distributed setting, the sensor node k does not know the consequences its move will have on the routes. The uncertainty is due to simultaneous motion of neighboring relays. A relay decides to move based on the assumption that its neighbors are static. When multiple sensor nodes can move simultaneously, the uncertainty in the benefit of a move is significantly larger; to reduce the likelihood of this, one may set $p = 1/N$ for this case of static sources and sinks.

3.3.1 Jointly optimizing over routes and position

The greedy algorithm optimizes over existing routes, i.e., the relays move to reduce transmission power of current routes. In this sense, the greedy algorithm converges to a "suboptimal" solution because it ignores moves that can alter routes to lower total transmission power, i.e.,

$$\begin{aligned} \min_z \Delta_k V(x,z) + c|z - x_k|/T \ &\geq \\ \min_z V((x_{-k};z), R((x_{-k};z))) \\ - V(X,R(X)) + c|z - x_k|/T. \end{aligned} \tag{3.3}$$

Equation (3.3) states that the greedy move with current routes can potentially be worse than a move *jointly* optimizing position and routes.

For example, consider two target positions z_1 and z_2 for relay k. Assuming relay k is equi-distant from z_1 and z_2, it will move to the position that minimizes $V(x_k; z, R(X)) - V(X, R(X))$. Assume z_1 is the better choice with current routes, i.e.,

$$V(x_{-k}; z_1, R(X)) - V(X,R(X)) < V(x_{-k}; z_2, R(X)) - V(X,R(X)).$$

Note that relay k assumes current routes at new position, but the routing algorithm can find better routes. Also it is possible for z_2 to be a better position with updated routes than z_1, i.e.,

$$V(x_{-k}; z_1, R(x_{-k}; z_1)) - V(X, R(X)) > V(x_{-k}; z_2, R(x_{-k}; z_2)) - V(X, R(X)).$$

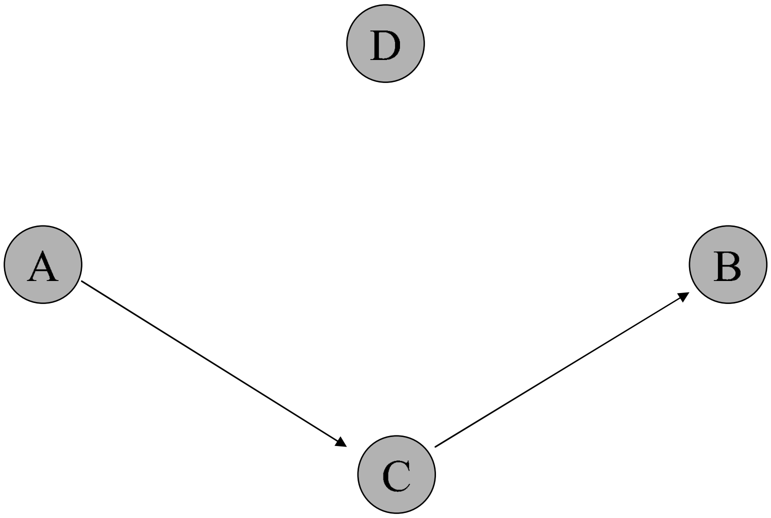
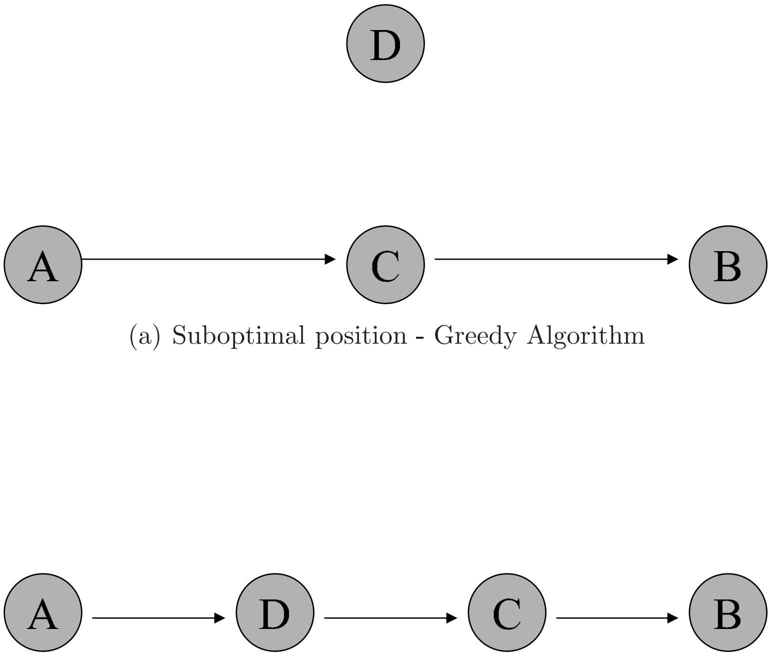


Figure 3.2. Sensor nodes after routing algorithm

An example of a "suboptimal" convergence of the greedy algorithm is explained next. Consider the network shown in figure 3.2. The routing algorithm selects a path from source node A to destination node B through, say, node C. For the greedy algorithm implementation, the optimal position achieved will be as shown in figure 3.3(a). But the optimal position for network is as shown in figure 3.3(b). Since the greedy algorithm optimizes on the current route, it fails to explore moves that could (predictably) alter routes and find better positions for the relays.

To overcome the limitations of greedy algorithm, a joint optimization over position X and routes $R(X)$ is considered. Unlike the greedy algorithm that chooses the next position to minimize $V(Y, R(X)) - V(X, R(X)) + c||Y - X||/T$, the joint optimization chooses the next position to minimize $V(Y, R(Y)) - V(X, R(X)) + c||Y - X||/T$. Note, it is assumed that the relay has apriori knowledge of the routes $R(Y)$ at new position Y.

The motion is restricted to a lattice grid and a relay k (currently at x_k) selects the optimal neighboring position from one of the eight neighboring grid points, assuming the neighboring lattice points are unoccupied. A relay chooses the neighboring lattice point that minimizes $V((x_{-k}; z), R(x_{-k}; z)) - V(X, R(X)) + c||z -$



(a) Suboptimal position - Greedy Algorithm

(b) Optimal position

Figure 3.3. Sensor node position after move

$x_k||/T$ where $(x_{-k}; z)$ is the position vector X where the x_k is replaced by z. Note this is also a greedy algorithm that jointly optimizes over both position X and routes $R(X)$. Now, consider a global cost function given by equation (3.4)

$$\begin{aligned} U(X) &\equiv V(X, R(X)) + cT^{-1} \sum_{k=1}^{N} |X_k| \\ &\equiv V(X, R(X)) + cT^{-1} ||X||. \end{aligned} \tag{3.4}$$

where $U(X)$ is the cost function that represents the total transmission power with the sensor nodes at position X and the amortized cost of mobility from origin to the position X over a constant terrain.

The greedy algorithm attempts to minimizes $U(X)$, but $U(X)$ is not a unimodal function. That is, the greedy algorithm jointly optimizing routes and position may converge to a local minima. Therefore consider a randomized search based algorithm on the simulated annealing paradigm. The cost function for the simulated annealing framework is given by equation (3.4). Again, the motion of relays are restricted to lattice points, i.e., $X \in D$ where D is the lattice. The neighbor selection probability is given by Q_{xy}, i.e., the probability of selecting position y as the next position from current position x. Note $Q_{xy} > 0$ implies x and y are neighboring lattice points. The discrete-time Markov chain represented by transition-probability matrix (TPM) Q_{xy} is assumed time-reversible with stationary distribution given by μ, i.e., the detailed balance holds: $\mu_x Q_{x,y} = \mu_y Q_{y,x}$. P_{xy} is the heat bath acceptance probability given by

$$P_{x,y} \equiv Q_{x,y} \min\{1, \exp(-\beta(U(Y) - U(X)))\}$$

where $\beta > 0$ is interpreted as inverse temperature and a constant. The TPM P inherits aperiodicity and irreducibility from Q. The TPM P is also time-reversible with *Gibbs* stationary distribution:

$$\pi_x \quad = \quad \frac{\mu_x \mathrm{e}^{-\beta U(X)}}{Z_\beta}$$

where Z_β is the normalizing constant (partition function). Detailed balance: $\pi_x P_{xy} = \pi_y P_{yx}$ holds true and is proved in Appendix A.1.2.

Since π_x is Gibbs

$$\lim_{\beta\to\infty} \pi_x = \begin{cases} \frac{1}{|\Omega|} & \forall x \in \Omega \\ 0 & \forall x \notin \Omega \end{cases} \tag{3.5}$$

where the global minimizing set

$$\Omega \equiv \arg\min_{x\in D} U(x)$$

Proof of equation (3.5) is presented in Appendix A.1.2

Now define a TPM for the annealing process:

$$\begin{aligned} \bar{P}_{x,y} \quad \equiv \quad & Q_{x,y} \min\{1, \\ & \exp(-\beta[V(Y, R(Y)) - V(X, R(X)) + c||Y - X||/T])\} \end{aligned} \tag{3.6}$$

The TPM $\bar{P}$ continues to be irreducible and aperiodic, thereby yielding a unique stationary distribution $\bar{\pi}$ [23], but is no longer time-reversible. It can be shown that the simulated annealing chain $\bar{P}$ also has a stationary distribution with a

Gibbs-like property (eq. 3.5).

Lemma 1. *$P_{x,y} \geq \bar{P}_{xy}$ for all $x \neq y$, and $P_{x,x} \leq \bar{P}_{x,x}$ for all x.*

Proof: By the triangle inequality,

$$\begin{aligned} & V(Y, R(Y)) - V(X, R(X)) + c\|Y - X\|/T \\ \geq \;& V(Y, R(Y)) + c\|Y\|/T - [V(X, R(X)) + c\|X\|/T] \\ = \;& U(Y) - U(X). \end{aligned}$$

The first statement of the lemma directly follows from the definitions of P and $\bar{P}_{xy}$. The second statement is an immediate corollary of the first because P and $\bar{P}_{xy}$ are row-stochastic matrices.

□

From Lemma 1 and theorem 1 in section 3.4 it is shown that the simulated annealing algorithm given by TPM $\bar{P}$ converges to the global minima of $U(x)$.

The heat bath acceptance probability given by equation 3.6 assumes knowledge of routes $R(Y)$ at new position Y. However, predicting the impact of mobility on routes is a non-trivial task. It might be possible to implement the joint optimization as a centralized algorithm with global knowledge of the sensor network, but a distributed implementation is not possible since the impact of mobility on routes cannot be precisely ascertained locally. Since the sensor network requires a distributed implementation, a modification to the simulated annealing algorithm is considered.

3.4 Mobility by distributed simulated annealing

Since a distributed implementation of the joint optimization is not possible, a simulated annealing framework where existing routes are assumed for mobility is considered. More precisely a kind of Distributed Simulated Annealing (DSA) algorithm is considered where the routes are assumed constant for the move. After each move, the routing algorithm is executed to check for better routes. The DSA and routing algorithms attempt to replicate the joint optimization by alternating between the mobility and the routing algorithms.

Similar to the simulated annealing algorithm in section 3.3.1, the DSA algorithm also considers motion on a lattice. Also, the neighbor selection probability is given by Q_{xy}. Now the TPM for the distributed annealing process is given by equation (3.7)

$$\begin{aligned}\hat{P}_{x,y} \;\; &\equiv \;\; Q_{x,y}\min\{1, \\ &\quad\; \exp(-\beta[V(Y,R(X)) - V(X,R(X)) + c||Y-X||/T])\}\end{aligned} \tag{3.7}$$

$\hat{P}_{xy}$ is irreducible and aperiodic, with stationary distribution given by $\hat{\pi}_x$, but not time-reversible. It is shown below that the stationary distribution $\hat{\pi}_x$, similar to $\bar{\pi}_x$, has a Gibbs-like property (eq. 3.5).

Lemma 2. *$P_{x,y} \geq \hat{P}_{x,y}$ for all $x \neq y$, and $P_{x,x} \leq \hat{P}_{x,x}$ for all x.*

Proof: By the definition of R and the triangle inequality,

$$\begin{aligned}&V(Y,R(X)) - V(X,R(X)) + c||Y-X||/T \\ &\quad\geq \;\; V(Y,R(Y)) + c||Y||/T - [V(X,R(X)) + c||X||/T] \\ &\quad= \;\; U(Y) - U(X).\end{aligned}$$

The first statement of the lemma directly follows from the definitions of P and $\hat{P}$. The second statement is an immediate corollary of the first because both P and $\hat{P}$ are row-stochastic matrices. □

Theorem 1. *If Ω is a singleton set, i.e., $\Omega \equiv \{x^*\}$, then*

$$\lim_{\beta\to\infty} \hat{\pi}(\beta) \;\; = \;\; \mathbf{1}^{x^*}$$

where $\mathbf{1}^{x^*}_y = 0$ if $y \neq x^*$ and $\mathbf{1}^{x^*}_{x^*} = 1$.

Proof: First note that

$$[(\mathbf{1}^{x^*})'\hat{P}(\beta)]_y \;\; = \;\; \hat{P}_{x^*,y}(\beta).$$

If $y \neq x^*$ then by Lemma 2,

$$\hat{P}_{x^*,y}(\beta) \;\; \leq \;\; P_{x^*,y}(\beta) \to 0 \;\text{ as } \beta \to \infty.$$

Otherwise, if $y = x^*$ then again by Lemma 2,

$$\hat{P}_{x^*,x^*}(\beta) \quad \geq \quad P_{x^*,x^*}(\beta) \ \to 1 \ \text{ as } \beta \to \infty.$$

Therefore,

$$\lim_{\beta\to\infty} (\mathbf{1}^{x^*})'\hat{P}(\beta) \quad = \quad (\mathbf{1}^{x^*})'. \tag{3.8}$$

Now recall that $\hat{\pi}(\beta)$ is the unique solution to

$$\begin{aligned} \hat{\pi}(\beta)'(I - \hat{P}(\beta)) \quad &= \quad 0 \\ \text{and} \ \ \hat{\pi}(\beta)'\mathbf{1}^D \quad &= \quad 1 \end{aligned}$$

where $\mathbf{1}^D$ is a vector all of whose entries are 1. Let $\hat{V}(\beta)$ be the matrix obtained by replacing a column of $I - \hat{P}(\beta)$, say column n, by $\mathbf{1}^D$. Thus $\hat{\pi}(\beta)$ is the unique solution to

$$\hat{\pi}(\beta)'\hat{V}(\beta) \quad = \quad (\mathbf{1}^n)'.$$

Uniqueness implies that the null space of $\hat{V}(\beta)$ must be just the zero vector; this, in turn, implies that $\hat{V}(\beta)$ is nonsingular giving, for all $\beta > 0$:

$$\hat{\pi}(\beta)' \quad = \quad (\mathbf{1}^n)'[\hat{V}(\beta)]^{-1}.$$

Beginning with (3.8) and using the same argument,

$$\lim_{\beta\to\infty} (\mathbf{1}^n)'[\hat{V}(\beta)]^{-1} \quad = \quad (\mathbf{1}^{x^*})'.$$

The theorem statement follows from the last two equations.

□

A straight-forward extension to this theorem (using condition (3.5)) follows for optimal set Ω in which no two states are directly connected by the TPM Q.

In summary, its has been shown that randomized annealing motion of the sensor nodes, distributed in the sense that only local information is used, nevertheless retains a Gibbs-like property, i.e., a natural composite utility of communication and mobility costs (3.4) is minimized as the temperature β^{-1} cools.

Also applying Lemma 1 in theorem 1, it can be shown that the stationary distribution, $\bar{\pi}(\beta)$, has a Gibbs-like property and the “joint” simulated annealing algorithm converges to the global minima for $U(X) = V(X, R(X)) + c|X|/T$, i.e.

$$\lim_{\beta \to \infty} \bar{\pi}(\beta) \quad = \quad \mathbf{1}^{x^*} \tag{3.9}$$

where $\mathbf{1}_y^{x^*} = 0$ if $y \neq x^*$ and $\mathbf{1}_{x^*}^{x^*} = 1$. This shows that the alternating execution of the DSA and routing algorithms converges to the same global optima of $U(X)$ as the joint optimization.

3.4.1 Heuristics in DSA

Although the simulated annealing algorithm achieves global optima, it requires infinite time to reach it. The homogeneous Markov model require infinite time to achieve stationary distribution. Also, achieving global optima requires generation of descending values of temperature of infinitely long homogeneous Markov chains. This is clearly impractical. Inhomogeneous Markov models show logrithmic cooling schedule achieve global optima [71, 22], but still requires infinite time to reach it. A discussion on homogeneous and inhomogeneous modeling of simulated annealing is presented in the appendix A.

Since the time taken for convergence with logarithmic cooling schedule is infinite and thus impractical, heuristics to achieve the best possible result in a limited number of iterations $N < \infty$ is considered. Some of the techniques proposed include random restart and tracking the best achieved value.

Due to the distributed nature of the algorithm and the lack of synchronization among the sensor nodes, random restart and tracking best global transmission power is not practical. Random restart requires redeploying the sensor nodes to random positions. Implementation of such a system in a distributed fashion is non-trivial. Even if a distributed algorithm that generates a new position for sensor node redeployment were developed, the energy consumed in moving to the new position could be large.

Alternately keeping track of the best value is also impractical. As the algorithm is distributed, the notion of a best *global* values is meaningless. This would entail a global entity keeping track of the total energy consumption which ignores the design constraints of a distributed algorithm. Even if the the global state could be maintained and the global minima positions tracked, again, moving to such a position could be expensive.

For the sensor network, consider the following method. The temperature is kept constant at a suitable value to allow sensor nodes to "explore" with "bad moves" for better optima. If the lifetime of the sensor network is sufficiently high, in terms of months, cooling the temperature over time could be considered. But a more aggressive cooling schedule would be impractical to maintain, since the sensor nodes move independently of each other and do not coordinate their temperature values. Instead, consider a batch-based cooling schedule where the temperature is kept constant for M iterations and decreased by a constant value δT.

3.4.2 Link breakage during moves

Note the DSA does not consider link breakage during moves. For the sensor network considered in the thesis it is assumed that any two sensor nodes can communicate with one another, i.e. the communication radius of a sensor node is greater than the geometric diameter of the sensor network. Thus a source can directly communicate with a sink, but the routing algorithm selects multi-hop routes due to lower transmission power consumption. However, if there were restrictions on communication range, the transmission power needed by sensor nodes to communicate with out-of-range sensor nodes is quantified as infinite. Therefore transmission power, T_d for distance d is defined by

$$T_d = \begin{cases} Kd^{\alpha} & for \;\; d \leq R_T \\ \infty & for \;\; d > R_T \end{cases} \tag{3.10}$$

where R_T is the communication radius of a sensor node. The DSA algorithm implicitly ignores moves that break link since transmission power required to communicate beyond range is infinite, i.e. transmission power needed after moving to new position Y from current position X, where the move breaks an existing communication link, is $V(Y, R(X)) = \infty$. The DSA algorithm ignores moves where

$V(Y, R(X)) = \infty$ as the acceptance probability is zero, i.e.

$$min\{1, exp(-\beta[V(Y, R(X)) - V(X, R(X)) + c||Y - X||/T])\} = exp(-\infty) = 0$$

3.4.3 Modifications to DSA

The DSA algorithm compares the amortized mobility cost with the savings in transmission energy. However the cost of mobility is higher than communication cost by a few order of magnitude. For a relay to consider any moves, the cost of mobility would be amortized over a long time interval. The mobility cost is tied to the environment of the mobile relay. For example, if an Unmanned Air Vehicle (UAV) is considered, there is a fixed base cost to stay aloft, and mobility cost here is the cost to navigate the vehicle in air. In this case the mobility cost is not significant compared to the communication cost. Conversely, a robotic vehicle on land expends significant amount of energy for motion compared to the communication cost.

Now in certain sensor networks, DSA can be employed by mobile relays, that are equipped with large battery reserve to extend the lifetime of energy poor static sensor nodes. Note, the mobile relays with large, potentially, unlimited energy source (for e.g. solar panels), consider mobility to reduce the transmission power of its neighbor and ignores the mobility cost. The mobile relays extend the lifetime of energy poor static sensor nodes by reducing their communication cost. This could also help in maintaining equitable dissipation of communication energy over the sensor network by moving relays closer to regions that have energy poor sensor nodes relaying data. Note, mobility cost could still be included to choose moves with least amount of mobility cost.

