COMMUNICATION PROTOCOLS AND SENSING COVERAGE IN MOBILE AD HOC AND WIRELESS SENSOR NETWORKS

By

JIONG WANG

A dissertation submitted in partial fulfillment of
the requirements for the degree of

DOCTOR OF PHILOSOPHY

WASHINGTON STATE UNIVERSITY
School of Electrical Engineering and Computer Science

DECEMBER 2008

To the Faculty of Washington State University:

The members of the Committee appointed to examine the dissertation of JIONG WANG find it satisfactory and recommend that it be accepted.

_____

Chair

_____

_____

ACKNOWLEDGEMENT

To my advisor, Sirisha Medidi, I extend my deepest gratitude. Your continuous guidance and support was the bedrock upon which this dissertation was built. Through the (countless!) late nights and tight deadlines, your spirit of dedication and sheer persistence was a source of strength and inspiration.

To my committee members and mentors, Murali Medidi and K.C. Wang: thank you. Your deep knowledge and experience was invaluable. And where would a student be without a lab? My appreciation for the SWAN crew of Peter Cappetto, Yuanyuan, Monique Kohagura, Ghayathri Garudapuram, Chris Mallery, and Lynsey Compton-Drake runs deep. A doctorate is not built by a man alone I have been truly blessed by the incredible support of so many.

Finally, I extend a heartfelt and long-overdue credit to my mother and father. Without their understanding, support, gentle encouragements, and endless patience, this dissertation would still be but a dream yet fulfilled. I love you, and thank you.

COMMUNICATION PROTOCOLS AND SENSING COVERAGE

IN MOBILE AD HOC AND WIRELESS

SENSOR NETWORKS

Abstract

by Jiong Wang, Ph.D.
Washington State University
December 2008

Chair: Sirisha Medidi

In recent years, Mobile Ad-hoc Networks (MANETs) and Wireless Sensor Networks (WSNs) have received tremendous attention due to their desired features of self-configuration and self-maintenance. To address and mitigate the problems such as Broadcast Storm, latency, and congestion, this thesis presents a routing protocol for MANETs and a data transport protocol for WSNs. In addition, the coverage problem in WSNs is discussed with innovative solutions.

To address broadcast storm problems, stale route, faulty nodes and latency in MANETs, this thesis proposes a Fault resilient MANET Routing protocol, FaRM, in which a density-first route selection technique is used to select the most robust routes during route discovery. A local self-recovery mechanism is proposed for route maintenance. The performance evaluation based on *ns-2* shows significant improvements in FaRM's throughput, overhead, scalability and stability in demanding environments with high mobility and heavy traffic loads.

WSNs are generally used for harsh environments involving military actions. Due to severe resource constraints in sensor nodes, including memory space, energy storage, and communication bandwidth, in-network data aggregation is needed. We propose a sensor-to-sink transport protocol, which is suitable for data aggregation and provides reliable upstream packet delivery by dynamically configuring inactive nodes as *"monitors"* to assist in quick loss detection and recovery. *ns-2* based simulations confirm that the monitor-based transport protocol improves the throughput and data delivery rate with the intermittent traffic load and unpredictable node failures.

The goal of deploying a large-scale WSN is to utilize those spatially-distributed autonomous sensors for monitoring certain physical and environmental conditions in a target area. This thesis proposes a topology control technique to configure a densely deployed network and a distributed algorithm to utilize sensors with variable sensing radii for optimal sensing coverage. In addition, a group-based technique is discussed to provide a general approach that can extend any *1*-coverage algorithm into *k*-coverage. The performance comparisons confirmed that the proposed techniques reduce sensing energy consumption and maintain a sound coverage ratio for reliable surveillance.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER ONE

INTRODUCTION

Wireless Sensor Networks (WSNs) and Mobile Ad Hoc Networks (MANETs) are two self-maintained wireless radio networks that are used widely in many applications such as battlefield surveillance, environmental monitoring, emergency responses, and etc. These networks provide a convenient and inexpensive networking infrastructure but pose significant challenges due to resource-constraints, dynamic topology changes, error-prone communication links, to name a few. As wireless sensors and mobile devices become universally available with inexpensive prices due to recent advancement in embedded systems and communication techniques, it is critical to provide reliable communication protocols and guaranteed Quality-of-Service to WSNs and MANETs from a software perspective. The thesis discusses and addresses three key problems in wireless sensor and mobile ad hoc networks, *i. e.* fault-resilient routing that efficiently handles mobility and packet losses in mobile ad hoc networks, reliable data transport for upstream converge-cast data delivery in wireless sensor networks, and energy efficient coverage that provides flexible and guaranteed QoS for continuous surveillance and monitoring in sensor networks.

1.1 Routing Protocol in Mobile Ad Hoc Network (MANET)

1.1.1 Mobile Ad Hoc Networks

A Mobile Ad-hoc NETwork (MANET) [22] is a temporary, self-organized packet radio network that provides continuous connectivity without the assistance of preinstalled infrastructures. With the proliferation of portable computing devices, MANETs are becoming widely used in many places where dedicated routing infrastructures are not available, including in-home networking, wireless LAN, nomadic computing, and short-term networking for disaster relief, public event, and temporary offices [63].