If the energy source for a mobile relay is limited, the DSA algorithm can consider modulating the mobility cost with the residual battery energy. A modified DSA can consider mobility cost as $c||y - x||/(TB)$ where B is the residual battery energy. Now a relay with low residual battery energy is more likely to accept a move only if the transmission energy saving is significant. Again relays with large battery reserve can ignore mobility cost when deciding on moves. Also, the battery reserve could be dependent on T. For a relay with low battery reserve, it would be better to increase T and compare mobility costs amortized over longer duration of time.

3.5 Dual tasking: surveillance and relaying

This section examines the combination of the two tasks of surveillance and relaying for the relays. The goal of the surveillance task is to move to a location in the surveillance area not visited before or not visited by a surveillance node in a long time. To quantify this a lattice is laid over the surveillance area and the number of lattice points visited is tracked. A node selects a "surveillance move" with probability p_s and "relaying move" with probability $p_r = p - p_s$ where $0 < p_s, p_r < p$.

A simple method of surveillance involves maintenance of a "taboo list" of, say, the last 10 lattice points visited by the node. When considering a surveillance move, points on the taboo list are excluded and the remaining choices are chosen uniformly at random. Alternatively, the taboo list can also maintain the *time* of the last visit to implement the list as a sliding time-window.

For both of the approaches above, the taboo lists or lists of time-stamps can be exchanged by neighboring sensor nodes and merged to create a more up-to-date table (at each sensor node), where the latter will require some kind of time time-synchronization among the sensor nodes.

In another proposed method for surveillance, sensor nodes place *marker tags* (for e.g. RFID tags with additional radio interface) and mark regions that have been surveyed. The marker tag can periodically broadcast the timestamp it was last visited. Based on the timestamps received, sensor nodes can be "attracted" or "repelled" by marker tags. When a sensor node is proximal to the marker tag, it can rewrite the timestamp of the last visit. This gives the sensor network an option to interact with the environment to determine the areas covered.

The dual tasks can also be achieved by considering alternating time durations, where for time T_{dsa} seconds the sensor nodes perform the simulated annealing moves to conserve energy with increasing β and for time T_s seconds the sensor nodes perform surveillance moves. At the start of every T_{dsa} epoch, the sensor nodes can reset β to a low value and start the DSA algorithm, gradually increasing β.

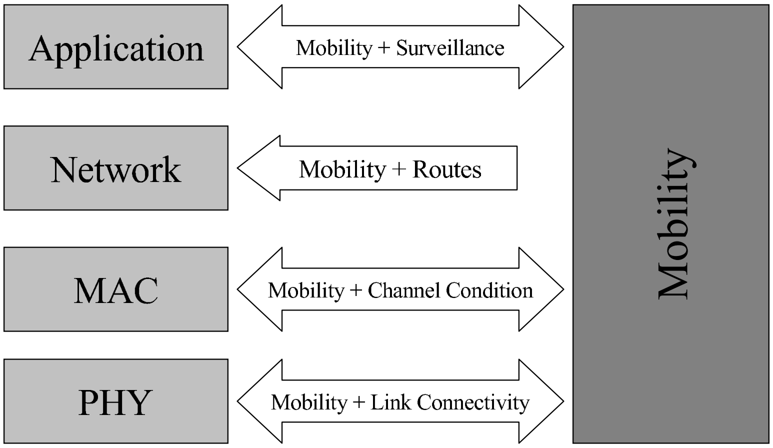


Figure 3.4. Wireless network stack and mobility interaction

3.6 Mobility and QoS issues in tactical sensor network

In the tactical sensor network considered there exists two types of traffic: a latency critical real-time video feed and a non-latency critical surveillance data. Each traffic type has specific QoS requirements and the affect of mobility, purposeful or otherwise, on the QoS is considered.

As mentioned in the introduction, mobility affects all the layers in the networking stack, starting from Application to the PHY layer. Figure 3.4 summarizes the effect of mobility on the network stack.

Here, consider the effect of mobility on the network layer, especially on routing algorithm. Since highly mobile sensor nodes disrupt the route they support, it is desirable to avoid such sensor nodes. Mobility can be included as a route metric along with metrics like, say, delay or residual energy or transmission power. Dual metric based routing algorithms are presented in [24, 16, 37]. In the current tactical sensor network transmission power and mobility metric can be considered. The mobility metric could indicates how much a node is likely to move over the next T seconds. Given such a metric that can “predict” mobility, the routing algorithm can consider the mobility metric along with the transmission power to create stable low powered routes.

For a very dynamic network, traditional routing algorithm are not favorable due to their overheads in creating routes. If the sensor network changes often enough, the routing overhead will be a large fraction of resource consumption. Alternate routing paradigm like *pheromone* based routing are considered. In a pheromone based routing algorithm, the sensor nodes flood the network and learn of all possible routes to sinks. The sensor nodes allocate pheromones for each available route to a sink and reinforces the route when it receives data on that route. Pheromone routing algorithms are explained in greater detail in [59, 66].

Mobility can be quantified as a binary metric, m, that indicates, if the sensor node will move in the next T seconds. This can be known for purposeful mobility or estimated from last T seconds for uncontrolled mobility. A mobility pheromone can be built for routes based on the mobility metric, for e.g.

$$Q_M^R = \sum_{i \in R} m_i \tag{3.11}$$

where R is the route to destination sink. The application of mobility pheromone is more relevant to uncontrolled mobility, since purposeful mobility tries to avoid moves that can break current communication links.

3.7 Discussion on implementing DSA in realistic sensor network

The problem formulation is for an ideal situation where the cost of mobility and communication is constant throughout the region of surveillance. This is not a realistic assumption. For a more realistic experiment, maps of the region can be developed, documenting mobility costs and communication cost for different locations of the surveillance region. Instead of a fixed Kd^α, consider $K_{(x_1,x_2)}d^{\alpha(x_1,x_2)}$ for communication between sensor nodes at positions x_1 and x_2 and $c_{y_1,y_2}||y_1 - y_2||$ for mobility cost between positions y_1 and y_2. If the terrain characteristics are fairly constant within communication range, the communication cost becomes $K_r d^{\alpha_r}$ and mobility cost becomes $c_s||x - y||$ where $s \in S_R$ represents a part of the surveillance region S_R where cost of communication and mobility is constant. For example, α can be large for a region with high building density (e.g. downtown Manhattan) compared to regions with more open spaces.

The values for communication and mobility can be learnt apriori and uploaded into each sensor nodes. Thus a sensor node based on its position would estimate the cost of mobility and communication to decide on a move. If such information is not available apriori, it can be learnt after deployment.

Chapter 4

Mobility strategy for surveillance

4.1 Introduction

In this chapter different mobility strategies for performing surveillance are presented. Also the different mobility strategies are compared with respect to their target detection time.

Consider a tactical sensor network deployed in region A to perform surveillance. The goal of the sensor network is to perform surveillance and report any target detection to the command-and-control center.

It is assumed that the cumulative coverage area of sensor nodes is significantly less than area of region A, i.e. if r_s is the sensing radius of a sensor node and there are N sensor nodes, then $N\pi r_s^2 \ll |A|$. The sensor nodes need to be mobile to cover the entire region A.

4.2 Relating results between surveillance domains

The purpose of this section is to relate target detection results for unbounded domains, wherein performance claims for random mobility strategies are typically more analytically tractable (e.g., [41],[45]), to those of more realistic bounded domains A with finite numbers of sensor nodes. It is assumed that the situation in the latter case is a single target located at the origin that is assumed to be in region A, i.e., the original “tile”.

In this section, it is assumed that all sensor nodes move independently. A very simplified special case of a random mobility strategy is what is subsequently called as *random initial direction* wherein each sensor node selects an initial direction uniformly at random and subsequently moves in a deterministic manner at a velocity that is common to all sensors. Another is a "hybrid" mobility strategy where each sensor node selects a new direction uniformly at random after a uniformly random number of moves, a derivative of random waypoint mobility model [13, 12]. If the sensor node is moving in the bounded domain under consideration, it will "reflect" off the domain's boundaries.

The region A is assumed to be such that it can tessellate (tile) the unbounded domain ($\mathbb{R}^2$ or $\mathbb{R}^3$) to which it belongs. The "center" of each tile is defined to be a point within it at the same relative position with respect to its boundaries that 0 is with respect to those of the original tile A, so that 0 is the center of A in particular. Motion of sensors between adjacently tiles in the unbounded domain (i.e., direction changes *at* tile boundaries) is such that the reflected motion defined the bounded region A, i.e., the relative position of the reflected sensor with respect to the center $0 \in A$, is the same as that of the "refracted" sensor with respect to the center of the tile in which it has moved, see Figure 4.1 for the example of a rectangular A under random initial direction motion.

4.3 Problem formulation

It is usually simpler to develop mobility models for sensor nodes in unbounded domains of infinite area [41],[45]. Sensor nodes are deployed according to a Spatial Poisson Process with density λ. However in practical cases, the region of surveillance is bounded. In a bounded region, again, consider N nodes deployed according to a Spatial Poisson Process with node density $\lambda = N/|A|$ where $|A|$ is the area of region. Simulations are performed to compare the target detection time distribution in a bounded and unbounded domain without tiled target.

"Tiled targets" refers to an unbounded domain wherein there is a target at the center of every tile. The following theorem immediately follows from a simple argument involving coupling of the motion of the finite number N of sensor nodes in the bounded and unbounded domain cases.

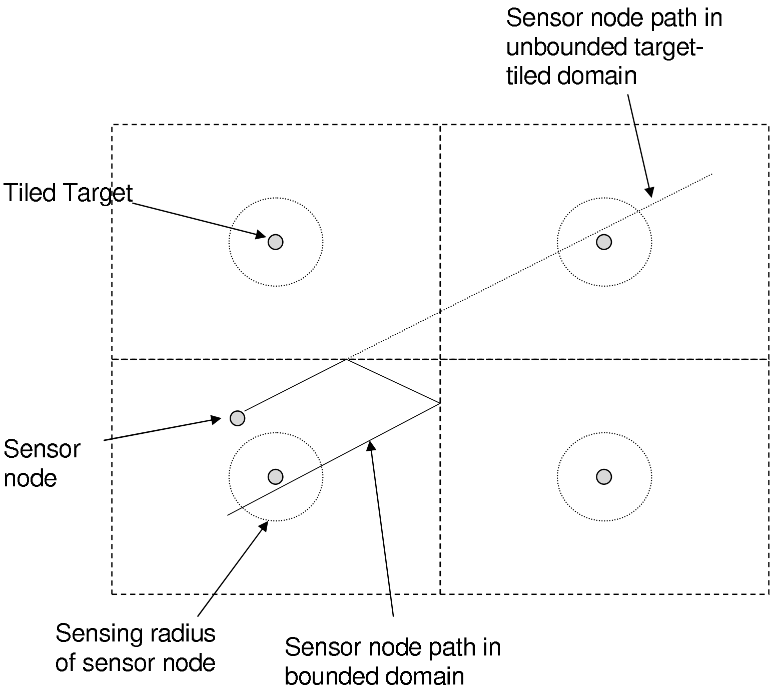

Sensor node path in unbounded target-tiled domain
Tiled Target
Sensor node
Sensing radius of sensor node
Sensor node path in bounded domain

Figure 4.1. Coupling between bounded and unbounded target-tiled domains

Theorem 2. *Under any random mobility strategy, the target detection time distribution for a bounded domain is equal to the* first *target detection time distribution of an unbounded domain with*

- *tiled targets, and*
- *the same finite number N of sensor nodes all initially located in the original tile A and moving according to the same distributions.*

Proof: From geometry (figure 4.1).

Hypothesis 1. *Under any random mobility strategy, the target detection time distribution for a bounded domain is* similar *to the target detection time distribution of an unbounded domain with*

- *a single target at the origin (not tiled targets),*
- *the initial mean sensor density $N/|A|$ throughout the unbounded domain (i.e., infinitely many sensors), and*

- *the same mobility distribution for all sensors,*

as in the finite domain case.

The intuition behind the hypothesis is that, through mobility, the *mean number* of sensors in A will always be the same (N) and, therefore, the target detection times will be similarly distributed. Figure 4.2 shows a representation of the assumptions made. The mean number of nodes in the bounded domain is constant as the number of nodes leaving and entering the bounded domain are equal in mean.

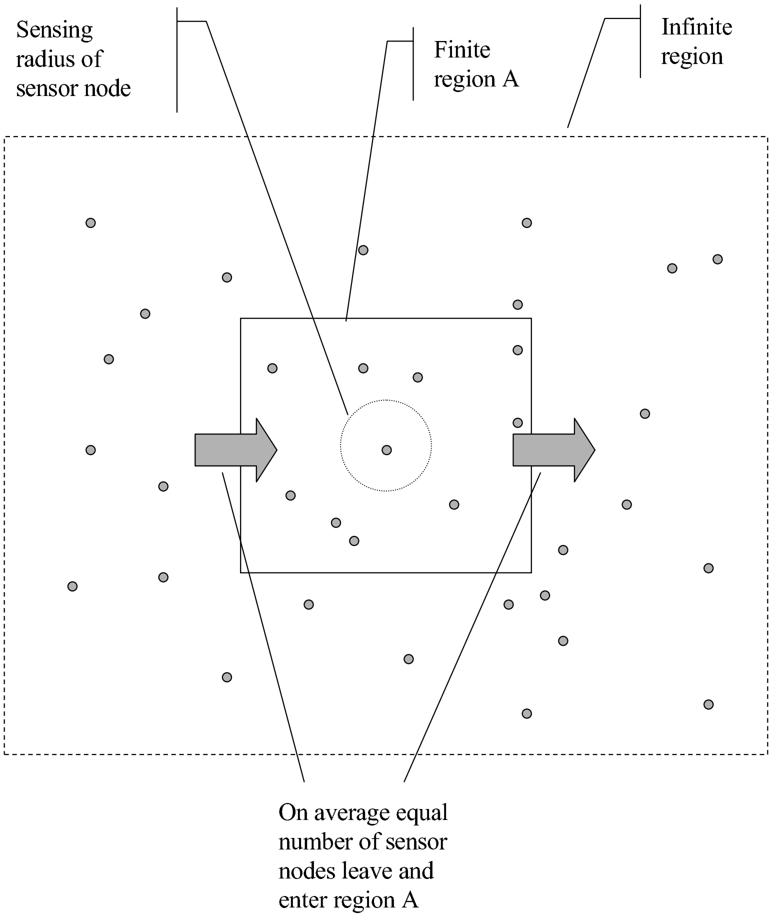

Sensing radius of sensor node
Finite region A
Infinite region
On average equal number of sensor nodes leave and enter region A

Figure 4.2. Coupling between bounded and unbounded with non-tiled target

The above hypothesis is explored by simulation for the case of initial placement of sensor nodes according to a spatial Poisson process with density $N/|A|$. More

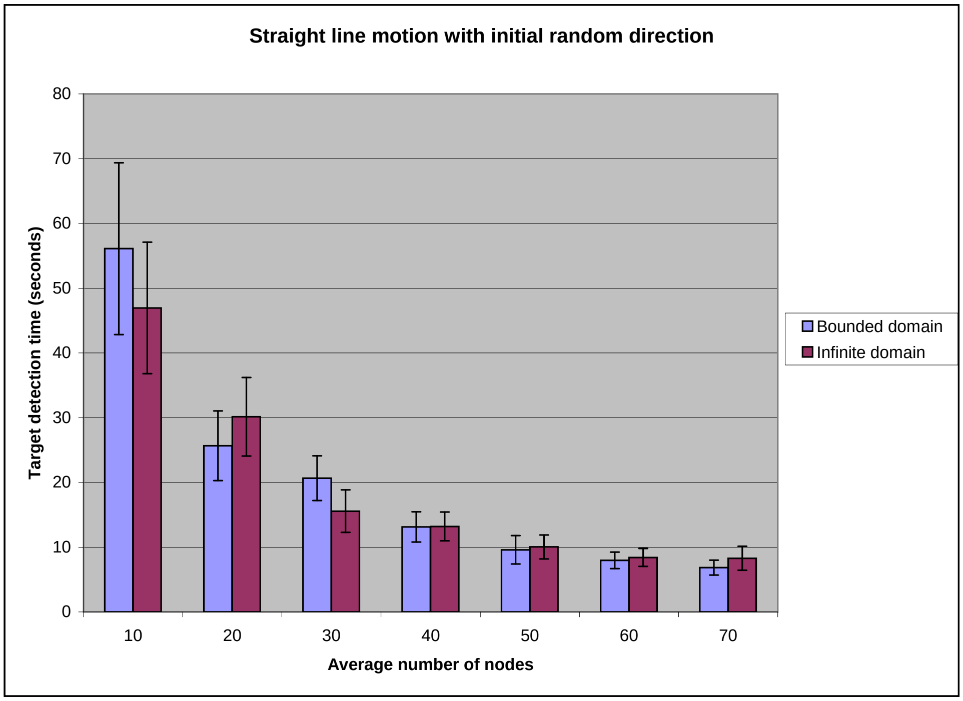


Figure 4.3. Comparison of finite domain and infinite domain mean target detection time under random initial direction

specifically, consider a 100m x 100m area for the bounded domain A and rectangle of 1000m x 1000m area for an unbounded domain without tiled-target, both literally centered at the origin. The target detection radius of a sensor node is 10m. The vertical line across the top of each bar in the graph indicates a confidence interval of 95% from 10 trials.

Figure 4.3 shows the comparison of the mean target detection time under initial random direction mobility strategy **with velocity** $v = 1m/s$ in the bounded and unbounded domains without tiled-targets. The x-axis represents the average number of nodes in a 100m x 100m area. The results for the case of hybrid mobility strategy **with parameters n** $\sim$ **U(2,10)**, shown in figure 4.4, are similarly consistent with the statement of the Hypothesis 1.

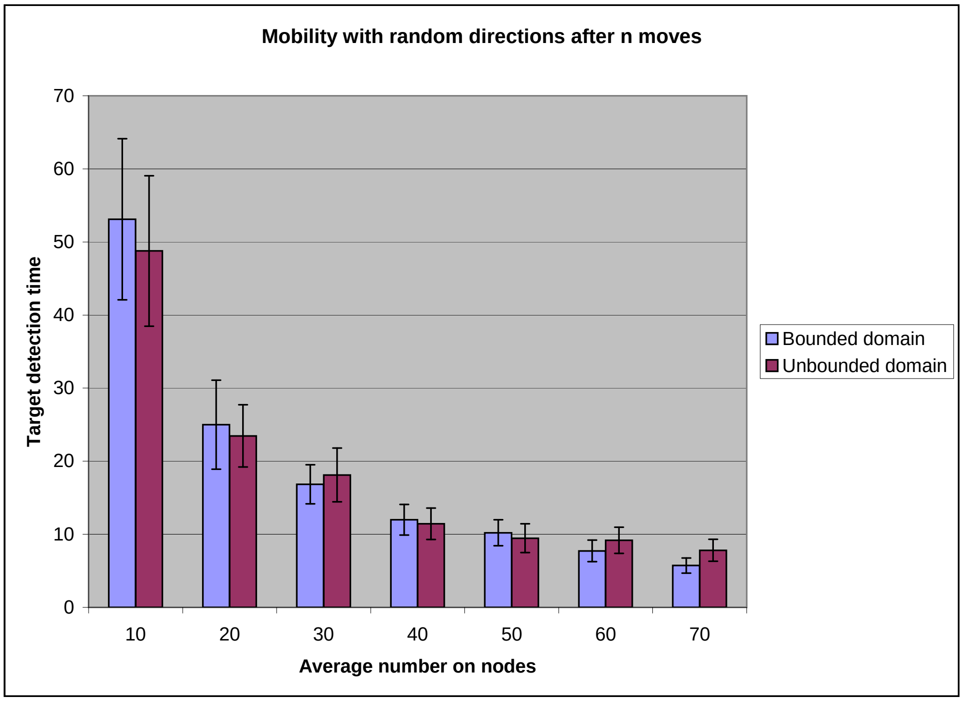


Figure 4.4. Comparison of finite domain and infinite domain mean target detection time for hybrid motion

Chapter 5

Simulation Study of Purposeful Mobility in Sensor Networks

In this chapter the simulation study of the DSA and the greedy algorithms is presented. The simulation consists of three parts: the first set of simulations compare the performance of DSA and the greedy algorithms, the second set of simulations compares the performance of DSA for different values of β and the last set of simulations examine the performance of the DSA algorithm when the relays also consider mobility for surveillance in addition to mobility for transmission energy conservation.

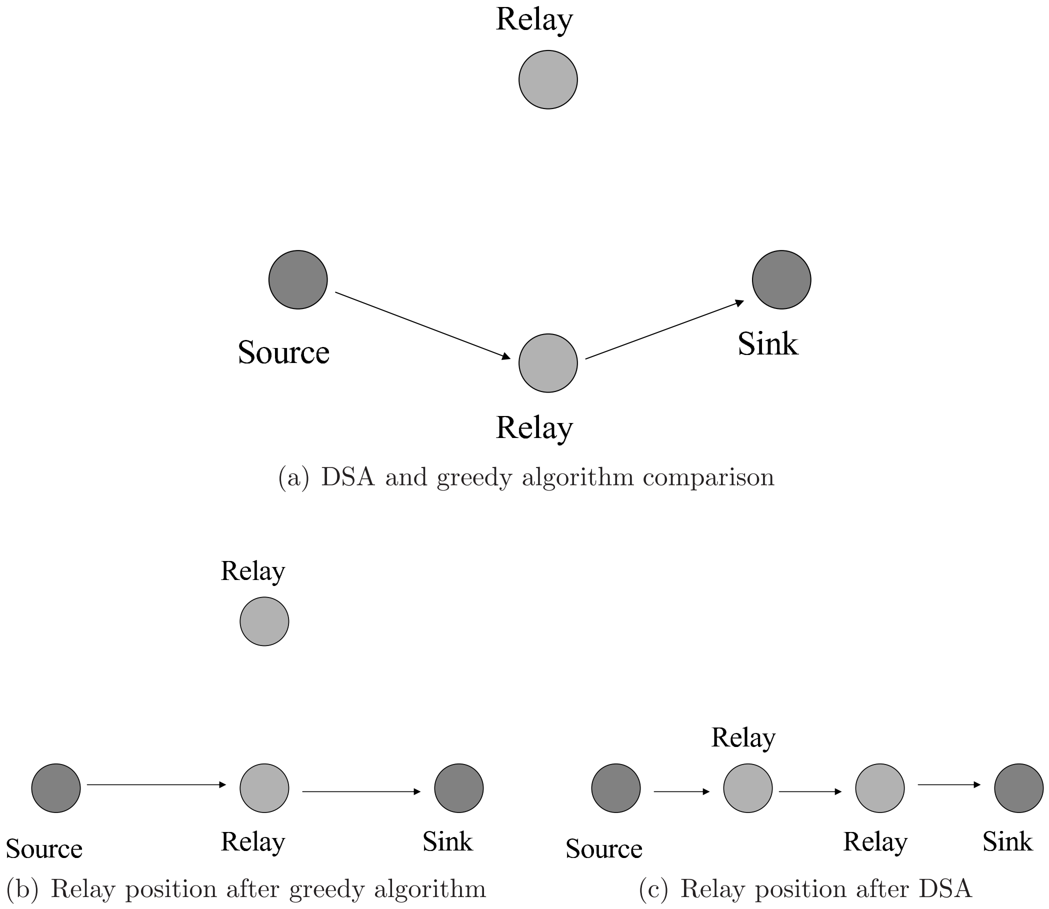
For all the simulations the sensor network consists of static sources and sinks with mobile relays. It is assumed that the intermediate relays do not generate any data of their own, i.e. data generated for surveillance task is ignored because of its low volume.

5.1 Comparison of DSA and greedy algorithm

In this section a comparison of DSA and greedy algorithm is presented. For this set of simulation, a specific sensor network layout, shown in figure 5.1(a) is considered. The sensor network consists of a source, a sink and two relays.

For the current simulation, the mobility constant, c, from the global utility function (3.4), is 5mJ/m, i.e., a relay weighing 1kg requires 5mJ to move a distance of 1m. The parameter K from equation (3.2), determines the communication power required to transmit packets over a distance of 1m, is assumed to be $1mW$. Note K also factors in the data transmission rate. Since, the data stream from sources is assumed to be live real-time video feeds, the data rate is of the order of Mbits/s. Also for this equation, the communication attenuation parameter is assumed to be $\alpha = 2.5$. The greedy and the DSA algorithms make a decision of mobility every $T = 10s$.

The values of c, K and T are conveniently chosen for the simulations. Note the value of c is choosen to make mobility cost more comparable to communication cost for $T = 10s$. For alternate values of c and k, the value of T can be changed to make the communication and mobility cost comparable. Say, if the cost of communication is significantly less, i.e., K is very low compared to c, the value of T can be increased to compare the transmission energy against the mobility cost amortized over larger period of time.



(a) DSA and greedy algorithm comparison

(b) Relay position after greedy algorithm

(c) Relay position after DSA

Figure 5.1. Sensor network for comparison of DSA and greedy algorithms

Consider the sensor network shown in figure 5.1(a) for the comparison of performance of DSA and greedy algorithm. The network layout demonstrates that the greedy algorithm converges to a "suboptimal" result (figure 5.1(b), while the DSA algorithm performs better on average and converges to the optimal position 5.1(c). The sensor network consists of a source, a sink and two relays. For the routing algorithm, a modified version of the distributed Bellman Ford routing algorithm is used with Kd^α as the metric.

Since the greedy algorithm execution depends on the initial node position, it converges to a deterministic result for a given initial set of sensor node position. Therefore the greedy algorithm is executed once. Conversely, the DSA algorithm may produce different result for each simulation run. Therefore the ensemble average of the total transmission power from ten simulation runs is considered. The vertical bar across the mean value in both the graphs represents a 95% confidence interval.

The results of the comparison of greedy algorithm and DSA algorithm are presented in figure 5.2. The graphs compares the total tranmission power of the greedy and DSA algorithm over time. The DSA algorithm is plotted for two values of β (0.5 and 5). Note, since a cooling schedule for the DSA algorithm is not possible (section 3.4.1), a constant β is considered. For lower values of $\beta = 0.5$ (i.e., higher temperature), the DSA algorithm performs worse than the greedy algorithm. The relays do not move to improve the total transmission power and continue to move to positions worse than the initial position. The bad performance of the DSA algorithm is due to the higher probability of accepting a "bad move" (for $\beta = 0.5$). A bad move is defined as a move where the amortized mobility cost is greater than the savings in transmission energy, i.e., a relay spends more energy is motion compared to the amortized savings in transmission energy. For lower values of β, a relay chooses a position at random and with very high probability moves to the position, irrespective of the affect on the total transmission power, i.e., in effect the DSA reduces to a random mobility algorithm. For $\beta = 5$ (i.e., lower temperature), the DSA algorithm performs better on average compared to the greedy algorithm. Note, after about 650s, the DSA algorithm converges to the optimal relay position (figure 5.1(c)). The relays have a lower probability of accepting a bad move, but accept bad moves to climb out of the local minima to

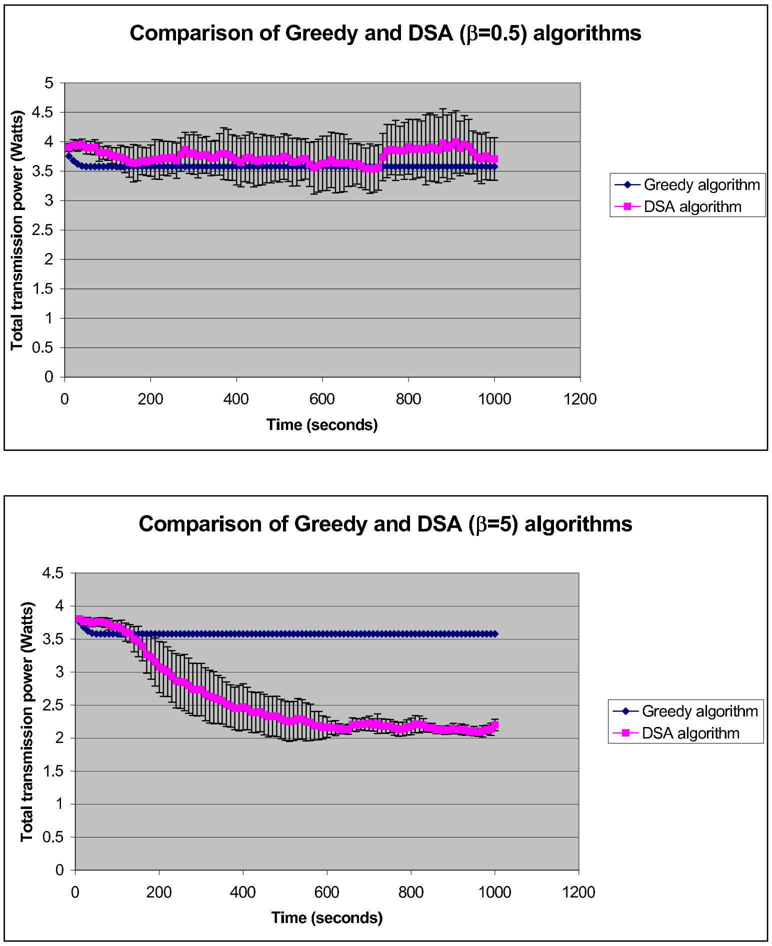


Figure 5.2. Comparison of total transmission power of DSA and greedy algorithm for different fixed β

reach the global optima. For higher values of β $(= 50)$, the DSA algorithm, on average, continues to perform better than the greedy algorithm, but it does not reach the globally optimal positions for all sample runs. For some sample runs the performance of the DSA algorithm is similar to the greedy algorithm. Note, the fraction of sample runs performing similar to the greedy algorithm increases for higher values of β. For even higher values of β $(= 500)$, the DSA algorithm performs similar to the greedy algorithm, i.e., relays only accept moves that reduce their total transmission power.

The DSA algorithm performs worse than the greedy algorithm for $\beta = 0.5$, better than the greedy algorithm for $\beta = 5$ and for higher values $(\beta = 50)$ it performs, at worst, similar to the greedy algorithm for a fraction of the sample

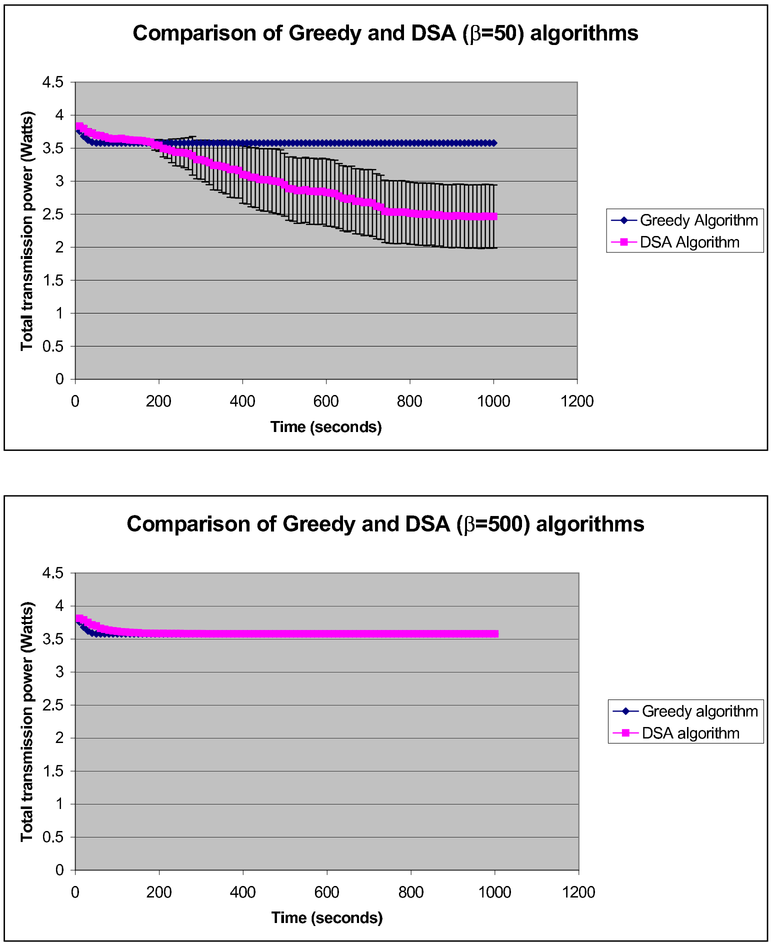


Figure 5.3. Comparison of total transmission power of DSA and greedy algorithm for higher β

runs. Note for very low values of $\beta = 0.5$, the DSA algorithm acts like a random mobility algorithm, where the impact of mobility on total transmission power is ignored and for very high values of $\beta = 500$, the DSA algorithm performs similar to the greedy algorithm.

Since the DSA algorithm performs worse than the greedy algorithm for $\beta = 0.5$, better than greedy algorithm for $\beta = 5$ and, at worst, similar to greedy algorithm for $\beta = 50$, it is necessary to select a convenient β such that a relay explores its immediate neighborhood for alternate routes and improves (if possible) on its current position. The DSA algorithm might not converge to the global optima, but converge to a local optima and in the current sensor network context, where sensor nodes may also be performing other tasks like surveillance, a local optima might be even desirable compared to a global optima. This is explained in greater detail in section 5.2.2.

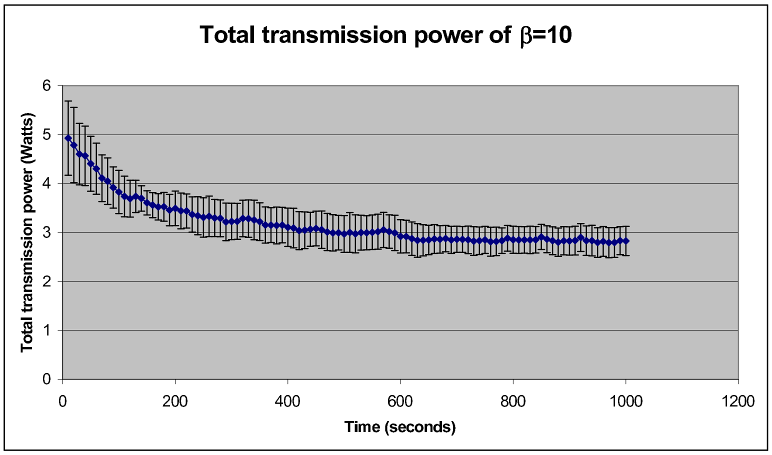
5.2 Simulation for single and dual tasking relays

The next set of simulations examine the performance of the DSA algorithm for various values of β and the impact of surveillance mobility on DSA. The sensor network considered for the simulations, consists of seven sensor nodes, made up of two stationary (immobile) sources, a stationary sink and four mobile relays, operating within a 40m×40m square area (for the following simulations, the units of distance can be scaled-up by suitable modification of the following mobility and communication parameters). Also the sources generate data at a constant bit rate. The communication radius is 50m, i.e., all sensor nodes can talk to one another.

Initially, the positions $X(0)$ of the relays are chosen independently and uniformly at random in the area under consideration. Also, it is assumed that the active relays move at all times, i.e., $p = 1$ instead of $p = 1/N$ as advised at the end of section 3.3. Note the term active relay refers to the relay actively involved in forwarding data to a sink. Relays not involved in data forwarding do not consider mobility. Due to the constraints of the simulation setup, only one relay can move at a time. Though all nodes decide to move with $p = 1$, each move is considered separately. Again, the communication and mobility parameters, c and K are 5mJ/m and 1mW per meter respectively. The value of α is 2.5 and relays decide on a move every $T = 10$ seconds. A transmission power-based (distributed Bellman-Ford with d^α as metric) routing algorithm is assumed to be in effect determining the routes $R(X)$ at node-positions X (again, assumed to operate at a time-scale much faster than that of node movement, T). Finally, from 10 simulation runs, an empirical mean and a 95% confidence interval is determined. The vertical bars across the mean values represent the 95% confidence interval.

5.2.1 Results for single task per node

Figure 5.4 shows a plot of the total transmission power of the network over time for different values of β (inverse temperature). Note β is constant for each simulation



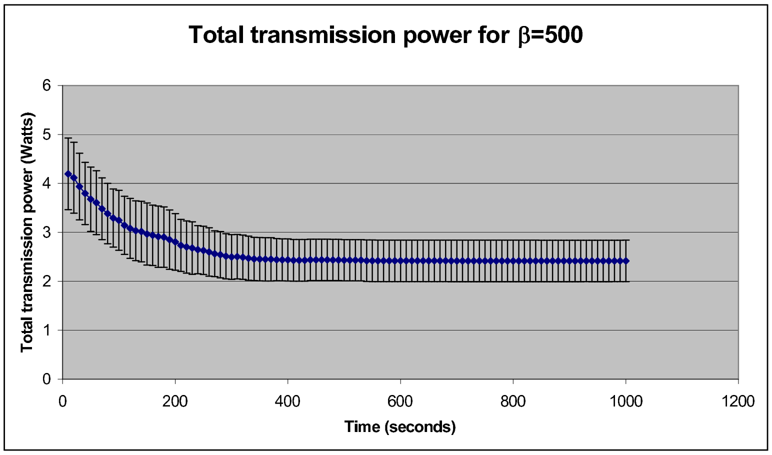


Figure 5.4. Total transmission power over time for different fixed β

run. Figure 5.4 plots the total transmission power for $\beta = 10$ and $\beta = 500$.

Both the plots show an improvement in the total transmission power. The total transmission power are averaged over different initial positions from 10 sample runs. The plots in figure 5.4 indicate that $\beta = 500$ performs better than $\beta = 10$. However, this does not imply that the relays find better positions for $\beta = 500$ compared to $\beta = 10$. For $\beta = 10$, the relays find the best possible positions (locally or globally optimal positions), but continue to move, due to higher probability of accepting a bad move. That is, the relays continue to move in the neighborhood of their best position and this is reflected in the temporal variance of the total transmission power for $\beta = 10$. Conversely, the relays with $\beta = 500$ reach their best possible position and remain stationary, since the probability of accepting a bad move is low. Thus, on average, a relay with $\beta = 500$ is more likely to be at its best possible position (a globally or locally optimal position) compared to a relay with $\beta = 10$.

Also the relays converge to their best possible positions faster for $\beta = 500$ than $\beta = 10$. In figure 5.4, for $\beta = 10$ the relays reach their best possible position around 610 seconds compared to 370 seconds for $\beta = 500$. A relay with $\beta = 500$ makes lesser number of bad moves compared to a relay with $\beta = 10$. This is due to the lower probability of DSA with $\beta = 500$ to accept a bad move compared to DSA with $\beta = 10$. Thus a relay with $\beta = 500$ is more likely to make moves that reduce its total transmission power and reach its best possible position faster than a relay with $\beta = 10$.