Current research in MANET design is focused on distributed routing [58]. Every mobile host in a MANET must operate as a router in order to maintain connectivity information and forward packets from other mobiles. Traditional routing protocols for wired networks, such as Distance Vector Routing and Link State Routing, are based on Dijastra's shortest pathes algorithm. Those routing protocols perform well under wired networks where static topologies and low-loss communication links are guaranteed. However, such

features including networking flexibility and node mobility in MANET can raise critical challenges with traditional routing protocols. First, node mobility can cause significant overhead and delay in order to update topology information for route maintenance. Secondly, due to media access contention and packet collision, flooding of control packets such as Route Requests, can result in substantial performance degradation, *i.e.* the Broadcast Storm problem [61]. Lastly, without a dedicated routing infrastructure, packet drops at the selfish/misbehaving nodes are difficult to avoid. In order to address and mitigate those problems associated with packet routing in MANET, desirable features of a MANET routing protocol should include:

- Low routing overhead and latency
- Routing stability
- Fault-resilience to various route failures

1.1.2 Table-Based Routing and Source-Initiated Routing

Current MANET routing protocols are classified as either table-based [16, 43] or source-initiated [15, 19, 47, 60, 62]. According OSI reference model, routing protocol is responsible for selecting a desired path for a given communication pair and forwarding data packets on such selected pathes. In wired networks, routing protocols are implemented on the dedicated routers in core networks. Such dedicated routers are responsible for maintaining a complete routing table for the network. Table-based routing for MANET is a similar approach to those used in wired networks. In table-based routing protocols, each node maintains a routing table for every destination using periodic topology updates. However, under frequent topology changes, routes become unavailable quickly. Therefore, an additional field, *i.e.* the sequence NO. [16], is used to keep the freshness of each entry in the routing table. Overhead is the known problem of table-based routing because of its proactive approach to keep a complete route table for every destination in the network. Also, under environments with high mobility and packet losses, experimental results demonstrated a large number of routing tables with inconsistent information which would degrade the performance of table-based routing protocols considerately.

In contrast, source-initiated routing is a completely different approach to those traditional routing protocols used in wired networks. Most of these protocols do not use routing tables or only use partial routing

tables for active flows. In source-initiated routing, all routes are obtained on-demand through a source-initiated route discovery phase. During the route discovery phase, the source broadcasts a Route Request packet to the network. Then, a Route Reply packet will be returned to the source once the Route Request packet reaches the destination. The obvious advantage of source-initiated routing is that it alleviates the overhead regarding to the table maintenance in highly-dynamic environments. However, since the route discovery phase is initiated on the fly and relies on flooding, problems such as latencies, stale routes, and routing overhead still exist.

1.1.3 A Fault-Resilient MANET Routing Protocol (FaRM)

This thesis includes a Fault-Resilient MANET (FaRM) routing protocol based on a two-phase, source-initiated routing. During route discovery, FaRM uses a density-first route selection technique to avoid transient routes as well as maintaining a sound throughput of the network. This route selection technique is based on an analytic model of link lifetime. This link-lifetime model allows the source to select an optimal route with longer residual lifetime and better throughput. For routing maintenance, FaRM builds in a local self-recovery mechanism in order to improve the routing stability and fault-resilience in a demanding environment with high mobility and unpredictable node failures. This route recovery mechanism uses a cooperative searching process based on a cone-shaped recovery zone and hence, can avoid faulty links and repair the broken route in the shortest possible time, as well as to achieve continuous packet-forwarding. The contribution of FaRM can be summarized as:

- FaRM selects a better route based upon a node's mobility and the size of its neighborhood, which does not involve in any extensive computation or communication operations.
- FaRM uses a cooperative searching process to locally mitigate a route error and reduce the delay and overhead from initiating the route discovery phase.

1.2 Transport Protocols in Wireless Sensor Networks (WSN)

1.2.1 Wireless Sensor Networks

Recent advances in integrated circuit technology have enabled construction of inexpensive sensor equipped with low-power signal processing and computation capabilities. There have been several successful academic initiatives in developing low-cost wireless sensors. For example, the "Smart Dust" mote from U. C.

Berkeley can transmit passively using novel optical reflector technology, hence provides an inexpensive way to probe a sensor or acknowledge that information was received [37]. The wireless smart sensor platform developed at UCLA supports hardware interface, payload and communication needs of multiple inertial and position sensors and actuators, using RF link for communications with low-cost [4]. Industrial sensor products, such as Crossbow's MICA family of wireless sensors, have also become available to the market. However, energy storage is still one of the major constraints due to limitations in current battery technology. According to "Smart Dust" nodes, the total stored energy is on the order of 1 Joule using the best available battery technology. Therefore, careful power management strategies have to be utilized in order for sensors to function over a practical period of time such as in the span of weeks or months.

A Wireless Sensor Network (WSN) is a collection of a large number of minuscule sensors deployed in potentially hostile environments for military surveillance, emergency response, and natural disaster monitoring. A WSN generally consists of a data acquisition network and a data distribution network, monitored and controlled by a management center (also referred to as the base station or sink) [14]. The objective of a WSN is to detect the relevant quantities, monitoring and collecting the data, accessing and evaluating the information, formulating meaningful user displays, and performing decision-making and alarm function. Due to the various limitations in sensor nodes, networking hardware, and battery technology, the challenging of designing a robust communication protocol for a ad-hoc, non-attendant, and dynamic network is enormous.