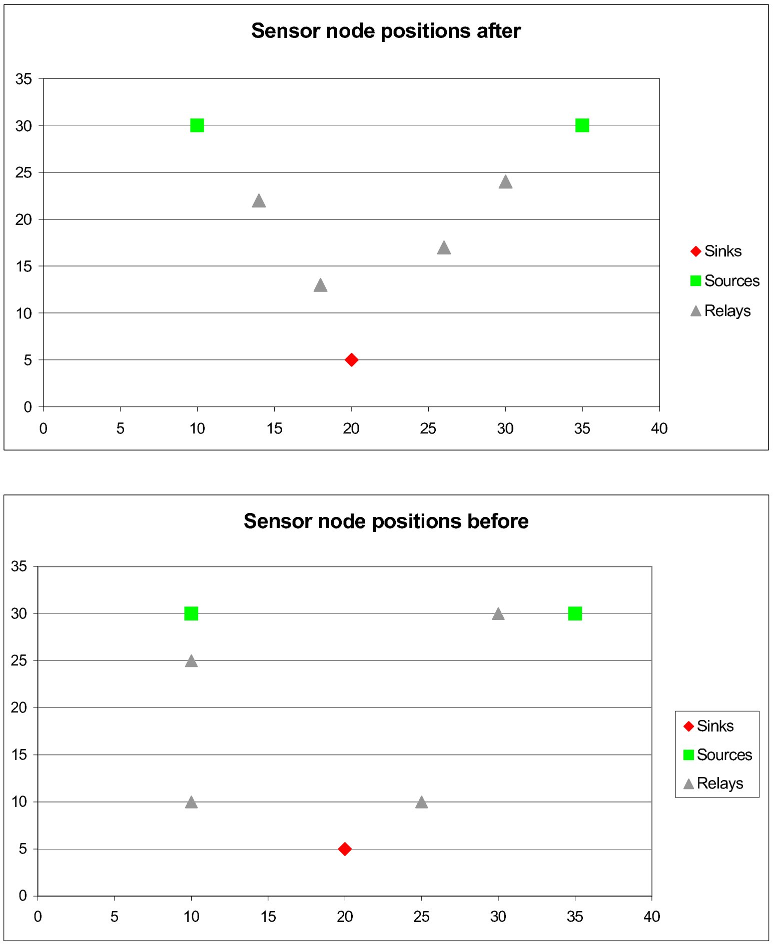


Figure 5.5. Initial and final node positions

Figure 5.5 shows the position of the sensor nodes before and after DSA algorithm for a sample run. The figure illustrates how the relays create a near-linear path between the sources and sink to reduce total transmission distance.

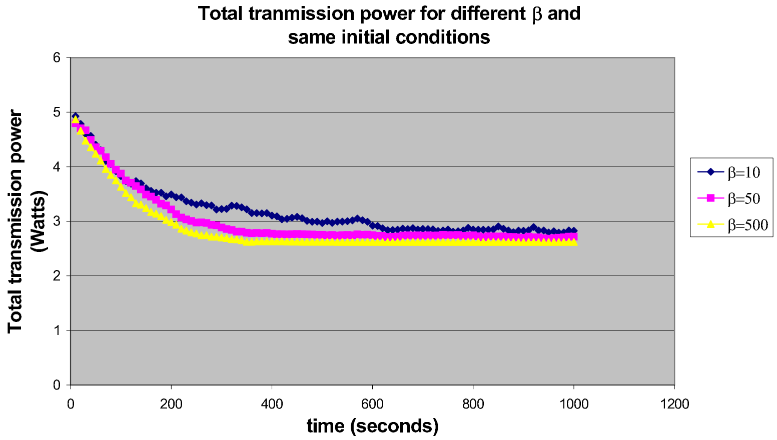


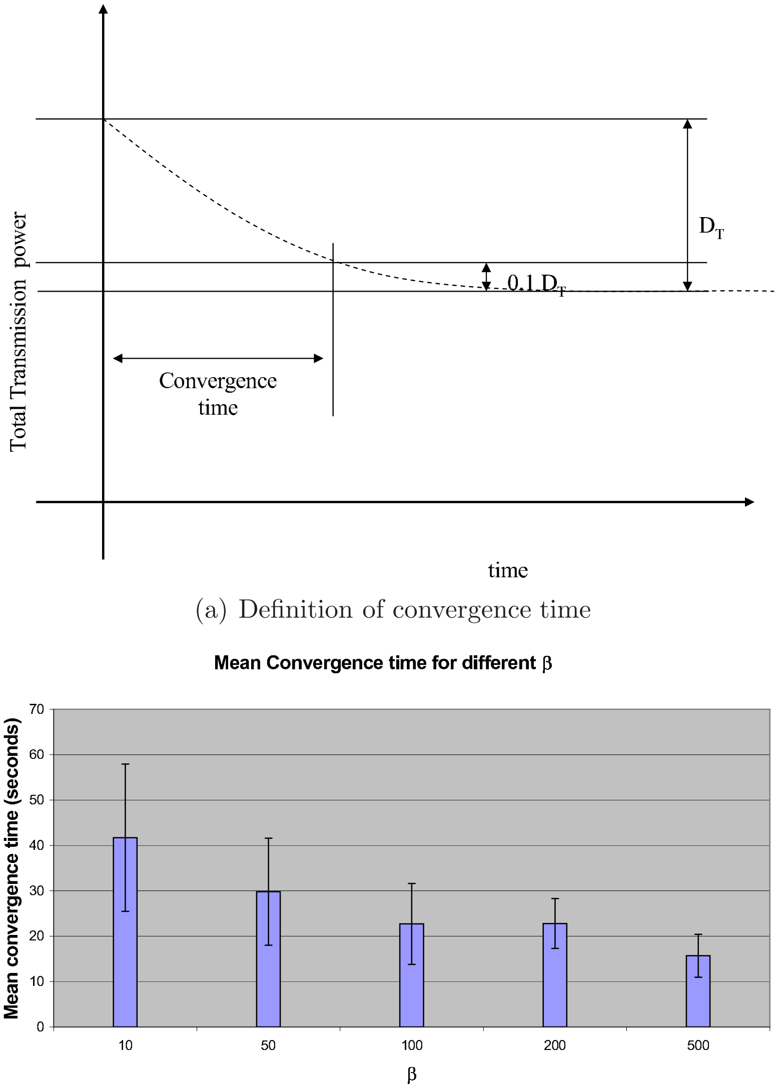
Figure 5.6. Total transmission power for different β with same intial relay position

Note the plots in figure 5.4 compares the mean total transmission power plot averaged over different initial positions. The next set of simulations, compares the total transmission power for different β with the same initial relay positions. For the following simulation, intial positions of relays are randomly generated, and the DSA algorithm is executed for different values of β. The simulation is repeated 10 times to get 10 sample runs for different β with same set of intial relay positions.

Figure 5.6 plots the mean total transmission power for $\beta = 10, 50, 500$ with same inital positions. The results are similar to the plots in figure 5.4. For higher β, the total transmission power is lower and relays reach their best possible positions faster than for lower β.

5.2.1.1 Convergence time

Figure 5.7(b) shows a plot of the *convergence* time for different values of β. Figure 5.7(a) illustrates the definition of convergence time. The convergence time is the time taken by the relays to reach their best possible positions. Since the relays continue to make bad moves for lower values of β, they do not settle on a single position, but occupy a region around their best possible position. This is reflected in the temporal variations of the mean total transmission power (figure 5.4). To define convergence time, a curve-fit on the total transmission power, i.e., a simple moving average (SMA), is considered. The SMA plot smoothens the total transmission power plot to better determine the final total transmission power of the



(a) Definition of convergence time

(b) Convergence time for different β

Figure 5.7. Convergence time

sensor network, especially for lower values of β. Now, the convergence time is defined as the time at which the SMA plot of total transmission power has reduced by 90% of the difference between the inital and final total transmission power. This is better illustrated in figure 5.7(a)

The convergence time is determined for each sample run from the SMA plot of the total transmission power. Note, the moving average is considered over the last 100 seconds, or the last 10 readings of the total transmission power. Figure

5.7(b) plots the average convergence time for difference values of β. The vertical bar across the top of the bar plot is the 95% confidence interval from the 10 sample runs. The convergence time decreases on average with increase in β.

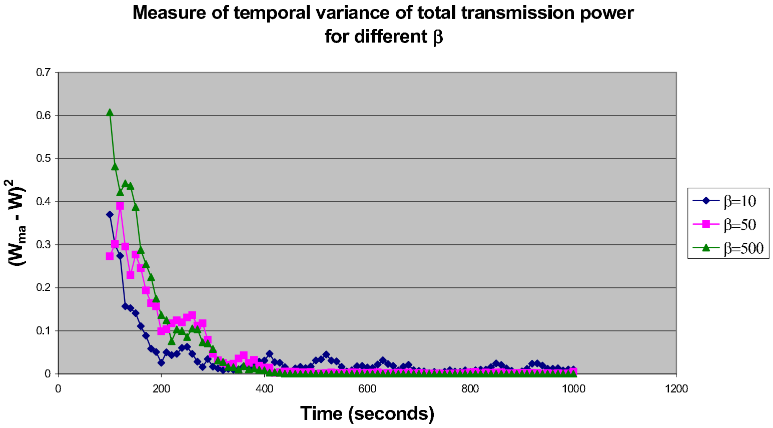
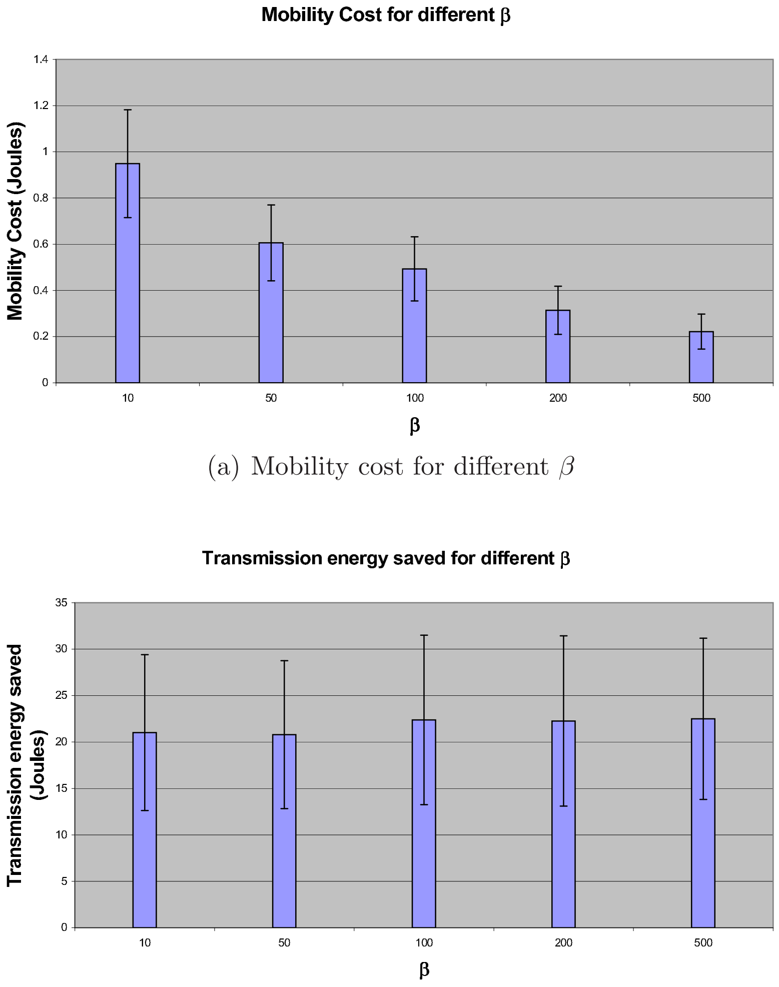


Figure 5.8. Temporal variations in total transmission power value for different β

Figure 5.8 plots the temporal variance seen in the plots of the total transmission power for different β. The total transmission power is considered as a time series and the variance over the last M seconds measures the temporal variance. The temporal variance after convergence time gives a measure of how much the relays move about the best possible position. The temporal variance can be measured from the variance of the moving average model for the last M seconds. For the current plot of the total transmission power, we consider a simple moving average model over the last 100 seconds (10 readings) to measure the temporal variance. The temporal variance is measured by $(W_{ma}(t) - W(t))^2$ where $W(t)$ is the total transmission power at time t and $W_{ma}(t)$ represents the simple moving average over the last 100 seconds (or 10 reading) of $W(t)$. Figure 5.8 plots the temporal variance for the total transmission power of the sensor network. The plot indicates that the temporal variance is higher for $\beta = 10$ and comparable for $\beta = 50$ and $\beta = 500$.

5.2.1.2 Energy consumption and savings

So far, the plots for the total transmission power do not indicate the mobility cost involved in moving the relays or the savings in the total transmission power. Figure 5.9 plots the mobility cost and the energy savings in transmission energy.



(a) Mobility cost for different β

(b) Total transmission energy saved

Figure 5.9. Mobility cost comparison for different values of β

The energy saved is defined as

$$(W(t-T) - W(t)) * T$$

where $W(t)$ represents the total transmission power at time t and T is the duration of the time interval between each move. It is assumed that the relays make their move during the start of every T interval and the saved transmission energy represents the difference in transmission energy between the two positions. Note, when a relay makes a bad move, the savings in transmission energy is negative. Figure 5.9(a) shows that the mobility cost decreases with increase in β, but figure

5.9(b) shows that the total transmission energy saved is similar for all value of β. This indicates that it is better to chose higher values of β where the mobility cost is the least, but transmission energy saved in similar to lower values of β.

Note, the mobility and communication costs are controlled by the c and K parameter chosen conveniently for the simulation. A more realistic sensor network could have mobility cost higher than the communication cost by couple of order of magnitude. The mobility cost is dictated by the environment of mobile relay. However, as mentioned in section 3.4.3, few energy rich relays could consider mobility to reduce the transmission power of neighboring energy poor static sensor nodes.

In comparing the total transmission power, the convergence time, mobility cost and energy saved higher value of β produce better results. However, in section 5.1 it is observed that β must be low enough to allow the relays to explore their neighborhood for alternate routes to reduce total transmission power. For the simulation setup considered, a good value of β is about 500.

5.2.1.3 Revisiting the motivation for DSA

The motivation for using DSA is that the cost function $V(X, R(X))$ could be multimodal and the greedy algorithm could be trapped in a local minima. However, the details of the cost surface are unclear and if there are not many local minimas, the benefits of the DSA algorithm will be limited. In such cases, the greedy algorithm would suffice. The next set of simulations are considered to determine the total number of minima present. The sensor network setup is similar to the previous simulations (figure 5.10), i.e., the position of the sources and the sink are same while the relays are randomly deployed in the region (rectangular box in figure 5.10) between the sources and the sink. To determine the local minimas, an initial random positions for relays is considered and the greedy algorithm executed. The greedy algorithm converges to a local minima. The relays converge to their final position and the total transmission power value is dependent on the relay positions and the route selection. The simulation is executed multiple times for different initial random placement of relays. Each local minima represents a unique position vector X' and route $R(X')$ giving a unique value for the total transmission power $V(X', R(X'))$. Figure 5.11 gives and example of six positions and the

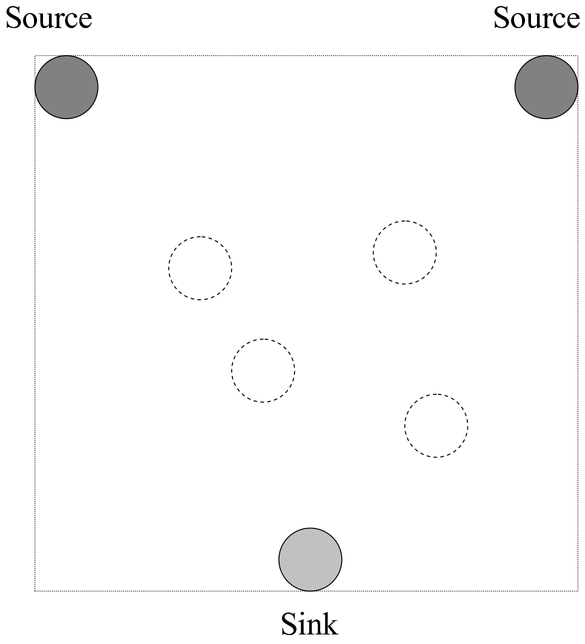


Figure 5.10. Sensor network layout to find number of local minima

total transmission power. The simulation resulted in 16 unique values of total transmission power (figure 5.12) indicating that there are atleast 16 minimas for the given sensor network setup. This is a significant number for a sensor network with two sources, a sink and four relays and is ample justification to consider the DSA algorithm.

5.2.2 Dual tasking per node

In this set of simulations, it is assumed that the relays have both, a surveillance and relaying function, as described in section 3.5. A simple method involving a taboo list ([76]) of the last 10 points visited by the node is simulated. Neighboring nodes do not exchange taboo lists. Finally, it is assumed that the transmission energy for surveillance (passive surveillance) traffic is negligible compared to that of the tracking traffic and the former is not accounted for.

For two values of the surveillance-decision probability parameter, $p_s \in \{0.4,\ 0.6\}$, the communication energy is plotted in figure 5.13. One can also compare with the second graph of figure 5.4 in which $p_s = 0$. Note that the effect of increasing p_s (i.e., a greater propensity for surveillance moves) has an effect on communication

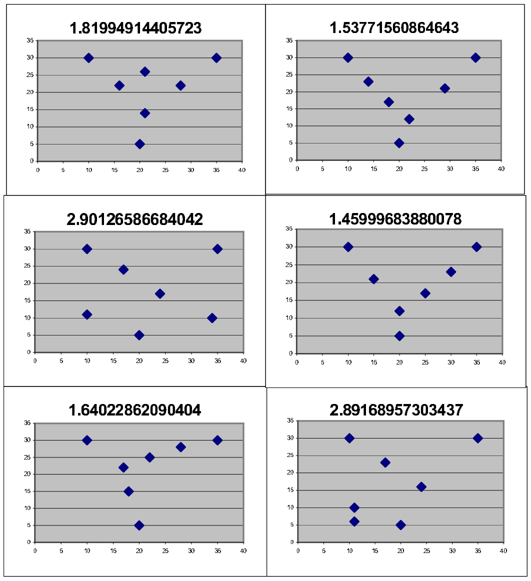


Figure 5.11. Example network positions with different local minimas

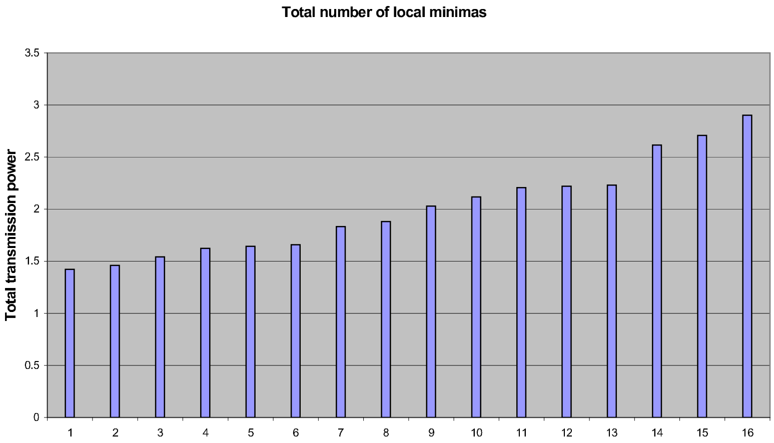


Figure 5.12. Total transmission power for different local minimas

energy similar to that of increasing temperature for singly tasked (relay) nodes as indicated in Figure 5.4. For $p_s = 0.4$, there is a better performance in terms of total transmission power compared to $p_s = 0.6$. A surveillance move can be considered similar to the bad move in the DSA algorithm, in that, the move may not help in reducing total transmisssion power. Of course, the advantage of increasing p_s is that the nodes cover (scan) more of the area under surveillance. Defining coverage as the total number of *different* points visited by all of the scan-relay nodes over a sliding time-window of twenty seconds, the $p_s = 0.6$ trials depicted in Figure

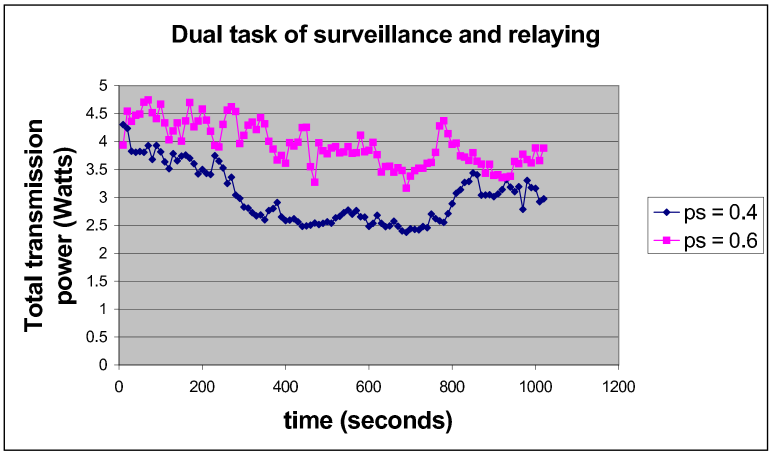
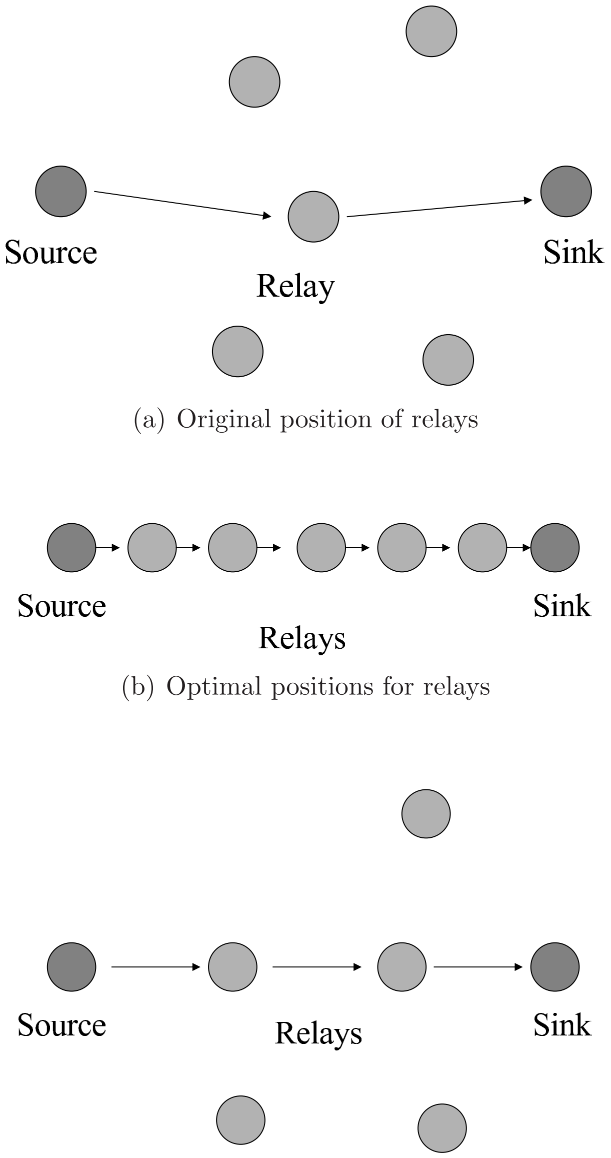


Figure 5.13. Total transmission power over time

5.13 achieved 25% more coverage on average than the trials using $p_s = 0.4$. Indeed both p_s and the temperature β^{-1} affect coverage which significantly increased with decreasing β for the trials of Figure 5.4.