1.2.2 Transport Protocols for Upstream and Downstream Data Delivery

Although significant efforts have been devoted to develop sensor hardware and communication protocols for link and networking layer, the transport protocol for WSN is still a relatively uninvestigated area. Based on the traffic pattern, Transport protocols for WSN can be categorized as downstream and upstream. Unlike traditional wired networks or MANET, there are substantial differences between downstream and upstream data delivery in WSNs. The downstream data delivery (sink-to-sensor) are similar to IP multi-casting in traditional TCP/IP networks. Control packets are generated by the sink and broadcast to all sensors in the network. In WSNs, downstream data transport takes place periodically when the sink needs to re-task or query sensor nodes; thus, a packet-level reliability is required in order to have the network function as expected. On the other hand, upstream data delivery (sensor-to-sink) is an event-driven, many-to-one converge-cast [17]. Once a group of sensors are trigger by certain events, those sensors will generate sensory

data and send them to the sink. Depending on types of events monitored by the network, multiple sources from different areas could send data to the sink simultaneously. The reliability of upstream data delivery is not as strict as downstream data delivery due to abundant redundancy of sensory data. Hence, event-level reliability is generally considered for upstream transport protocols.

Sensor nodes are fundamentally constrained in memory space, processing capability, communication bandwidth, and energy storage. The many-to-one converge-cast for upstream data delivery limits network scalability because sensors that are closer to the sink are more prone to congestion and essentially become a network traffic bottleneck. The dilemma between resource-constraints and intermittent heavy traffic has drawn considerable attention to in-network data-processing techniques such as data aggregation [12, 26, 76], for better scalability and energy-efficiency. For example, in Direct Diffusion [12], packets are forwarded based on their contents, and packets containing the correlated information will be aggregated at intermediate nodes. Since correlated data are reduced, a significant amount of information will be lost if one packet cannot be reliably delivered to the sink. However, most current transport protocols only provide event-level reliability for sensor-to-sink data delivery and, as such, cannot sufficiently provide a reliable service for applications in military surveillance or environmental monitoring. In order to be compatible with data aggregation, it is necessary to develop a new sensor-to-sink transport protocol which provides packet-level reliability with energy-efficiency.

1.2.3 A Monitor-Based Transport Protocol for Upstream Data Delivery

In order to provide packet-level reliability in environments with high loss-rate, serious bottlenecks, and limited energy storage, we developed a topology control technique to dynamically initiate inactive sensors as *"monitors."* In WSNs, sensor nodes are cooperative and redundantly deployed, hence, utilizing information provided by monitors (inactive non-forwarders) will assist in a more reliable loss detection and recovery in cases of congestion and sudden node-failures. The monitor-based transport protocol dynamically construct an auxiliary network with monitors based on current flows in the network. The auxiliary network will work cooperatively with active nodes to obtain more reliable upstream data delivery. However, monitors consume additional energy for several responsibilities, such as continuously listening to the shared media, caching packets for the monitored nodes, and resenting lost packets. In order to ensure energy-efficiency and packet-level reliability, it is necessary to identify a minimum set of monitors that covers all current flows

(all active nodes need to be monitored). We formulate this process of energy-efficient monitor-configuration as a Minimum Vertex Set Cover (MVSC) problem and present a distributed heuristic to solve this problem efficiently.

1.3 Sensing Coverage in Wireless Sensor Networks

1.3.1 Quality-of-Service and Energy-Efficient Coverage in WSN

A Wireless Sensor Network (WSN) is comprised of a large number of coin-size devices with radio communications, sensing, and low-power processing capabilities. One of the fundamental objectives of WSNs is to provide continuous surveillance such that each point therein is monitored by at least one sensor. In order to provide better accuracy and fault-tolerance, some applications, such as emergency response, military surveillance, and disaster-recovery, require each point within the target area to be monitored independently by multiple sensors. Therefore, coverage is one of the fundamental QoS (Quality-of-Service) metrics in order for a WSN to provide required reliability in service.

There are two problems regarding to providing QoS-satisfied coverage to a target area. One problem is to ensure no uncovered or under-covered area (hole) exists with all sensors turned on. This problem is related to the sensor-deployment problem. Under the assumption that the area is well-covered with redundant sensors, the other problem is energy-efficient coverage, which is to configure an optimal coverage that satisfy the QoS requirements of an application and obtains maximum energy-efficiency by reducing redundant coverage. This problem is related to the topology control problem. There are three possible ways to achieve energy efficiency by optimizing sensing coverage: one is to turn-off those sensors that generate redundant coverage; the second way is to reduce the overlapping coverage by utilizing sensors with variable sensors; and the last way utilizes mobile sensors to obtain an optimal topology for coverage. This thesis focuses on first two approaches because of the availability of sensors with sleep/wakup scheduling and variable sensing range.

1.3.2 1-*Coverage with Uniform Sensing Radius*

Once deployed to a target area, most applications require continuous coverage where each point therein is monitored by at least one sensor. This is commonly known as *1*-coverage. Due to the energy-constraint of sensor nodes, it is of vital importance to dynamically schedule a minimum set of sensors that can provide guaranteed coverage of a target area [67].

The problem of energy efficient *1*-coverage with unform sensing radius assumes all sensors have an identical sensing range which is a perfect circle with radius R. The challenges of solving this problem lies in several aspects: First of all, sensor networks are deployed in large-scale, which discourages the use of any centralized optimization techniques. Secondly, the resource-constraints of sensor nodes such as size, power, and bandwidth, require a lightweight algorithm with localized communication overhead. Lastly, when WSNs are employed for use in border surveillance, monitoring for biological, chemical and nuclear weapons, or military reconnaissance, the need for full coverage is of critical importance. Thus, a coverage scheduling algorithm should always maintain a full coverage with minimum required sensors.

To address these issues, this thesis proposes a technique that builds a network topology based on a square mesh or equilateral-triangular mesh in order to obtain minimum overlapping coverage among sensors. To build this mesh-based topology, we propose a self-adaptive mesh construction mechanism. Key features of this technique include:

- Quick mesh construction with local overhead
- Randomized reconstruction that allows energy-balancing
- Guaranteed full coverage

1.3.3 1- *Coverage with Variable Sensing Radii*

Due to sensors' energy constraints, redundant coverage can be reduced either by using sleep/wakeup scheduling, or by varying sensing radii. The idea of configuring optimal sensing radii for sensing coverage is similar to energy-efficient routing in MANET. In wireless communication, the energy consumption of sending a packet is proportional to r^n where r is the transmission range and n is the path loss exponent. Therefore, the commonly-used shortest path routing may include longer hops and become less energy efficient. For energy consumption in sensing, large sensing radii increase energy consumption because they require more sophisticated filtering and signal-processing methods to improve the signal-to-noise ratio and achieve the desired confidence level [75].

Recently, sensors with adjustable sensing range have been developed and become available to the market. To utilize sensors with variable sensing radii, this thesis proposes a dynamic radii configuration technique that guarantees reliable surveillance and provides energy-efficiency through eliminating redundant

coverage. Due to resource constraints and wide-spread deployment of sensor networks, a light-weighted and distributed algorithm is preferred. Therefore, objectives of our design should include:

- A distributed algorithm based on one-hop information
- Guaranteed full coverage
- Energy-efficiency in sensing

1.3.4 A Generalized Technique for k-*Coverage*

Although *1*-coverage is the most common case for WSN, some applications may require a higher level of coverage. For example, localization techniques (*e.g.* [41]) based on triangulation need at least three sensors in order to decide the location of a given point. Furthermore, many applications may prefer coverage with configurable redundancy, where a higher level of coverage can be obtain in critical moments for better fault-tolerance and accuracy. Unfortunately, current research for *k*-coverage is either limited to solving the decidability problem of *k*-coverage or lack of the reconfiguration capability.

Instead of proposing a brand-new technique for *k*-coverage, this thesis discusses a generalized approach that can extend any existing *1*-coverage algorithms into *k*-coverage without sacrificing their original properties in energy efficiency and QoS. The technique we are proposing is based on distributed grouping, where *k* mutually-exclusive groups are constructed with an identical distribution and density. With a fair division, each group independently decides the scheduling scheme or sensing radii configuration for its own sensors. Therefore, the properties of original *1*-coverage algorithm, such as energy-efficiency and full coverage, are well-preserved. Furthermore, the technique of group construction allows the division of groups to be reconfigurable. Thus, variant QoS requirements of the application can be easily satisfied.

CHAPTER TWO

A FAULT-RESILIENT MANET ROUTING

2.1 Related Work

Table-driven routing protocols are proactive and incur significant overhead whereas, source-initiated routing protocols use reactive on-demand route discovery, and thereby are more suitable for MANET. In DSR [19], route discovery is initiated on-demand by broadcasting a Route Request packet. The Route Request packet contains the route that it has passed so far. Once the Route Request reaches the destination, a Route Reply packet containing the complete route will be returned to the source. To reduce the broadcasting overhead, routes are temporally cached in order to allow any intermediate node to send a reply. AODV [15] is a combined approach with the source-initiated routing and the table-based routing. During route discovery, AODV establishes an end-to-end flow between the source and the destination after route discovery is completed. The route maintenance phase of AODV will detect the broken route and propagate a link failure notification to the source for reinitiating route discovery. In TORA [62], based on the link-reversal technique, multiple routes are obtained for any desired source/destination pair after route discovery. During TORA's route maintenance, a localized control packet is propagated near the topology changes in order to update link states. In LAR [72], the source node uses information on nodes' location and movement to reduce the broadcasting overhead during route discovery. LAR estimates an *expected zone* where the destination is most likely located. Based on the *expected zone*, LAR broadcasts the Route Request within a *request zone* that contains possible routes towards the destination.

For the lack of any dedicated routing infrastructure in MANET, route stability has become a desirable feature. In ABR [60], stable routes are given a higher preference than the shorter ones. Each node generates periodic beacons (associativity ticks) to maintain an associativity table. This information is used later on during route discovery to select a route with maximum degree of stability. In SSA [47], routes are chosen by classifying links as weak/strong connected sets based on signal strength and local stability. By choosing the most stable routes instead of the shortest path, SSA has shown significant reduction in the number of route reconstructions. However, the original paper of SSA does not provide an evaluation of throughput with their new route selection scheme. Authors in [38] propose a metric called "affinity" based on a prediction

of a link's lifetime. The "affinity", a_{nm}, is defined as the time taken by a node n to move out of the range of a node m. A localization technique is required in order to estimate the "affinity" between two nodes. In RABR [51], authors extend this affinity-based metric by considering throughput for TCP connections, where both the signal strength and route length are considered in order to select a more stable route without significantly degrading the throughput of the network. In FaRM, we estimate a route's residual lifetime based on a statistical model of node distribution and mobility, which gives a more accurate and convenient estimation in selecting the most robust route.