Note, in a sensor network where sensor nodes perform the dual task of relaying and surveillance, it is desirable that the DSA algorithm converge to local optima and not to global optima. Global optimal would require the relays to occupy the region between the sources and sink and could lead to sensing holes in rest of the region. For example, consider the sensor network shown in figure 5.14 that illustrates this point better. The original position of sensor nodes is shown in figure 5.14(a), with the sink and source indicated by the darker circles and the relays represented by lighter circles. The globally optimal position for the relays is shown in figure 5.14(b). However this is not possible with a constant β considered for the DSA algorithm and a sufficienlty high β is necessary for a relay to explore its immediate neighborhood for better routes and converge to a local optima as shown in figure 5.14(c). The advantage in limiting the mobility of relays, is the prevention of sensing holes in the region. Note, since a global optima could lead to sensing holes (figure 5.14(b)) and the sensor network is also tasked with surveillance, a global optimal is not a desirable outcome. Therefore limitation on relay mobility prevents creation of such sensing holes.

Note the figures in 5.14 are example sensor network to illustrated the influence of β and not simulation results.



(a) Original position of relays

(b) Optimal positions for relays

(c) Non-optimal, but acceptable positions for relays

Figure 5.14. Influence of β on optimal relay positions

Chapter 6

Anycast Routing

6.1 Introduction

The simulation results in the previous chapter only consider a single sink as target and all the sensor nodes route to this sink. However in a tactical sensor network there are multiple sinks and sensor nodes can route data to any of the sinks. The routing algorithm applied here follow the *anycasting* paradigm. In this section the application of DSA to tactical sensor networks with anycasting is considered. Although anycasting does not affect the mobility of the relay, in that, the decision of a relay to move is unmodified, but due to its affect on routing, it is necessary to consider affect of anycasting on DSA.

In the internet context anycasting is applied to "server farms" where a group of servers offers the same service. An anycast address represents the group of servers. A client requests service by querying the anycast address. The request is forwarded to any one of the servers by anycast-aware routers [55]. The anycasting here is done at the networking layer. An alternate implementation of anycasting is through the application layer [9], where a request for service is received at a anycast domain name resolver. The anycast domain name resolver directs the request to one of the servers based on some criterion, for e.g. distance from client to server or current load experienced at servers.

In the sensor network context, anycasting is applied to communicate with groups of sensor nodes or sinks that can perform a particular task. For example, sensor nodes co-located in a region are grouped into an anycast group and are responsible for providing data for that region. Anycasting can be implemented by extending unicast based routing algorithm [54].

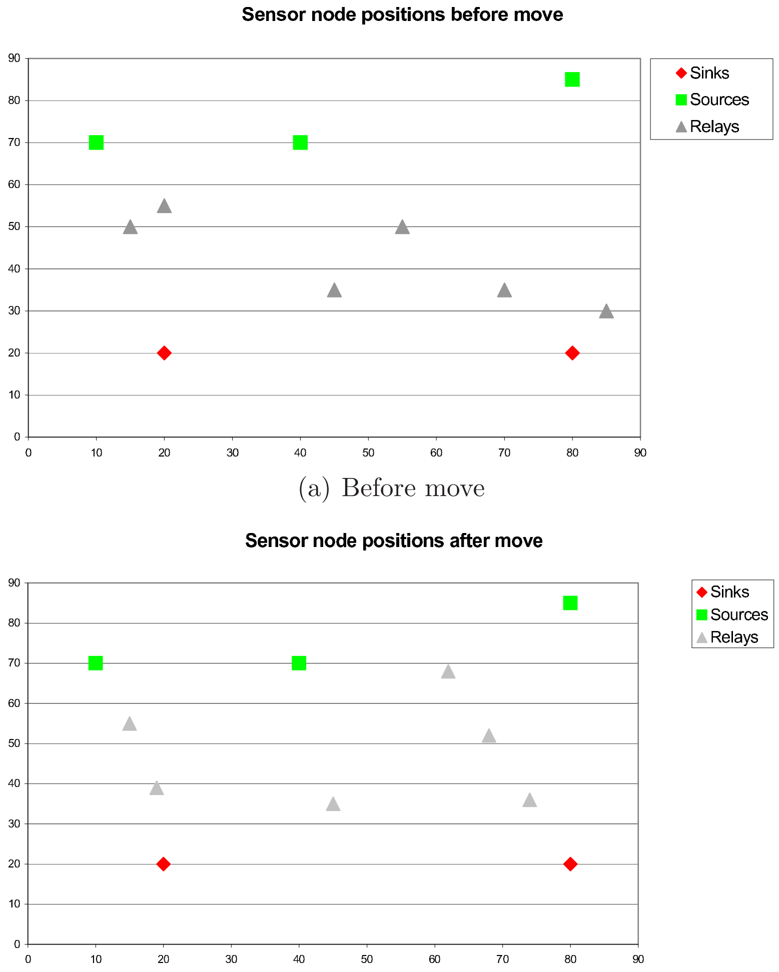
In the current simulation anycasting is implemented to forward data from sources to the nearest sinks. The Distributed Bellman Ford algorithm is applied to find routes to all the available sinks. A source selects the nearest sink from the list in its routing table based on route costs. In the current simulation setup only transmission power cost is considered, but this can be easily extended, with some feedback from sinks, to include other metrics, e.g., energy richness of path to sinks, delays to sinks and load handled at each sink. In particular anycasting in conjunction with pheromone based routing is particularly suited for scenarios where the sink and sensor network characteristics are dynamic.

Note, the destination address of the anycast group (i.e. sinks) is resolved at the source. Alternately a sensor node can be assigned to act as an anycast address resolvers that monitor the status of the sinks to determine the best sink for a source node. Instead of the sinks flooding changes in their status, e.g. load handled, residual energy, they can forward their status to anycast address resolver nodes. Sources forward their request to access an anycast group member to the anycast address resolver node. The anycast address resolver node determines the best sink for the source based on certain criterion.

6.2 Extending DSA to Anycasting

In the current simulation setup consider "nearest" sink as the sink closest to the source in terms of routing metric, i.e. $\sum Kd^{\alpha}$, the total transmission power cost. Again, a distributed Bellman Ford routing algorithm is considered with the transmission energy, Kd^{α}, as the route metric. The sensor nodes maintain routes to all the sinks and choose the route with the best metric. Note that during moves made by DSA, the sources could find an alternate sink closer than its initial target sink. However this does not affect the effectiveness of DSA since the assumption that a DSA move will not create a higher cost route still holds true. If a sensor node switches sinks after a DSA move, it will only do so if the route to the alternate sinks is more cost effective.

For the simulation setup consider a tactical sensor network with multiple sinks,



(a) Before move

(b) After move

Figure 6.1. Anycasting in sensor network with multiple sources and sinks

sources and relays. Figure 6.1 shows an application of DSA to a sensor network with multiple sources and sinks. Figure 6.1(a) shows the sensor nodes before the DSA algorithm is run and figure 6.1(b) shows the sensor nodes after the DSA algorithm is run.

Chapter 7

Capacity of Sensor Network

7.1 Introduction

In this chapter, an analytical model is developed to determine the capacity of a sensor network in terms of outage probability.

7.2 Capacity of sensor network

In the previous chapter an extension of the DSA algorithm for multiple sinks was presented. The extension considers sources routing data to the nearest sinks. The relation between a sink and the number of sensor nodes it can support is presented in this section. As the number of sensor nodes increases, the capacity of the sensor network is limited by the number of sinks present to handle the traffic generated by the sensor nodes. In particular the capacity is limited by the funnel effect of the sensor network. It is assumed that the links between the sinks and central base station have high capacity and do not saturate when the network scales. It is also assumed that a sink is an expensive hardware that is capable of communicating with greater number of sensor nodes compared to a sensor node. For example a sink equipped with directional antennas can handle more traffic than sensor nodes equipped with omni-directional antenna [57, 75]. The amount of traffic that can be transferred to a sink from and by its local sensor nodes may be significantly limited by a number of factors. Particularly, the focus is on the funnel effect experienced by the one-hop neighboring sensor nodes (henceforth called “the one-hop sensor nodes”) of the sink, which experience the greatest relaying burden in the region served by the sink. Again, sensor nodes are assumed to have limited hardware and energy resources. This and channel and medium access conditions (spanning factors such as ambient noise power, fading, shadowing, and interference) collectively limit the number of data flows that a sensor node can relay and its transmission radius. In this section, the physical and data-link/MAC layers are simply parameterized with an upper bound on the number of flows that can be handled by a sensor node, i.e., its relaying capacity, and on its transmission range. The outage probability, i.e., the fraction of sensor nodes that cannot connect to the sink is calculated from the limit on the number of flows one-hop sensor nodes can handle, the sensor node and sink density and the communication radius of a sensor node.

7.3 Problem formulation

The deployment of sensor nodes is modeled as a spatial Poisson process with density λ nodes per unit area. The sink nodes are assumed to be similarly distributed but with a node density of $\sigma < \lambda$ sinks per unit area. The area over which the sinks are distributed can be tessellated to form Poisson Voronoi cells [53] that contain a single sink and the sensor nodes it serves.

Consider a tile A that contains a sink and sensor nodes connecting to the sink. The area of the tile is $|A|$ and the mean number of sensor nodes in the tile is $EN = \lambda|A|$. The one-hop sensor nodes are assumed resident in a region S, a circular disk of radius D, centered at the sink with $S \subset A$. That is, D is the communication radius of a sensor node and note that the assumption $S \subset A$ is reasonable when D or σ are small. Since the one-hop sensor nodes are proposed to be the primary relaying bottlenecks, the mean number of bottleneck nodes is given by:

$$EN_1 = \lambda \pi D^2. \tag{7.1}$$

Figure 7.1 depicts such a region A with S and a sink.

The mean number of nodes served by the N_1 one-hop nodes, i.e. the number of nodes in region $A \setminus S$, that forward traffic to the sink through one of the one-hop

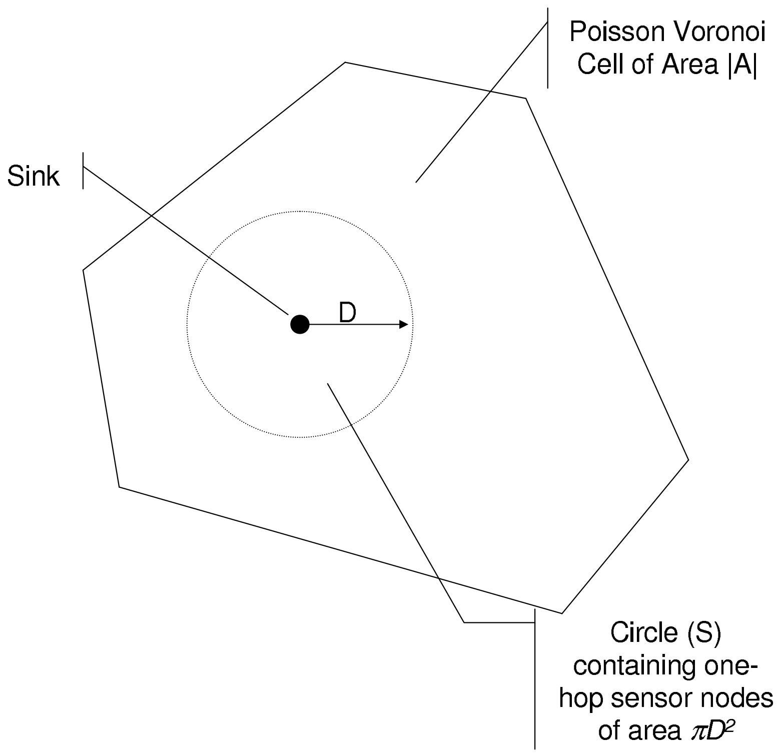


Figure 7.1. A Poisson Voronoi Cell with regions A and S

nodes is given by

$$\begin{aligned} EN_2 &= E(N - N_1) \\ &= \lambda|A| - \lambda\pi D^2 \\ &= \lambda(|A| - \pi D^2) \end{aligned} \tag{7.2}$$

Now let F be the number of external flows that a sensor node can support (i.e., not including its own). Recall that F depends on D: a larger D increases the contention region of the one-hop sensor nodes and thereby reduces F. Here it is implicitly assumed that the one-hop sensor nodes support equal amounts (F) of sensor nodes in region $A \setminus S$, i.e., each one-hop node supports N_2/N_1 other sensor nodes. Again, N_1 is the number of one-hop sensor nodes and N_2 is the number of non one-hop sensor nodes. This assumption is equivalent to assuming that the routing algorithm balances the flows equally between all one-hop sensor node. Such a routing algorithm becomes a necessity if a comparable energy consumption rate is to be maintained among the one-hop sensor nodes. Note, since the death of a one-hope sensor nodes can cause network breakage, any premature demise of a one-hope sensor node breaks the network. To ensure longer lifetime of the network, it is necessary that all one-hop sensor nodes expend energy at comparable rates.

Let Ω be the *outage* defined as the fraction of the sensor nodes in the tile that cannot connect to the sink. The outage is approximated by

$$\Omega \equiv \frac{N - N_1 F - N_1}{N} \tag{7.3}$$

where $N_1 F$ represents the *maximum* number of sensor nodes in region $A \setminus S$ that are supported by the one-hop sensor nodes. The additional N_1 term denotes the flows represented by the one-hop sensor nodes themselves.

The outage in equation (7.3) does not include the outage due to relaying limits of sensor nodes that are non one-hop neighbors of the sink nor of isolated sensor nodes that cannot reach any the sink. The latter is determined by the connectivity of the sensor network, which depends on the sensor node density λ and communication radius (transmission range) D. If a constant mean sensor node density λ is assumed, the number of isolated sensor nodes will obviously be reduced as D increases. If $\pi D^2 \approx 1/\sigma$ then the entire tile is covered by the sink, i.e., all the sensor nodes are the one-hop neighbors of the sink, but this increases channel contention. Therefore,

$$\frac{1}{\lambda} \ll \pi D^2 \ll \frac{1}{\sigma} \tag{7.4}$$

so as to be able to reasonably assume a largely connected sensor network with manageable channel contention. Note that for large λ sensor nodes in region $A \setminus S$ can reach the flow limit and cause outage, a scenario that is, again, not considered in equation (7.3).

Now, calculation of $P(\Omega > \varepsilon)$ determines the probability that the outage exceeds a lower bound (an outage tolerance) ε.

$$P(\Omega > \varepsilon) \;=\; P\left(\frac{N - N_1(F+1)}{N} > \varepsilon\right)$$

$$\begin{aligned} &= P\left(\frac{N_1 + N_2 - N_1 F - N_1}{N_1 + N_2} > \varepsilon\right) \\ &= P\left(N_2 > \frac{N_1(F+\varepsilon)}{1-\varepsilon}\right) \\ &= P\left(N_2 > \alpha N_1\right) \qquad (7.5) \end{aligned}$$

where $\alpha \equiv \frac{F+\varepsilon}{1-\varepsilon}$. Given $|A|$, note that N_2 and N_1 are Poisson distributed with mean $\lambda(|A| - \pi D^2)$ and $\lambda|A|$ respectively. Conditioning on the cell's area

$$|A| \equiv \mathcal{A} > \pi D^2,$$

equation becomes

$$P(\Omega > \varepsilon) = \int_{\pi D^2}^{\infty} P\left(N_2 > \alpha N_1 | \mathcal{A} = a\right) f_{\mathcal{A}}(a) \mathrm{d}a \qquad (7.6)$$

where $f_{\mathcal{A}}(a)$ is the distribution of the area of a Poisson Voronoi cell. Now,

$$\begin{aligned} P(N_2 > \alpha N_1 | \mathcal{A} = a) &= \sum_{n_1=0}^{\infty} P(N_2 > \alpha N_1 | N_1 = n_1, \mathcal{A} = a) \\ &\quad \times \frac{e^{-\lambda \pi D^2} \ (\lambda \pi D^2)^{n_1}}{n_1!} \\ &= \sum_{n_1=0}^{\infty} \sum_{n_2=\alpha n_1}^{\infty} \frac{e^{-\lambda(a-\pi D^2)} \ (\lambda(a-\pi D^2))^{n_2}}{n_2!} \\ &\quad \times \frac{e^{-\lambda \pi D^2}(\lambda \pi D^2)^{n_1}}{n_1!} \qquad (7.7) \end{aligned}$$

Thus,

$$\begin{aligned} P(\Omega > \varepsilon) &= \int_{\pi D^2}^{\infty} \sum_{n_1=0}^{\infty} \sum_{n_2=\alpha n_1}^{\infty} \frac{e^{-\lambda a}(\lambda \pi D^2)^{n_1}}{n_1!} \\ &\quad \times \frac{(\lambda(a-\pi D^2))^{n_2}}{n_2!} \ f_{\mathcal{A}}(a) \ \mathrm{d}a. \qquad (7.8) \end{aligned}$$

An approximate distribution for area $f_{\mathcal{A}}$, given in [53], is

$$f_{\mathcal{A}}(a) = \frac{b^q a^{q-1} e^{-ba}}{\Gamma(q)} \tag{7.9}$$

where $b = 5.38\sigma$ and $q = 5.38$. Equation 7.9 is used to numerically compute the outage probability given by equation (7.8). Figures 8.1-8.3 depict the outage probability comparison with simulation results for different values of F, D and λ, as described in the section 8.2.