Due to node mobility and unpredictable node failures, route maintenance becomes a bottleneck for MANET routing protocols. ABR [60] developed a partial route discovery technique through backtracking the pivoting nodes which use the locally cached routing information to restore the broken route. In WAR [35], after the witness node detects a broken link, it rebroadcasts the data packet until the packet reaches its original route. In NSR [45], each relay node uses periodic topology updates to maintain its two-hop neighborhood information for route repairing. In CHAMP [5], packets are cached at relay nodes for salvaging when a route error packet is received from downstream nodes. However, these route recovery techniques have some limitations. For example, ABR can cause significant latency if its pivoting nodes fail; WAR requires witness nodes to be dedicated to each link and causes new overhead for broadcasting undeliverable packets; NSR may overload the links with periodic link-state updates; and CHAMP requires additional storage for packet recovery.

Fault-resilient routing protocols are demonstrated by using redundant routers in wired networks [74]. For MANET, multipath routing [2, 50] can provide limited fault-resilience under low node mobility. As mobility increases, routes become stale quickly, therefore, multipath routing may introduce additional delay and overhead through the usage of backup routes. For those faulty routes caused by malicious nodes, SEAD [29] and Ariadne [30] utilize a one-way hash chain to provide authentication schemes in order to secure DSDV and DSR. Also, authors in [6] have proposed a binary search method to find the malicious node within $LogN$ ACKs, where N is the length of the route. In contrast, FaRM focuses on a quick loss recovery and avoidance technique after packet-losses occur. Therefore, we design a novel route self-recovery technique that can locally handle various route-failures caused by mobility, power outage, and malicious attacks in the shortest possible time.

2.2 Protocol Design

Fault-Resilient MANET (FaRM) routing protocol is based on source-initiated routing. It consists of two phases: route discovery and route maintenance. During route discovery, a robust route with a longer lifetime and better throughput is chosen. The route selection technique is based upon the size of a node's neighborhood, mobility, and route length. For route maintenance, a route self-recovery technique is initiated to achieve more efficient route repair and better fault-resilience when there is a frequent occurrence of route failures.

2.2.1 Route Discovery

Route discovery is initiated by the source node when a route is needed or a route error message is received. Since route discovery is broadcast-based, selecting a route with the maximum residual lifetime is preferred in order to avoid unnecessary route discovery. In FaRM, each node maintains the information about its local density and its neighbors' speeds. A Route Request packet collects these information as well as estimating the link's residual lifetime at each intermediate node in the route. Once multiply Route Request arrives at the destination, a Route Reply containing the best route will be returned to the source based on the estimation of each route's lifetime and throughput.

Estimation of Link Lifetime

In order to investigate a route's lifetime in MANET, we develop a model for analyzing the upper-bound of a link's expected lifetime. We assume the distribution of nodes has a Homogeneous Poisson Point Process (HPPP) with density λ (*Definition 1*). A HPPP is commonly used in modeling ad-hoc networks for their initial placement and investigating point processes that are neither completely random nor regular [64, 49] .
Definition 1: Homogenous Poisson Point Process (HPPP): On a two-dimensional space Ω with points, $N(A)$ is a counting measure (number of points) of a bounded Borel set A ($A \in \Omega$) and $\sigma(A)$ denotes the Lebesgue measure of set A. The spatial distribution of points across Ω is an HPPP if and only if: (1) $N(A)$ is the Poisson Distribution; and (2) $N(A_i)$ and $N(A_j)$ are independent for any disjoint set A_i and A_j.

Theorem 1 *An ad-hoc network has a homogeneous Poisson point process of density λ, nodes move with a random speed V and have a transmission range D. For the worst-case scenario where the node is moving away from its neighbor, the average link lifetime $E(T)$ satisfies:*

$$E(T) \leq (D - \frac{1}{2\sqrt{\lambda}})/E(V)$$

Proof: A homogeneous Poisson point process has

$$Pr(N(A) = k) = \frac{e^{-\lambda\sigma(A)}(\lambda \cdot \sigma(A))^k}{k!} \tag{2.1}$$

If X is a random variable of the distance between a node and its nearest neighbor, the cumulative distribution function of X is:

$$\begin{aligned} F(x)&=Pr\{X \leq x\} = 1 - Pr\{X > x\} \\ &=1 - Pr\{N(\pi x^2) = 0\} = 1 - e^{-\lambda\pi x^2} \end{aligned} \tag{2.2}$$

From Equation 2, we can denote the probability density function and the expected value of X as:

$$f(x) = dF(x)/dx = 2\lambda\pi x e^{-\lambda\pi x^2} \tag{2.3}$$

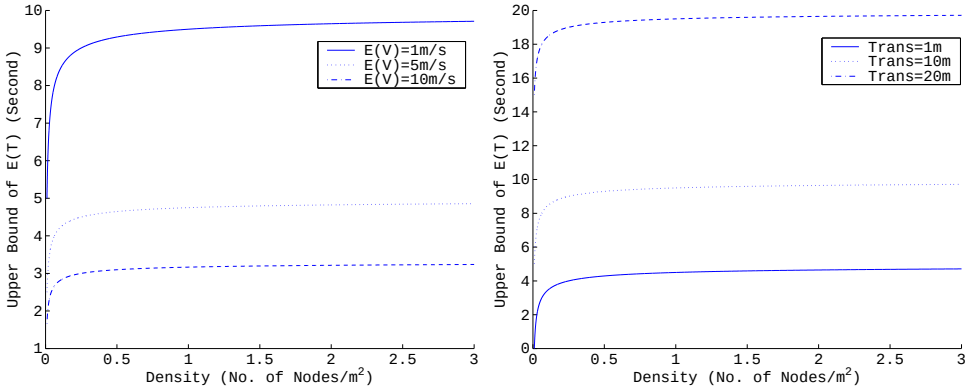
$$E(X) = \int_0^\infty x f(x)\mathrm{d}x = \frac{1}{2\sqrt{\lambda}} \tag{2.4}$$

Assume random variable $\mathcal{X}$ is the length of the link (the physical distance between two nodes), then the upper bound of $E(T)$ is:

$$E(T) \quad = \quad E(\frac{D - \mathcal{X}}{V}) = \frac{D - E(\mathcal{X})}{E(V)} \leq \frac{D - E(X)}{E(V)} = \frac{(D - \frac{1}{2\sqrt{\lambda}})}{E(V)} \tag{2.5}$$

■

The result, as illustrated by *Theorem* 1, is that the link lifetime is affected both by node mobility and network density. In dense networks, the distance between two communicating nodes tends to be shorter and, consequently, the links become more difficult to break when nodes are moving. Based on *Theorem* 1, Figure 2.1 shows the analytical results of the upper-bound of $E(T)$ with various node densities (λ).



(a) (b)

Figure 2.1: Average Link Lifetime: (a) Transmission Range = 10m (b) Average Speed = 1m/s

Density-First Route Selection

During route discovery, multiple routes can be identified after the Route Request (RReq) packet is propagated throughout the network. Due to mobility, the shortest route can break up quickly and, as a result, may incur a significant amount of overhead and delay from re-initiating route discovery.

In order to avoid transient links and obtain good throughput, FaRM uses density, mobility, and route length to choose an optimal route based on *Theorem* 1. Each RReq packet's header contains the complete route that it has passed, which is denoted as $\Omega = v_1 v_2 ... v_n$. Then, an intermediate node receiving the RReq ranks the route Ω as defined by Equation 2.6. In Equation 2.6, $\overline{V_i}$ is the average speed of node v_i; N_i is the number of neighbors of node v_i ($i \in 1..n$), thus $N_i/\pi D^2$ is an approximation of local node density; $|\Omega|$ is the length of route Ω; and α is a constant for the tradeoff between route lifetime and throughput.

The average moving speed $\overline{V_i}$ is calculated at each node according to Equation 2.7 using Weighted Moving Average [53], where $\overline{V_i^t}$ and V_i^t represent the average and current speed of the node v_i at time t, and β is the weighted average ($0 < \beta < 1$). Furthermore, Equation 2.6 uses the minimum value of all links' estimated lifetime because a route's lifetime is decided by the link that breaks up first. When multiple RReqs reach the destination, the route with the highest ranking will be returned to the source via a Route Reply (RRel) packet.

$$R(\Omega) = Min((D - \frac{1}{2\sqrt{N_i/\pi D^2}}) \cdot \overline{V_i}) + \alpha \frac{1}{|\Omega|} \tag{2.6}$$

Table 2.1: FaRM's Route Table

Src	Dest	NextHop	RepairTimer	FlowTimer

$$\overline{V_i^t} = (1-\beta) \cdot V_i^t + \beta \cdot \overline{V_i^{t-1}} \tag{2.7}$$

FaRM does not allow an intermediate node to return the RRel packet from its cached route information in order to avoid staled routing information and obtain the best route available in the network during each route discovery phase. Furthermore, FaRM uses a routing table to maintain an end-to-end connection between the source and the destination. A routing table is more efficient in forwarding packets; and by distributing the route information at each intermediate node, it allows for a transparent recovery during the route maintenance phase. As in Table 1, Src, $Dest$ and $NextHop$ are basic routing information used for packet-forwarding; the $RepairTimer$ is setup to control the maximum delay during route recovery; and the $FlowTimer$ indicates whether the link is currently used by an active flow. If links are symmetrical, the end-to-end flow can be established using the routing table when the RRel reaches the source; otherwise, it will be established after the first data packet is delivered to the destination.

2.2.2 Route Maintenance

The route maintenance phase consists of three modules: Route Breakup Detection (RBD), Local Self-Recovery (LSR) and Stale Route Deletion (SRD). The transition between these three modules is shown in Figure 2.2. When RBD determines that one of the entries in the routing table is unreachable, it checks for whether that entry is used by an active flow and, if it is (in which case the $FlowTimer$ is still valid), RBD initiates LSR to find a detour for route recovery. If no flow is using that entry or no valid detour can be found before the $RepairTimer$ expires, SRD is initiated to delete the obsolete route and, if necessary, re-initiate route discovery.

Local Self-Recovery

In order to obtain better fault-resilience, LSR is initiated to bypass faulty links and quickly determine a detour (alternative route) that can reconnect the broken part without notifying other nodes of the route changes.