7.4 Capacity with mobile relocated sinks

In the previous section it is assumed that the sensor nodes and sinks are static and distributed according to a spatial Poisson process. Here it is assumed that the sinks are mobile ([2, 64]) and equipped with location information (GPS systems). Note, this is not completely unreasonable since the sinks are limited in number and as mentioned before more expensive hardware. Given that the sinks are mobile, it is desirable to relocate them after their initial deployment to increase sensor network capacity and reduce outage.

Primary motivation for sink relocation is management of equal number of sensor nodes. Since the initial deployment could lead to sinks managing different number of sensor nodes leading to under utilized resources in certain sinks and higher outage probability in other cells.

However equal distribution of sensor nodes is not a trivial task and requires knowledge of location of sensor nodes. One possible solution is to move the sinks towards the relay node with the maximum number of flows. This could potentially increase the number of one-hop nodes and increase the capacity of the one-hop nodes to handle more flows in the directoin that has the maximum number of flows. It is assumed that the one-hop node with the maximum number of flows is more likely to have additional flows it could not accomodate. If there are multiple one-hop nodes relaying the maximum number of flows, a target one-hop sensor node is chosen at random. One of the disadvantage with this method is that the determination of the direction of the relay with maximum number of flows might not be trivial. Another disadvantage is that the sink could disconnect from one-hop nodes in the opposite direction of the move and potentially, break the network. Also it is not certain that the move would reduce the outage probability in the direction of move.

An alternate mobility strategy considered in this thesis is relocation of sinks to manage equal *area* of the sensing region instead of equal number of sensor nodes. Although this does not guarantee equal distribution of sensor nodes among the sinks, it achieves better sensor nodes distribution than the original deployment, which is modeled as a spatial Poisson process. The implementation of the mobility strategy is simpler compared to the previous mobility strategy since the sinks are aware of their location and the central base station can coordinate their moves to distributed the sinks equally over the sensing region.

It is assumed that the central base station coordinates the relocation of the sinks and the sensing region is tesselated into squares with a sink at the center of a square.

Now, the affect of relocation of the sinks on the outage probability (equation 7.3) is examined. The modified analytical model assumes that the region A is a square of area $|A|$ instead of a Poisson Voronoi cell. Note the calculations would also hold for other shapes (e.g. hexagon or circles). As in section 7.3 the following are assumed:

- The sensor nodes are distributed according to a spatial poisson process with density λ.
- The one-hop sensor nodes are the bottleneck
- Each sensor node can forward upto F flows
- The one-hop sensor nodes forward equal number of flows from the non one-hop sensor nodes
- Communication radius of a sensor node is given by D.

From the assumptions, the mean number of sensor nodes in region A is given by $EN = \lambda|A|$. The mean number of one-hop sensor nodes, in region S, is given by $EN_1 = \lambda\pi D^2$ and the mean number of non one-hop sensor nodes, in region $A \setminus S$, is given by $EN_2 = \lambda(|A| - \pi D^2)$. Figure 7.2 shows an example of region A

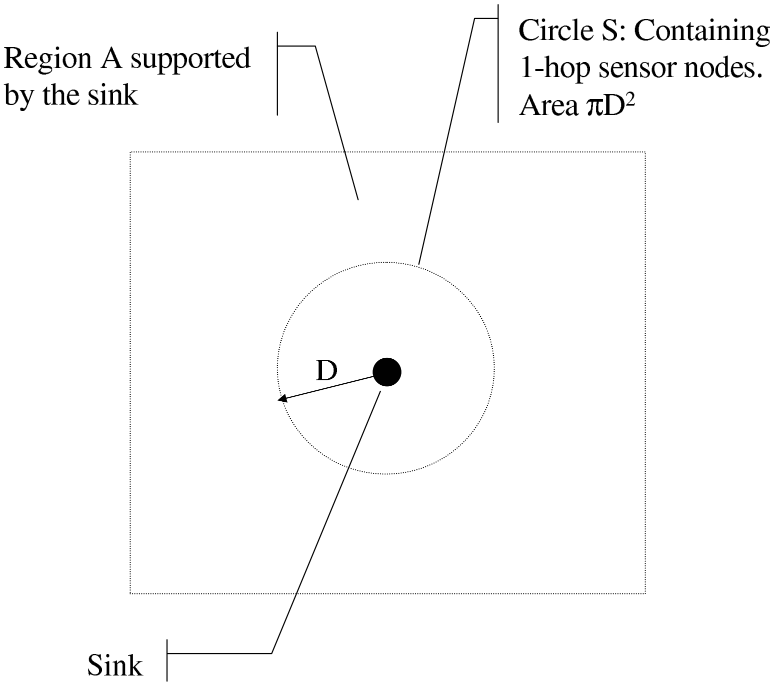

Region A supported by the sink
Circle S: Containing 1-hop sensor nodes. Area πD²
D
Sink

Figure 7.2. Deterministic region of support created by mobile sink

with one-hop sensor node region, S, and the sink at the center. Outage is defined as the number of sensor nodes unable to communicate with the sink due to the flow limit and outage probability is defined by equation 7.5.

To ensure $S \subset A$, and the sensor network is connected, it is assumed that the communication radius satisfies condition

$$\frac{1}{\lambda} \ll \pi D^2 \ll |A| \tag{7.10}$$

If $S \approx |A|$, then all sensor nodes are one-hop sensor nodes that can increase channel contention and if $\pi D^2 \approx 1/\lambda$, then the sensor network is disconnected.

From equation 7.5

$$\begin{aligned} P(\Omega > \varepsilon) &= P(N_2 > \alpha N_1) \\ &= \sum_{n_1=0}^{\infty} \frac{P(N_2 > \alpha N_1 | N_1 = n_1) e^{\lambda \pi D^2} \ (\lambda \pi D^2)^{n_1}}{n_1!} \end{aligned} \tag{7.11}$$

and

$$
\begin{aligned}
P(N_2 > \alpha N_1 \quad | \quad N_2 = n_2, N_1 = n_1) \; &= \\
&= \sum_{n_2=\alpha n_1}^{n_2=\infty} \frac{e^{\lambda(|A|-\pi D^2)} \; (\lambda(|A| - \pi D^2))^{n_2}}{n_2!} \qquad (7.12)
\end{aligned}
$$

Thus,

$$
\begin{aligned}
P(\Omega > \varepsilon) \;\; &= \;\; \sum_{n_1=0}^{\infty} \sum_{n_2=\alpha n_1}^{\infty} \frac{e^{\lambda \pi D^2} \; (\lambda \pi D^2)^{n_1}}{n_1!} \\
&\qquad \frac{e^{\lambda(|A|-\pi D^2)} \; (\lambda(|A| - \pi D^2))^{n_2}}{n_2!} \qquad (7.13)
\end{aligned}
$$

The numerical computation and comparison with simulation results are presented in figures 8.4 - 8.6.

Chapter 8

Simulation for Capacity problem

8.1 Simulation Setup

A customized program written in Java and available upon request is developed to perform the simulation. Data structures presented in [53] were used to generate instances of Poisson Voronoi cells (PVC) via the method in [26]. Having created a random PVC, with the sink at the origin, a sensor network is generated consisting of sensor nodes located inside the PVC. A routing algorithm is executed to connect all sensor nodes to the sink. In deriving (7.8), it is assumed that the one-hop sensor nodes serve equal numbers of non one-hop sensor nodes. To approximate this assumption, the sensor nodes use a customized distributed Bellman Ford routing algorithm [18], where the metric for computing routes is the number of sensor nodes supported by the upstream one-hop sensor node. If a sensor node has a choice between two neighbors for the best route, the sensor node chooses the neighbor whose upstream, one-hop sensor node currently relays the minimum number of flows. This can be implemented in an ad hoc network using an AODV [56] based routing algorithm. Route request packets are forwarded all the way to the sink. The sink receives the route request from the one-hop sensor nodes and replies to the route request such that there are an equal number of sensor nodes (surveillance data flows) supported by the one-hop sensor nodes. Since sensor nodes are assumed static, it is also assumed that route discovery does not occur very often. Thus, the overhead for the customized routing algorithm would not be severe. The routing algorithm *attempts* to balance flow among the one-hop sensor nodes, but there are instances of sensor node deployment where, e.g., it is just not possible for nodes to connect to more than one one-hop node. Such cases result in a skewed distribution of flows among the one-hop nodes.

In the preliminary simulations, two potential outage phenomena were evaluated that could be experienced by a sensor node. One wherein the sensor node is not connected to the network (i.e., it is isolated by communication range limitation) and the other where the upstream one-hop node reaches its maximum number of flow limit. The simulation is repeated of a single PVC instance with 25 instances of randomly generated sensor network to estimate $P(\Omega > \varepsilon)$ for that PVC instance. This is repeated for 10 instances of randomly generated PVCs to estimate $P(\Omega > \varepsilon)$. Vertical bars on $P(\Omega > \varepsilon)$ estimates depicted in figures 8.1-8.3 represent confidence intervals of 95%.

8.2 Simulation Results without sink mobility

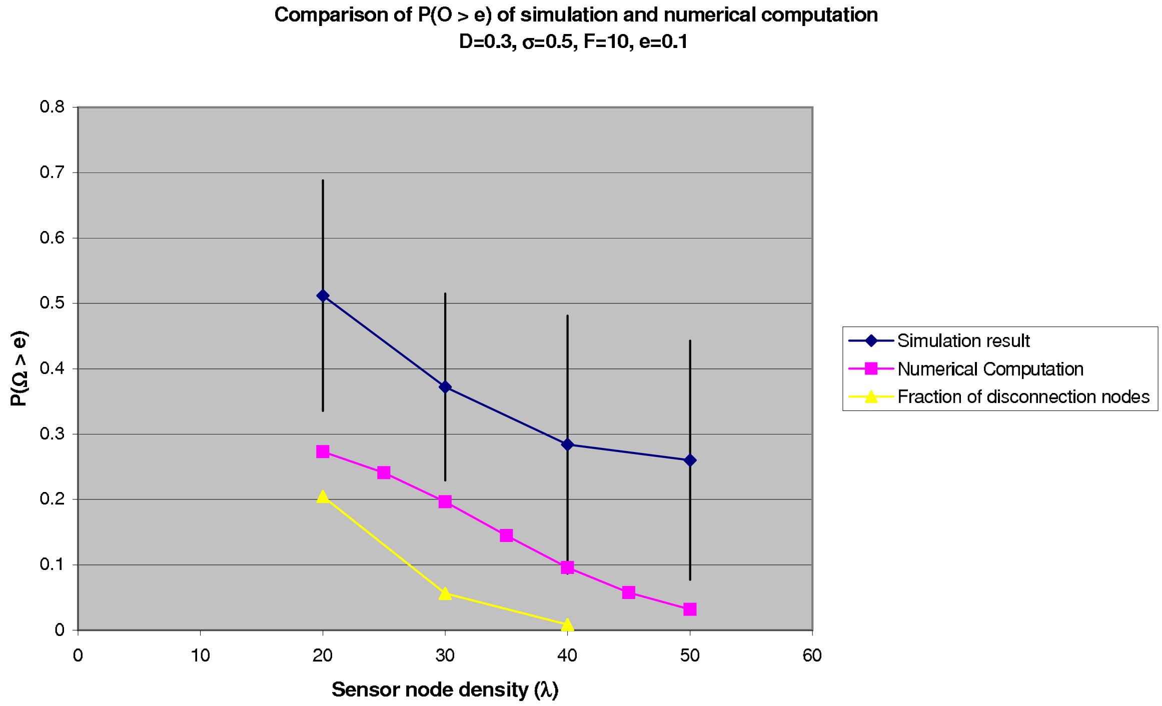


Figure 8.1. Comparison of $P(\Omega > \varepsilon)$ for different λ

Figure 8.1 compares $P(\Omega > \varepsilon)$ for different values of λ for the simulation results and numerical evaluation of equation (7.8). The figure also plots the fraction of nodes that are disconnected. This plot is necessary to determine the accuracy of the outage probability. For a sensor network that is largely disconnected, i.e., has a large fraction of nodes unable to connect to the sink, the $P(\Omega > \varepsilon)$ estimate is not accurate. In figure 8.1, $P(\Omega > \varepsilon)$ for $\lambda = 20$ is an inaccurate estimate, as the fraction of disconnected nodes is high (about 20%). For values of $\lambda > 30$, the fraction of disconnected nodes is negligible and the $P(\Omega > \varepsilon)$ estimate is more accurate. Due to the assumption of equal distribution of flows among the one-hop nodes in (7.8), simulation results do not match the numerical computation of (7.8). However, note the lower range of confidence interval of $P(\Omega > \varepsilon)$ is closer to numerical evaluation of (7.8) indicating instances of the sensor network where equal distribution of flows among the one-hop nodes is possible. The numerical evaluation of equation (7.8) is a lower bound of the outage probability. Also the simulation calculates the average outage probability and the confidence interval for multiple PVC instances. However, the confidence interval includes the average over multiple sensor node deployment for a single PVC instance and does not capture the variance of the outage probability from multiple sensor network instances.

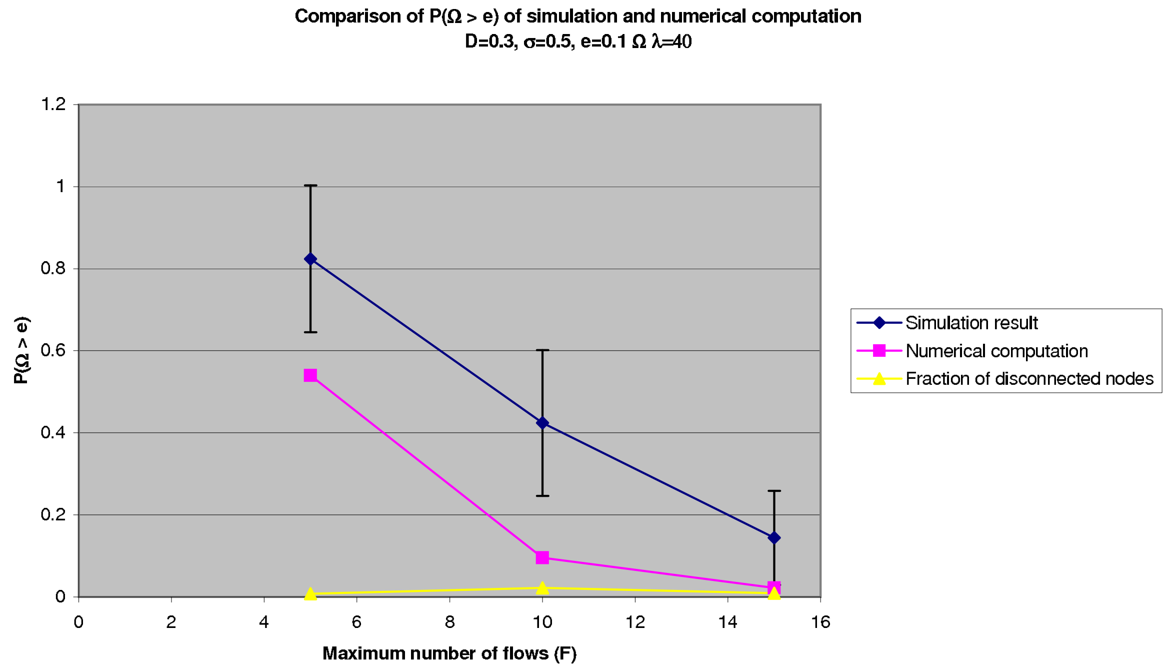


Figure 8.2. Comparison of $P(\Omega > \varepsilon)$ for different F

Figure 8.2 compares $P(\Omega > \varepsilon)$ for different values of F. Note that the fraction of disconnected nodes are negligible for all values of F. Recall the connectivity of the sensor network depends on λ and D and they are suitable choosen to ensure network connectivity. Again due to the assumption of equal flows among the one-hop sensor nodes in (7.8), the simulation result and numerical evaluation of (7.8) do not match, but this difference naturally reduces as the flow limit F increases allowing greater balance.

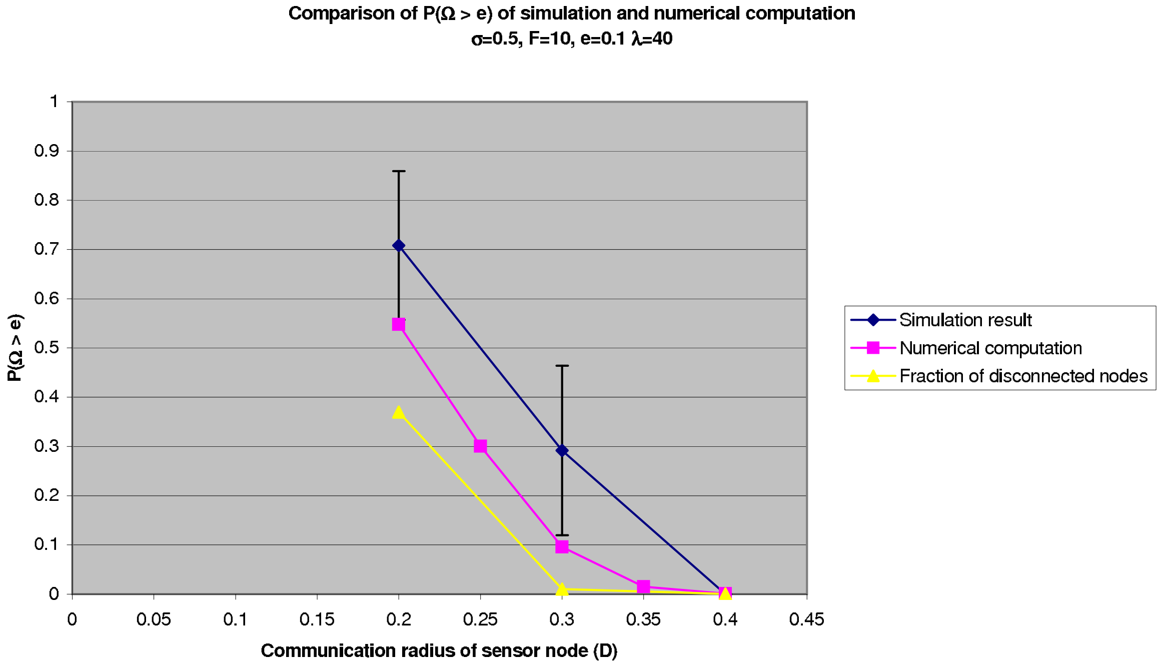


Figure 8.3. Comparison of $P(\Omega > \varepsilon)$ for different D

Figure 8.3 compares $P(\Omega > \varepsilon)$ for different values of D. For $D = 0.2$, the fraction of disconnected nodes is high (about 35%), but for larger values of D this fraction reduces. Note again there is a difference between the simulation estimate and numerical evaluation of $P(\Omega > \varepsilon)$ and this difference reduces significantly for larger values of D (≥ 0.3). This is because for larger values of D, the connectivity of network increases, thus creating routes to alternate one-hop nodes to better balance the flows among the one-hop nodes.