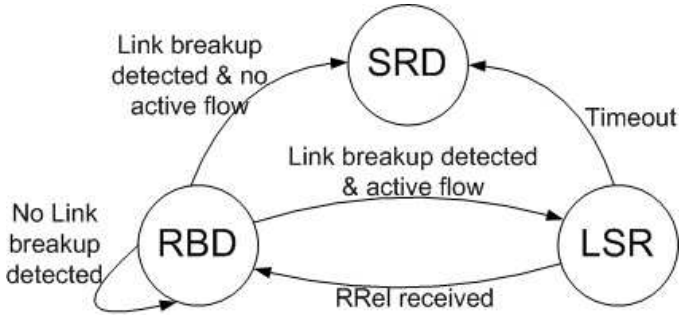


Figure 2.2: State Transform in Route Maintenance

One intuitive way of doing route recovery is to broadcast a route request by intermediate node; however, doing so incurs significant overhead and cannot effectively avoid those faulty nodes. In our approach, the node that starts LSR broadcasts a Route Recovery Request (RRReq) packet, wherein its recovery zone is defined. The recovery zone is a cone-shaped region with its apex located at the sender and its bisector passing through the faulty node. The node receiving the RRReq checks whether it is located within the recovery zone and then determines whether the RRReq needs to rebroadcast or dropped accordingly. By executing that sequence at each RRReq receiver, the faulty node is avoided and the broadcast area is confined to the union of multiple cone-shaped recovery zones. Furthermore, the coverage of those zones is centered at the faulty node and includes an area determined by the local topology and the angle of the recovery zone.

Figure 2.3 illustrates the coverage of collective broadcasts using the cone-shaped recovery zone after node A detects the faulty/misbehaving node F. In Figure 2.3-a, with a low node-density and small recovery zone angle, the coverage of RRReq is insufficient and will thereby affect LSR's successful recovery rate. In Figure 2.3-b, with the same request angle but higher node density, the coverage of RRReqs is improved. In Figure 2.3-c, with a higher node-density and larger angle, complete coverage can be achieved in the vicinity of node F. Therefore, the tradeoff between the successful recovery rate and overhead can be adjusted by varying the angle of the recovery zone. A study of determining an optimal angle is presented in section 2.3.5.

Figure 2.4 shows the process of a local route self-recovery. Node A detects the broken link and sends out a Route Recovery Request (RRReq) packet containing its recovery zone information. In this instance, the recovery zone is a cone-shaped region with its apex at node A and its bisector passing through the faulty node E. Any node located within the cone-shaped recovery zone will rebroadcast the packet if no

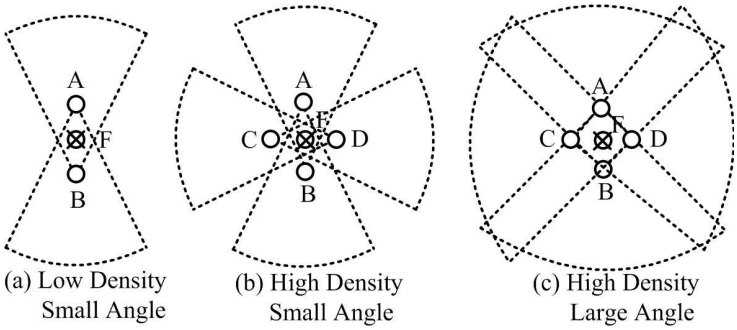


Figure 2.3: Broadcast Coverage Under Different Node Density and Angle

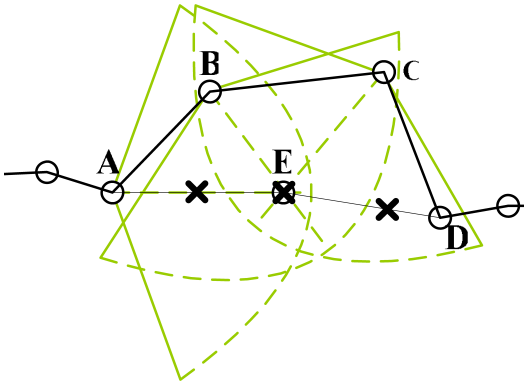


Figure 2.4: Cone-shaped Route Recovery Zone

available route can be found in its local cache. The rebroadcasting node follows the same rule to determine its recovery zone. For example, after node B receives the RRReq, it determines its cone-shaped recovery zone. The apex of its zone is at B and its bisector passes through E. If the route is only partially broken, an intermediate node on the original route will reply with a Route Recovery Reply (RRReply) packet and the repaired route can then be established.

The route maintenance can be summarized as follows:

1. Once RBD has detected a faulty/misbehaving node and the flow is still active, it starts the LSR by sending a Route Recovery Request (RRReq) packet containning information about its recovery zone. Then, the $Repair\text{-}Timer$ is setup.

2. When a RRReq packet has been received, the receiving node looks for a route to use for the request. If one is available, it returns a Route Recovery Reply (RRReply) packet. If the node is unable to locate a suitable route to send the reply, it determines whether its own location is within RRReq sender's

recovery zone. If it is, the node rebroadcasts the RRReq packet with its recovery zone's information.

3. When a RRReply packet has been received, the receiving node updates the corresponding route-entry in its routing table according to the RRReply it received.

4. If the RRReply packet is received by the initiator before its $RepairTimer$ expires, the buffered packets for that flow are sent instantly in accordance with the new routing information; otherwise, SRD is initiated and a route error packet will be returned to the source for a new route discovery.