8.3 Simulation results with sink mobility

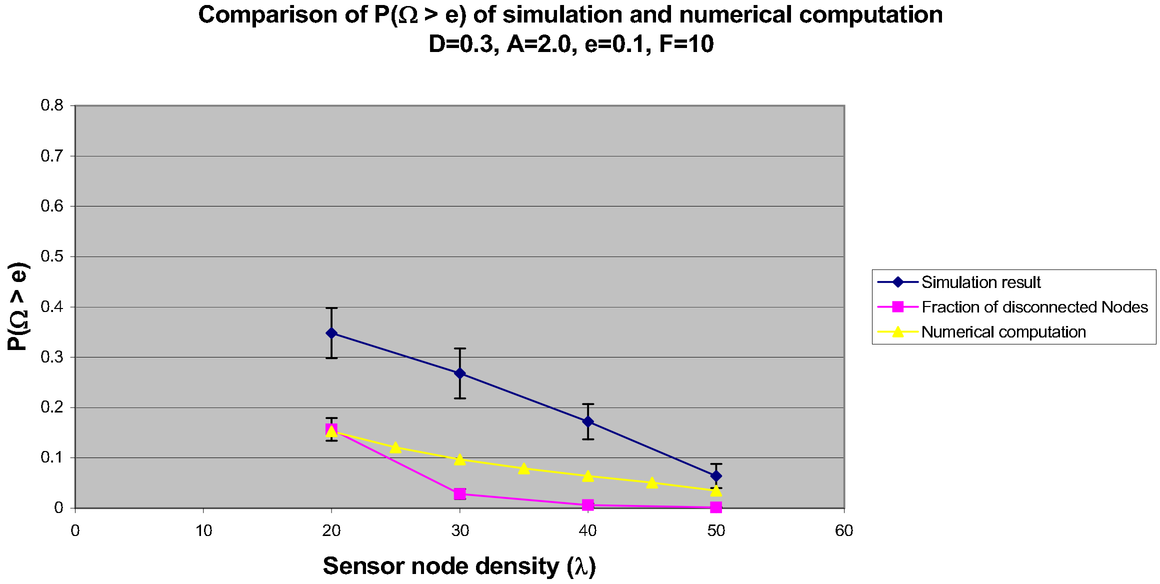


Figure 8.4. Comparison of $P(\Omega > \varepsilon)$ for different λ with deterministic sink placement

The simulation results with relocated sink is different from the random placement of sink, in that, the outage probability are better for both simulation and numerical calculations. Although the simulations results are "closer" to the numerical results for deterministic sink placement than for random sink placement, the results are still not close for numerical calculations to substitute simulation results.

Figure 8.4 shows the comparison of $P(\Omega > \varepsilon)$ from simulation and numerical computation using equation 7.13 for different values of λ. The graph also plots the fraction of disconnected node to ensure a valid comparison between the simulation and numerical results. The two plots show a significant difference between the simulated results and the numerical computation. The difference is mainly due to the assumption of equal distribution of sensor flows among the one-hop sensor nodes. However as the sensor node density increases, there is a decrease in the difference. Though the numerical computation is an inaccurate estimation for the simulation results, it serves as a lower bound on the outage achievable with given number of sensor nodes and one-hop sensor nodes.

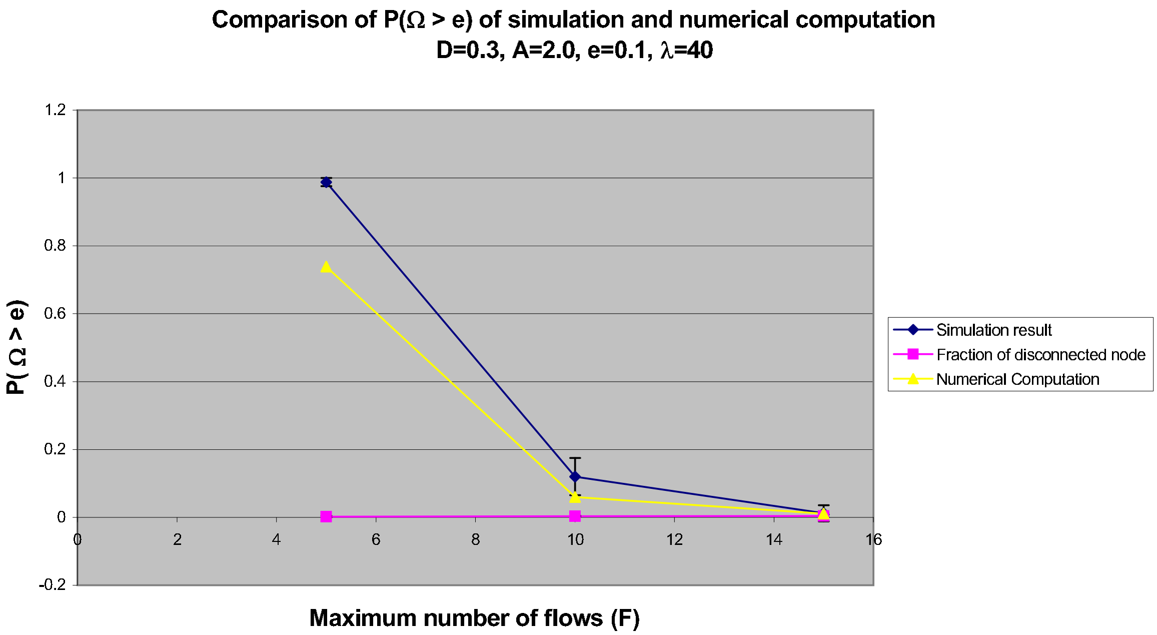


Figure 8.5. Comparison of $P(\Omega > \varepsilon)$ for different F with deterministic sink placement

Figure 8.5 compares the outage probability for different values of maximum flow limit F. The number of disconnected sensor nodes are negligible, as the sensor node density and communication radius ensure a connected sensor network. Again for lower values of F, the simulation results and numerical results are not similar. However as F increases the simulation results and numerical results converge.

Figure 8.6 shows the comparison of outage probability for different D. As the D value increases, the network connectivity increases. Again for lower values of D the simulated and numerical results are not similar, but as D increases, the two results converge.

The graphs in figures 8.1-8.6 (both simulation and numerical computation) show that the outage probability depends more strongly on communication radius D and flow limit F than on the sensor node density λ. This suggests that the communication radius has a greater influence on outage probability.

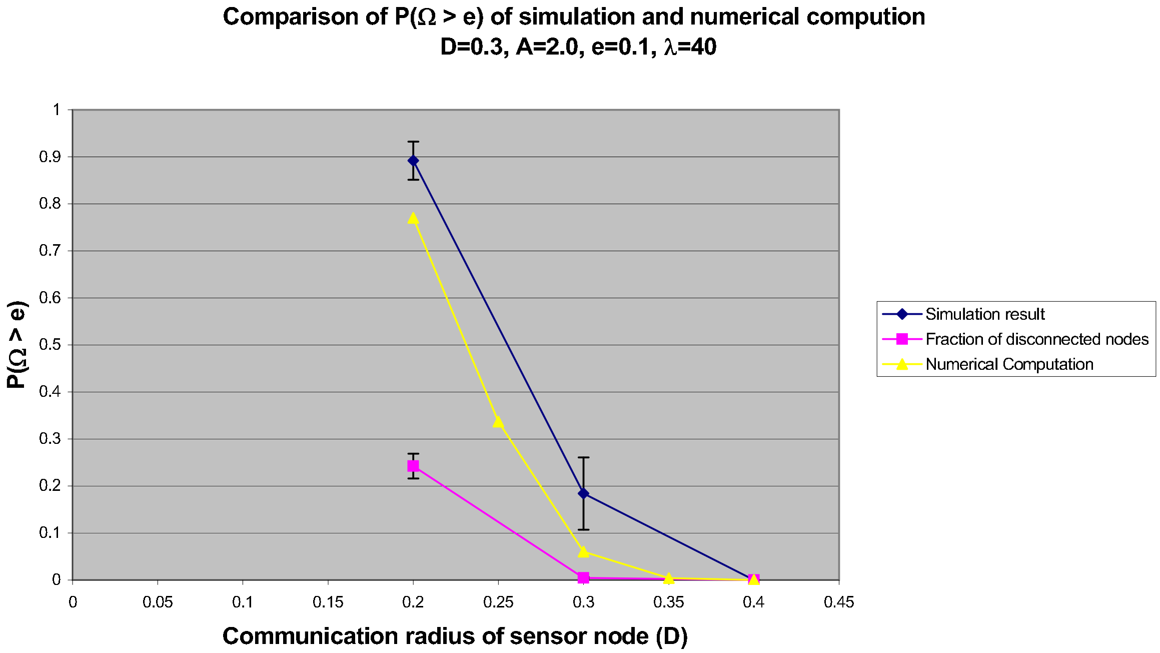


Figure 8.6. Comparison of $P(\Omega > \varepsilon)$ for different D with deterministic sink placement

Chapter 9

Conclusions and Future Work

In this chapter the conclusions and future work are presented

9.1 Summary and future work for Purposeful Mobility

In the thesis the effect of purposeful mobility and capacity issues on sensor networks were studies. The effect of mobility on a sensor network was studied and *purposeful mobility* was presented as a design paradigm. Particularly a Distributed Simulated Annealing (DSA) algorithm to move sensor nodes acting as relays to optimal position for transmission energy conservation was presented. The DSA algorithm measures the tradeoff of mobility with amortized savings in transmission energy before deciding on any moves. Analytical proof were developed that showed the DSA algorithm converged to global optima and the results were validated with simulations. Also the affect of anycast based routing algorithm on performance of DSA was studied.

The simulataneous task of a relay to move for transmission power conservation and surveillance was also examined. Also a metric to quantify surveillance was developed and the affect the affect of mobility on both surveillance and transmission power conservation was studied.

Simulation results to validate the hypothesis that target detection time in a bounded domain can be replicated by target detection time in an unbounded domain with tiled target placement was also presented.

Future work plans include a more comprehensive mobility strategy for sensor network that spans all the layers of the wireless network stack. Current mobility strategy only considered network layer benefits but ignored its affect on Application, MAC and PHY layers. The future work would include a study of the effect of purposeful mobility on all layers and develop metrics such that mobility does not adversely affect any particular layer.

Also, there are plans to look at mobility for other tasks, such as equal energy distribution in the sensing region. If energy depletion is occurring more rapidly in certain regions, movement of sensor nodes to that region would help in reducing the energy depletion rate.

9.2 Conclusions and future work for capacity issues in sensor network

The thesis also looked at capacity related issues in sensor network and in particular determined capacity in terms of the number of sensor nodes a given sink can support with acceptable outages. It was assumed that the immediate neighbors of the sinks, the one-hop sensor nodes, handle the maximum amount of traffic. *Outage* was defined as the fraction of sensor nodes that cannot connect to the sink due to the capacity limit reached by the one-hop sensor nodes.

The sensor nodes and sink were modeled as Spatial Poisson process and analytical results that estimated the outage probability exceeding certain limit were developed. Analytical model for outage calculation for relocated sinks to the center of square region was also presented. And in both cases the outage calculations from the analytical results were compared to outage calculations from simulation results.

Although the comparisons did not yield a close match, it helped us develop a lower bound on the best outage possible. The graphs also showed the affect of individual parameters on outage performance and it was concluded that increases in communication radius and flow limit improved the outage probability more than increases in sensor node density.

Note that outage probability can be improved with increasing the communication radius, D, of the sensor nodes, however it fails to show that F is also dependent on D. Note, an increase in communication radius increases the neighbors of a sensor node and hence increases the channel contention. The future work plans to include a study that replaces the simplistic limit F with a more representative model of the MAC layer ([10]) to determine forwarding limits in terms of contentions at MAC layer.

Lastly a study of the effect of purposeful mobility on capacity of sensor network is planned in the future work. DSA algorithm moved relays to create a more "linear" route between the sources and sinks, thus reducing the channel contention. The future work plans to look at ways to use purposeful mobility to increase capacity in more direct ways.

Appendix A

Simulated Annealing: A brief introduction

We present a brief introduction to simulated annealing.

A.1 Introduction

Simulated Annealing is a non-linear optimization technique. It is considered as a *meta-heuristic* optimization technique. The simulated annealing algorithm "searches" the solution space, $\mathcal{S}$ for the global optima. Given a starting point in the solution space $\mathcal{S}$, the simulated annealing algorithm selects the next solution based on certain criterion. The algorithm progresses by moving from one point in $\mathcal{S}$ to another till it converges to the global optima.

A.1.1 Next Neighbor Selection Criterion

Let $C(x)$ represent the cost function we are trying to minimize over $x \in \mathcal{S}$. Currently let us assume that $x = x_i$. The simulated annealing algorithm chooses a point $x_j \in \mathcal{S}$ from a neighborhood set $\mathcal{N} \subset \mathcal{S}$ and decides to move to x_j with probability given by equation A.1.

$$A_T(x_j|x_i) = \begin{cases} 1 & if \quad C(x_j) < C(x_i) \\ \exp(\frac{C(x_i)-C(x_j)}{T}) & if \quad C(x_j) \geq C(x_i) \end{cases} \tag{A.1}$$

This can also be written as

$$A_T(x_j|x_i) = min\left\{1, \exp\left(\frac{-(C(x_j) - C(x_i))}{T}\right)\right\} \tag{A.2}$$

Equation A.1 is the *Metropolis* criterion and the algorithm based on this criterion is called the Metropolis algorithm. According to the Metropolis criterion, the Simulated Annealing algorithm moves from state x_i to state x_j if it is a “good” move, i.e. if $C(x_j) < C(x_i)$. If the move is a “bad” move, i.e. $C(x_j) \geq C(x_i)$ the Simulated Algorithm moves to x_j with probability $\exp((C(x_i) - C(x_j)/T)$. The probability of accepting a “bad” move is controlled by the parameter T, the *temperature*. Higher the temperature, greater the probability of accepting a “bad” move and greater the probability of “climbing out” of local optima. The Simulated Annealing algorithm starts with a high T and reduces T, i.e., cools the temperature, very slowly, as the algorithm proceeds to achieve global optima.

A.1.2 Markov Chain model and Gibbs Distribution

The neighbor selection is modeled as a Markov chain according to the transition probability matrix given by $Q(X_i, X_j)$. The TPM is assumed aperiodic, irreducible, time-reversible with stationary distribution μ. Since $Q(X_i, X_j)$ is time-reversible, detailed balance equation holds, i.e. $\mu(x_i)Q(x_i, x_j) = \mu(x_j)Q(x_j, x_i)$.

The transition probability matrix for the Simulated Annealing algorithm is given by

$$P_T(x_i, x_j) = A_T(x_j|x_i)Q(x_i, x_j) \qquad \forall x_i, x_j \in \mathcal{S} \tag{A.3}$$

$P_T(X_i, X_j)$ inherits aperiodicity and irreducibility and is time-reversible with Gibbs stationary distribution give by equation (A.4).

$$\pi_T(x_i) = \frac{\mu(x_i)e^{-C(x_i)/T}}{Z} \tag{A.4}$$

where $Z = \sum_{\forall x_k \in \mathcal{S}} \mu(x_i)e^{C(x_k)/T}$. Note the T subscript denotes that the TPM is valid only for a particular value of T.

To prove that equation (A.4) is the stationary distribution for Markov chain

$P(X_i, X_j)$, simply note that equation (A.4) satisfies the detailed balance equations:

$$\begin{aligned} \pi(x_i)P_T(x_i,x_j) &= \pi(x_j)P_T(x_j,x_j) \quad \forall x_i, x_j \in \mathcal{S}. \\ \pi(x_i)A_T(x_i,x_j)Q(x_i,x_j) &= \pi(x_j)A_T(x_j,x_i)Q(x_j,x_i) \end{aligned} \tag{A.5}$$

To this end, note

$$\begin{aligned} \pi(x_i)A_T(x_i,x_j) &= \frac{\mu(x_i)exp(-C(x_i)/T)}{Z} \;\; min\left\{1, exp\left(\frac{-(C(x_j)-C(x_i))}{T}\right)\right\} \\ &= \frac{\mu(x_i)exp(-C(x_i)/T)}{Z} \;\; min\left\{1, \frac{exp(-C(x_j)/T)}{exp(-C(x_i)/T)}\right\} \\ &= \frac{\mu(x_i)}{Z} \;\; min\left\{exp(-C(x_j)/T), exp(-C(x_i)/T)\right\} \end{aligned} \tag{A.6}$$

Similarly

$$\pi(x_j)A_T(x_j,x_i) = \frac{\mu(x_j)}{Z} \;\; min\left\{exp(-C(x_i)/T), exp(-C(x_j)/T)\right\} \tag{A.7}$$

Applying equation (A.6) and (A.7) in equation (A.5) we get,

$$\begin{aligned} &\frac{\mu(x_i)}{Z} \;\; min\left\{exp(-C(x_j)/T), exp(-C(x_i)/T)\right\} Q(x_i,x_j) \\ &= \frac{\mu(x_j)}{Z} \;\; min\left\{exp(-C(x_i)/T), exp(-C(x_j)/T)\right\} Q(x_j,x_i) \end{aligned} \tag{A.8}$$

Equation (A.8) simplifies to

$$\mu_T(x_i)Q(x_i,x_j) = \mu(x_j)Q(x_j,x_i) \tag{A.9}$$

which is true since $Q(X_i, X_j)$ is a time-reversible Markov chain.

The stationary distribution of Markov chain represented by TPM $P^T(X_i, X_j)$ is given by equation A.4 for a given temperature T. As the temperature cools, i.e. $T \rightarrow 0$, the steady state converges to the global optima.

Let $X_{opt} \subset \mathcal{S}$ be the set of global optimal solutions. We have

$$\begin{aligned}\lim_{T\to 0}\pi_T(x_i) &= \lim_{T\to 0}\frac{\mu(x_i)e^{-C(x_i)/T}}{Z} \\ &= \lim_{T\to 0}\frac{\mu(x_i)e^{(C(x_{opt})-C(x_i))/T}}{\sum\limits_{\forall x_k\in\mathcal{S}}\mu(x_i)e^{(C(x_{opt})-C(x_k))/T}}\end{aligned}$$

where $x_{opt} \in X_{opt}$

For $x_i \in X_{opt}$ we get

$$\pi(x_i) = \frac{\mu(x_i)}{\sum\limits_{\forall x_k\in X_{opt}}\mu(x_k)} \tag{A.10}$$

and for $x_i \notin X_{opt}$ we get

$$\pi(x_i) = 0 \tag{A.11}$$

which satisfies the proof that $\pi_T(x_i)$ converges to the optimal solution as $T \to 0$.

A.1.3 Inhomogeneous Markov Chain model

The homogenous Markov chain model requires infinite time to reach the Gibbs stationary distribution. The algorithm could require implementing infinitely long sequences of Markov chains with descending values of temperature. Clearly this is not practical. Alternately the algorithm can be modeled as a series of finite length Markov chains, each generated by reducing values of temperature. This leads to a time-inhomogeneous Markov chain which is strongly ergodic and converges in distribution if the cooling schedule is given by $T(k) = \frac{c}{\log(1+k)}$ where $T(k)$ is the temperature at time instance k and c is sufficiently large (to overcome the deepest local minima not the global minima) [32]. That is, the time-inhomogeneous Markov chain is slowly cooled or "annealed" so that its distribution tracks the Gibbs distribution at temperature $T(k)$.

Appendix B

Simulation outline

In this chapter we present a brief outline of the simulation and the data structures used in simulation.

B.1 Simulation outline for Purposeful Mobility

We implement the simulations for DSA algorithm in C++ and ported it later to Java. The simulation has three components:

- Placement of sources, sinks and relays
- Routing algorithm (Distributed Bellman Ford)
- Distributed Simulated Annealing Algorithm

B.1.1 Node placement

The simulation is generated in a square region of area 40m x 40m. The sources and sink are deterministically placed and the relays are randomly placed between the sources and sink.

B.1.2 Routing Algorithm

We then execute the Distributed Bellman Ford routing algorithm to find routes from source nodes to the sink. The routing algorithm consists of three main methods:

- Create routes: In this method the program loops through all the nodes (except the sinks) and identifies the nodes that are neighbors to the sink. The immediate neighbors of the sink update their routing table and generate a message for their neighbor. The message is pushed into a queue.
- Update route: In this method the message queue is popped and the neighbors check if the message update generates an alternate minimum path. If the message shows an alternate minimum path, the neighbor nodes updates its routing table and generates a new message for its neighbor and pushes it into the message queue. After a message is processed, the next message is popped from the queue till the queue is emptied.
- Update flow: This propagates from the source to the sink and populates the data structure representing the neighbors from which a node receives flow. For the DSA algorithm, both the transmission and reception flow information are required.

B.1.3 Distributed Simulated Annealing

The DSA algorithm is implemented as one method. The node executing the DSA algorithm finds the current power consumption in its neighborhood. It generates an array of positions where it can move, assuming that no node exist there. The node randomly chooses a position and with that as its new position calculates the power consumption in its neighborhood. If the new power consumption is better, the node accepts the move. If the new power consumption is worse, the node accepts the new position according the Metropolis Criterion. If the new position if rejected, the node updates the position with the old values.

A brief outline of the class structure with the major attributes and methods is presented in tables [B.1 - B.3]

B.2 Simulation outline for Capacity Results

We have divided our capacity simulation into three major components:

Table B.1. Class structure of MobileNode

MobileNode			
Attributes			
	double	x	X value in position (x,y)
	double	y	Y value in position (x,y)
	int	nodeAddress	int value to uniquely identify node
	int	nodeType	int identifying if node is Source,Sink or Relay
	Vector	upstreamNode	List of nodes transmitting to it
	RoutingTable	routingTable	Routing table for mobile node
Methods			
	double	getLocalTxPower(NodeList)	Gets the transmission power in neighborhood
	double	getTxPower(NodeList)	Power spent in transmission
	double	getRxPower(NodeList)	Power spent in receveing from neighboring nodes
	NodeList	getMyNeighbor(NodeList)	Returns a vector containing the neighboring nodes
		Getter/Setter	The Get and Set methods for all attributes

- Generating the Poisson Voronoi Cell. This step is skipped for the second simulation

- Generating the sensor nodes inside the cell according to a Spatial Poisson Process

- Distributed Bellman Ford routing algorithm

B.2.1 Generating the Poisson Voronoi Cell

We start with an initial point at the origin and generate a Poisson Voronoi Cell around it. We generate the neighboring points according to Spatial Poisson process of density σ and create the Poisson Voronoi Cell from the perpendicular bisectors of the line joining the origin to the neighboring points. We start with two random

Table B.2. Class structure of RoutingTable

RoutingTable			
Attributes			
	double	routeEntryList	List of RouteEntry values in routing table
Methods			
	double	getBestRouteToSink()	Returns the RouteEntry with lowest route cost to a sink
	RouteEntry	getEntryForNodeID(int)	Returns the routeEntry for Destination node given by int
	double	clearAllRoutes()	Deletes all route entry in routing table
	NodeList	addRouteEntry(RouteEntry)	Adds a new RouteEntry to routing table
	boolean	isRoutingEntryPresent(int)	Checks if a RouteEntry is present for destination node given by int
		Getter/Setter	The Get and Set methods for all attributes

variables, Q_i an exponentially distributed random variable of density 1 and θ_i a random variable uniformly distributed in range $(0, 2\pi)$. $R = Q_0 + Q_1 + .. + Q_{i-1} + Q_i$ represents the radial distance of the neighboring point generated after i^{th} interation. θ_i represents the angle of the neighboring point generated after i^{th} iteration. We get the Spatial Poisson Process of density σ multiplying the radial distance of the unit density Spatial Poisson Process by a factor of $\sqrt{Q_i/\pi\sigma}$. The neighboring points are represented by $(X, Y) = (\sqrt{Q_i/\pi\sigma}\ cos\theta, \sqrt{Q_i/\pi\sigma}\ sin\theta)$. We generate the Poisson Voronoi Cell by drawing the perpendicular bisectors of the line joining the origin and the neighboring points. We stop the simulation when the new neighboring point does not alter the dimension of the Poisson Voronoi Cell, i.e., the perpendicular bisector from the new Poisson point is beyond the farthest vertex of the Poisson Voronoi cell.

Table B.3. Class structure of RouteEntry

RouteEntry			
Attributes			
	int	destinationID	Node ID of the destination node
	int	nextHopID	Node ID of the next hop node
	int	numberOfFlows	Number of flows in current routeEntry
	int	numberOfHops	Number of hops to destination in current routeEntry
	int	routeCost	Route cost to destination node
Methods			
		Getter/Setter	The Get and Set methods for all attributes

B.2.2 Generating the sensor nodes

To generate the sensor nodes position, we generate a poisson random variable according to density $\lambda|A|$ that represents the number of sensor nodes in area $|A|$. The region considered is much larger that the Poisson Voronoi Cell. The locations of the sensor nodes are generated by generation uniformly distributed random variables for their X and Y coordinates. Sensor nodes that are not inside the Poisson Voronoi Cell are discarded.

B.2.3 Distributed Bellman Ford

The routing algorithm is similar to the one implemented for DSA simulations. The only change is that the route metric used is the number of flows managed by the upstream one-hop sensor nodes.

Bibliography

[1] Arup Acharya, Archan Misra, and Sorav Bansal. MACA-P: A MAC for concurrent transmissions in multi-hop wireless networks. In *Proceedings of the First International Conference on Pervasive Computing and Communication (PerCom)*, pages 505–508, Mar 2003.

[2] Kemal Akkaya, Mohamed Younis, and Meenakshi Bangad. Sink repositioning for enhanced performance in wireless sensor networks. *Elsevier Computer Network Journal*, Vol 49(No. 4):512–534, November 2005.

[3] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *IEEE Communication Magazine*, pages pp. 102–114, Aug 2002.

[4] I. F. Akyilidiz, W. Su, Y. Subramanian, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38:393–422, Dec 2002.

[5] Francois Baccelli and Charles Bordenave. The radial spanning tree of a Poisson point process. In *The 43rd Annual Allerton Conference on Communication, Control, and Computing*, Sept, 2005.

[6] Francois Baccelli and Sergei Zuyev. Poisson-Voronoi spanning trees with applications to the optimization of communication networks. *Oper. Res.*, 47(4):619–631, 1999.

[7] Paramvir Bahl, Ranveer Chandra, and John Dunagan. SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 216–230, 2004.

[8] Nikhil Bansal and Zhen Liu. Capacity, delay and mobility in wireless ad-hoc networks. In *The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Mar 2003.

[9] Samrat Bhattacharjee, Mostafa H. Ammar, Ellen W. Zegura, Viren Shah, and Zongming Fei. Application-layer anycasting. In *Inproceedings of IEEE Conference of Computer Communication*, pages 1388–1396, 1997.

[10] Guiseppe Bianchi. Performance analysis of the IEEE 802.11 distributed coordinate function. *IEEE Journal on Selected Areas in Communications, Wireless Series*, pages 535–547, Mar 2000.

[11] A.N. Borodin and P. Salminen. *Handbook of Brownian Motion - Facts and Formulae*. Birkhauser, Boston, 1996.

[12] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Mobile Computing and Networking (MobiCom)*, pages 85–97, 1998.

[13] Tracy Camps, Jeff Belong, and Vanessa Davies. Mobility models for ad hoc network simulations. In *Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2002.

[14] Vijay Chandramohan and Ken Christensen. A first look at wired sensor networks for video surveillance. In *Proceeding of the 27th Annual IEEE Conference on Local Computer Networks*, pages 728–729, 2002.

[15] Dazhi Chen and Pramod K. Varshney. Qos support in wireless sensor networks: A survey. In *Proceedings of International Conference of Wireless Networks*, pages 227–233, 2004.

[16] Shigang Chen and Klara Nahrstedt. On finding multi-constrained paths. In *Inproceedings of IEEE International Conference on Communication (ICC)*, volume Vol 2, pages 874 – 879, June 1998.

[17] Chee-Yee Chong and Srikanta P. Kumar. Sensor networks: Evolution, opportunities, and challenges. In *Proceedings of the IEEE*, volume 91 no. 8, pages 1247–1256, Aug 2003.

[18] T. Cormen, C. Leiserson, and R. Rivest. Introduction to Algorithms. *MIT Press*, 1990.

[19] Renato M. de Moraes, Hamid R. Sadjadpour, and J. J. Garcia-Luna-Aceves. On mobility-capacity-delay trade-off in wireless ad hoc networks. In *12th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS'04)*, pages 12–19, 2004.

[20] E. J. Duarte-Melo and M. Liu. Data-gathering wireless sensor networks: organization and capacity. *Computer Networks (COMNET) Special Issue on Wireless Sensor Networks*, 16 no. 5:519–537, 2003.

[21] Arjan Durresi, Vamsi Paruchuri, and Leonard Barolli. Delay-energy aware protocol for sensor and actor networks. In *Proceedings of Parallel and Distributed Systems*, volume Vol. 1, pages 292–298, 2005.

[22] Emile H. L. Aarts and Jan Korst. *Simulated Annealing and Boltzmann Machines*. John Wiley and Sons, 1989.

[23] W. Feller. *An Introduction to Probability Theory and its Applications*. Wiley, New York, 1968.

[24] Gang Feng, Christos Douligeris, Kia Makki, and Niki Pissinou. Performance evaluation of delay-constrained least-cost QoS routing algorithms based on linear and nonlinear Lagrange relaxation. In *Inproceeding of IEEE International Conference on Communication*, volume Vol 4., pages 2273 – 2278, Miami, FL, 2002.

[25] Mark Fleischer. Simulated annealing: past, present, and future. In *Proceedings of the 27th conference on Winter simulation*, pages 155–161. ACM Press, 1995.

[26] Andreas Frey and Volker Schmidt. Marked point processes in the plane I: A survey with applications to spatial modeling of communication networks. *Advances in Performance Analysis*, 1(1):65–110, 1998.

[27] Abbal El Gamal, James Mammen, Balaji Prabhakar, and Devavrat Shah. Throughput-delay trade-off in wireless networks. In *The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Mar, 2004.

[28] David K. Goldenberg, Jie Lin, A. Stephen Morse, Brad E. Rosen, and Y. Richard Yang. Towards mobility as a network control primitive. In *Proceedings of MobiHoc*, Japan, Tokyo, May 2004.

[29] M. Grossglauser and D. Tse. Mobility increases capacity in ad-hoc wireless networks. In *Proc. IEEE INFOCOM*, 2001.

[30] P. Gupta and P.R. Kumar. Critical power for asymptotic connectivity in wireless networks. In *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming*, pages 547–556, Boston, ISBN 0-8176-4078-9, 1998. (Editors: W.M. McEneany, G. Yin, and Q. Zhang) Birkhauser.

[31] P. Gupta and P.R. Kumar. The capacity of wireless networks. *IEEE Trans. Info. Theory*, Vol. 46 :No. 2:388–404, March 2000.

[32] Bruce Hayek. Cooling schedules for optimal annealing. *Mathematics of Operations Research*, Vol. 13(2):311–329, May 1988.

[33] N. Heo and P. K. Varshney. A distributed self spreading algorithm for mobile wireless sensor networks. In *IEEE Wireless Communication and Networking*, pages 1597–1602, New Orleans, Louisiana, Mar. 2003.

[34] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Inproceeding of the Sixth Annual International Conference on Mobile Computing and Network (MOBICOM)*, 2000.

[35] Ali Jadbabaie, Jie Lin, and A. Stephen Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transaction on Automatic Control*, Vol. 48:988 – 1001, June 2003.

[36] David Jea, Arun Somasundara, and Mani Srivatsa. Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines. In *Proceedings of IEEE Real Time Systems Symposium (RTSS)*, December 2004.

[37] P. Jeon and G. Kesidis. A pheromone-aided multipath qos routing protocol and its applications in MANETs. *To appear in Journal of Communication Software and Systems*, 2006.

[38] Jangeun Jun and Mihail L. Sichitiu. The nominal capacity of wireless mesh networks. *IEEE Wireless Communication Magazine*, 10(5):8–14, 2003.

[39] Joseph M. Kahn, Randy Howard Katz, and Kristofer S. J. Pister. Mobile networking for smart dust. In *Proceedings of ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MobiCom)*, volume 91 no. 8, Seatle, WA, Aug 1999.

[40] Aman Kansal, Mohammed Rahimi, Deborah Estrin, William J. Kaiser, Gregory J. Pottie, and Mani B. Srivastava. Controlled mobility for sustainable wireless sensor networks. In *Prroceeding of Sensor and Ad Hoc Communication Networks (SECON)*, pages 1–6, Oct 2004.

[41] G. Kesidis, T. Konstantopoulos, and S. Phoha. Surveillance coverage of ad hoc sensor networks under a random mobility strategy. In *Proc. IEEE Sensors*, Toronto, Oct. 2003.

[42] G. Kesidis and E. Wong. Optimal acceptance probability for simulated annealing. *Stochastics and Stochastics Reports*, Vol. 29:pp. 221–226, 1990.

[43] George Kesidis and Rajesh Rao. Purposeful mobility for relaying and surveillance in mobile ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, 3(3):225–232, July-Sept 2004.

[44] Mathew Laibowitz and Joseph A. Paradiso. Parasitic mobility for pervasive sensor networks. In *Proceedings of Third International Conference on Pervasive Computing, PERVASIVE*, 2005.

[45] Benyuan Li, Peter Brass, Olivier Dousse, Philippe Nain, and Don Towseley. Mobility improves coverage of sensor networks. In *The Sixth Annual Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc*, pages 300–308, May 2005.

[46] Jiandong Li, Zygmunt J. Haas, Min Sheng, and Yanhui Chen. Performance evaluation of modified IEEE 802.11 MAC for multi-channel multi-hop ad hoc network. In *17th International Conference on Advanced Information Networking and Applications (AINA)*, pages 312–317, Mar 2003.

[47] Jinyang Li, Charles Blake, Douglas S.J. De Couto, Hu Imm Lee, and Robert Morris. Capacity of ad hoc wireless networks. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 61–69, 2001.

[48] Qun Li, Michael De Rosa, and Daniela Rus. Distributed algorithms for guiding navigation across a sensor network. In *The Annual International Conference on Mobile Computing and Networking (MOBICOM)*, pages 313–325, San Diego, Sept. 2003.

[49] J. G. Lim and S. V. Rao. Mobility enhanced position in ad-hoc networks. In *IEEE Wireless Communication and Networking*, pages 1832–1837, New Orleans, Louisiana, Mar. 2003.

[50] Haiyun Luo, Fan Ye, Jerry Cheng, Songwu Lu, and Lixia Zheng. A two-tier data dissemination model for large-scale wireless sensor networks. In *Inproceedings of the eight Annual International Conference on Mobile Computing and Networking*, pages 148–159, 2002.

[51] M. Mauve, J. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad-hoc networks. *IEEE Network Magazine*, 15(6):30–39, Nov. 2001.

[52] Prashant Mohapatra, Chao Gui, and Jian Lui. Group communications in mobile ad hoc networks. *IEEE Computer*, pages 52–59, Feb 2004.

[53] Atsuyuki Okabe, Barry Boots, and Kokichi Sugihara. *Spatial tessellations: concepts and applications of Voronoi diagrams.* John Wiley & Sons, Inc., New York, NY, USA, 1992.

[54] Vincent D. Park and Joseph P. Macker. Anycast routing for mobile networking. In *Proceedings of the Military Communication Conference (MILCOM)*, volume 1, pages 1–5, Oct 31-Nov 1, 1999.

[55] C. Patridge, T. Mendez, and W. Milliken. Host anycasting service. *Internet RFC 1546*, Nov. 1993.

[56] Charles E. Perkins and Elizabeth M. Royer. Ad hoc on-demand distance vector routing. In *Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Application*, pages 90–100, New Orleans, LA, Feb. 1999.

[57] Ram Ramanathan, Jason Redi, Cesar Santivanez, David Wiggins, and Stephen Polit. Ad hoc networking with directional antennas: A complete system solution. *IEEE Journal on Selected Areas of Communication*, 23(3):496–506, 2005.

[58] Volkan Rodoplu and Teresa H. Meng. Minimum energy mobile wireless networks. *IEEE Journal in Selected Areas in Communications*, Vol. 17(No. 8):1333–1344, Aug 1999.

[59] Martin Roth and Stephen Wicker. Termite: Ad-hoc networking with stigmergy. In *Inproceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, volume Vol. 5, pages 2937 – 2941, Dec 2003.

[60] Winston K.G. Seah, Kevin Z. Liu, Marcelo H. Jr. Ang, J. G. Lim, and S.V. Rao. Tarantulas: Mobility-enhanced wireless sensor-actuator networks. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing - Vol 1 (SUTC'06)*, pages 548–551, June 2006.

[61] Archana Sekhar, B. S. Manoj, and C. Siva Ram Murthy. Dynamic coverage maintenance algorithms for sensor networks with limited mobility. In *Proceedings of IEEE International Conference on Pervasive Computing and Communication*, pages 51–60, March 2005.

[62] S. Shakkottai, R. Srikant, and N. Shroff. Unreliable sensor grids: Coverage connectivity and diameter. In *Proceedings of IEEE Conference on Computer Communications, INFOCOM*, 2003.

[63] Bo Sheng, Qun Li, and Weizhen Mao. Data storage placement in sensor networks. In *Inproceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*, pages 344–355, New York, NY, USA, 2006. ACM Press.

[64] Arun A. Somasundara, Aman Kansal, David D. Jea, Deborah Estrin, and Mani B. Srivastava. Controllably mobile infrastructure for low energy embedded networks. *IEEE Transaction on Mobile Computing*, Vol. 5(No. 8):958 – 973, August 2006.

[65] D. Stoyan, W.S. Kendall, and J. Mecke. *Stochastic geometry and its applications.* J. Wiley, New York, 1995.

[66] Devika Subramanian, Peter Druschel, and Johnny Chen. Ants and reinforcement learning: A case study in routing in dynamic networks. In *Inproceedings of International Joint Conference on Airtificial Intelligence*, pages 832 – 839, 1997.

[67] Herbert Tanner, Ali Jadbabaie, and George J. Pappas. Flocking in fixed and switching networks. *IEEE Transaction on Automatic Control*, Submitted July 2003.

[68] Y. C. Tay and K. C. Chua. A capacity analysis for the IEEE 802.11 MAC protocol. *Wireless Networks*, 7(2):159–171, 2001.

[69] Niwat Thepvilojanapong, Yoshito Tobe, and Kaoru Sezaki. Har: Hierarchy-based anycast routing protocol forwireless sensor networks. In *Proceedings of The 2005 Symposium of Applications and Internet (SAINT '05)*, pages 204–212, 2005.

[70] D. Turgut, B. Turgut, R. Elmasri, and T. V. Le. Optimizing clustering algorithm in mobile ad hoc networks using simulated annealing. In *IEEE Wireless Communication and Networking*, pages 1492–1497, New Orleans, Louisiana, Mar. 2003.

[71] P. J. M. van Laarhoven and Emile H. L. Aarts. *Simulated Annealing: Theory and Application.* Kluwer Academic Publishers, 1987.

[72] G. Wang, G. Cao, and T. La Porta. Movement-assisted sensor deployment. In *Proceedings of IEEE Conference on Computer Communications, INFOCOM*, Hong Kong, Mar 2004.

[73] Guiling Wang, Guohong Cao, Tom La Porta, and Wensheng Zhang. Sensor relocation in mobile sensor network. In *Proceedings of IEEE Conference on Computer Communications, INFOCOM*, volume Vol 4, pages 2302–2312, Miami, FL, Mar 2005.

[74] Jianxin Wang, Yuan Zheng, Cheng Leung, and Weijia Jia. A-DSR: A DSR-based anycast protocol for ipv6 flow in mobile ad hoc networks. In *Vehicular Technology Conference*, volume 5, pages 3094–3098, Oct 2003.

[75] Yu Wang and J. J. Garcia-Luna-Aceves. Spatial reuse and collision avoidance in ad hoc networks with directional antennas. In *IEEE Global Telecommunication Conference (GLOBECOM)*, pages 112–116, 2002.

[76] Wayne L. Winston and Munirpallam Venkataramanan. *Introduction to Mathematical Programming.* Duxbury Press, 4 edition, 2002.

[77] Mohamed Younis, Kemal Akkaya, Mohamed Eltoweissy, and Ashraf Wadaa. On handling QoS traffic in wireless sensor networks. In *Proceedings of 37th Annual Hawaii Conference on System Sciences (HICSS)*, volume Vol. 9, 2004.

[78] Wensheng Zhang, Guohong Cao, and Tom La Porta. Data dissemination with ring-based index for wireless sensor networks. In *Inproceedings of IEEE International Conference on Network Protocols*, pages 305–314, Nov 2003.

[79] Y. Zou and Krishnendu Chakrabarty. Sensor deployment and target localization based on virtual forces. In *Proceedings of IEEE Conference on Computer Communications, INFOCOM*, San Francisco, Mar 2003.

Vita

Rajesh N. Rao

Rajesh N. Rao received his bachelors degree from BMS College of Engineering, Bangalore University. He got his M.S. and Ph.D. from Pennsylvania State University in Electrical Engineering. His area of research is in mobile ad-hoc networks and includes, energy efficient routing, mobility management of nodes and intrusion detection.