If LSR fails, the overhead and the impact on other traffic is kept to a minimum since the broadcast area is confined to the vicinity of broken links. Due to the path-loss and the fading/interference of wireless links, there is no guaranteed delivery for the routing-related control packets, such as RRReq and RRReply. In the worst-case scenario, if a RRReply gets lost, a $RepairTimer$ time-out will be triggered. During such an occurrence, a route error packet will be sent to the source and a new route discovery will be initiated. The angle of the recovery zone is set up with the same default value for all nodes at the time of network bootstrap. Those angles can be individually adjusted by each node, based upon the traffic load and application requirements.

Calculation of Recovery Zone

In order for each RRReq receiver to determine whether it is located within the RRReq sender's recovery zone, the location information of both the RRReq sender and the failed node is required. Since the localization can be expensive to obtain without specialized hardware such as GPS [46], we only use the relative distances of one-hop neighbors, which can be more easily calculated using methods such as Time of Arrival [69]. After each node obtains its one-hop distance to neighboring nodes, it exchanges that information with its neighbors. Then, each node will have the one-hop distance information among any pair within its two-hop vicinity. Since the route recovery is done more efficiently within the locality of failed nodes, we only exchange distance information between neighbors. This exchange allows any node within the two-hop distance of the failed node to calculate the RRReq sender's recovery zone.

Figure 2.5 shows the calculation of the recovery zone using only one-hop distances. The first example is when the RRReq receiver is a one-hop neighbor of the failed node. Once node A fails, node B detects the failure and initiates route recovery by broadcasting a RRReq packet. Node C, the one-hop neighbor of

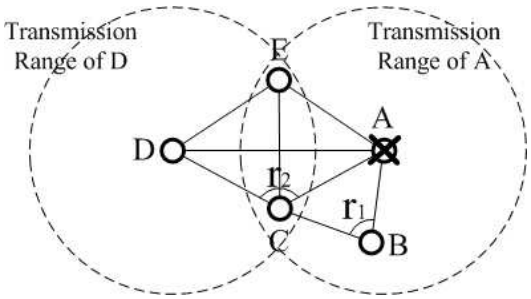


Figure 2.5: Calculating Recovery Zone with Distance Information

A, receives the RRReq from node B and, since the distances of AB, BC, and CA are known, node C can calculate γ_1 using Equation 2.8. The information of γ_1 allows for the conclusion that C is located within B's recovery zone if $\gamma_1 < \alpha/2$, where α is the angle of B's recovery zone. The second example shown in Figure 2.5 is when the node receiving RRReq is a two-hop neighbor of the failed node. Continuing with the previous example, when node D receives a RRReq, it does not directly know the distance of AD, because D is not in A's communication range. However, Node D can derive the distance of AD from CD, DE, CA, and AE using Equation 2.9, where $\alpha_1 = arccos(\frac{|CD|^2+|CE|^2-|DE|^2}{2|CD|\cdot|CE|})$ and $\alpha_2 = arccos(\frac{|CA|^2+|CE|^2-|AE|^2}{2|CA|\cdot|CE|})$. Then, Equation 2.10 can be used by D to calculate the recovery zone of C ($\gamma_2 < \alpha/2$).

$$\gamma_1 = arccos(\frac{|BC|^2 + |BA|^2 - |AC|^2}{2|BC| \cdot |BA|}) \tag{2.8}$$

$$|AD| = \sqrt{|CD|^2 + |CA|^2 - 2|CD| \cdot |CA| \cdot cos(\alpha_1 + \alpha_2)} \tag{2.9}$$

$$\gamma_2 = arccos(\frac{|CD|^2 + |CA|^2 - |AD|^2}{2|CD| \cdot |CA|}) \tag{2.10}$$

2.3 Performance Evaluation

Our experiments were based on the ns-2 simulator. First, we evaluated our density-first route selection technique by comparing the average route length and route lifetime of FaRM with those of DSR. Then, we investigated various local route recovery techniques by comparing the usefulness (percentage of successful recovery rate) of FaRM, NSR and WAR. Next, we evaluated FaRM, NSR, WAR, and DSR's performance

by examining their throughput and overhead. Furthermore, we tested the protocol's stability and scalability by varying simulation time and traffic load. A study of optimal recovery angle is presented at the end.

All simulations were conducted on an identical network setup which consisted of 50 randomly-deployed nodes in a 1,000m by 1,000m grid. Each node had a maximum transmission range of 250m and moved according to the Random Waypoint model. The traffic pattern used CBR connections each at a rate of 2.5 KB/s. The number of connection varied according to different experiments. Finally, the result of each experiment was obtained by calculating the average of 20 trials with randomly-generated topology and mobility scenarios.

2.3.1 Route Quality: Lifetime and Length

In order to evaluate density-first routing decision, we examined the average route length and the average lifetime of FaRM; then, using the same traffic pattern, we obtained the same averages of DSR and compared the two sets of data. The traffic pattern used consisted of 20 CBR flows at a rate of 2.5KB/s. Figs. 2.6-a and b illustrate the average route lifetime as mobility (maximum speed) increases. In Figure 2.6-a (with a 5-second pause-time), a decrease of route lifetime can be observed for both FaRM and DSR; however, because FaRM selects the routes with nodes that have higher density and lower mobility, its average route lifetime was improved by about 10%. A similar trend can also be observed in Figure 2.6-b (with a 30-second pause-time). In FaRM's density-first routing decision, route length is also considered. Figure 2.7-a and b shows that FaRM has only marginal increase in its average route length.

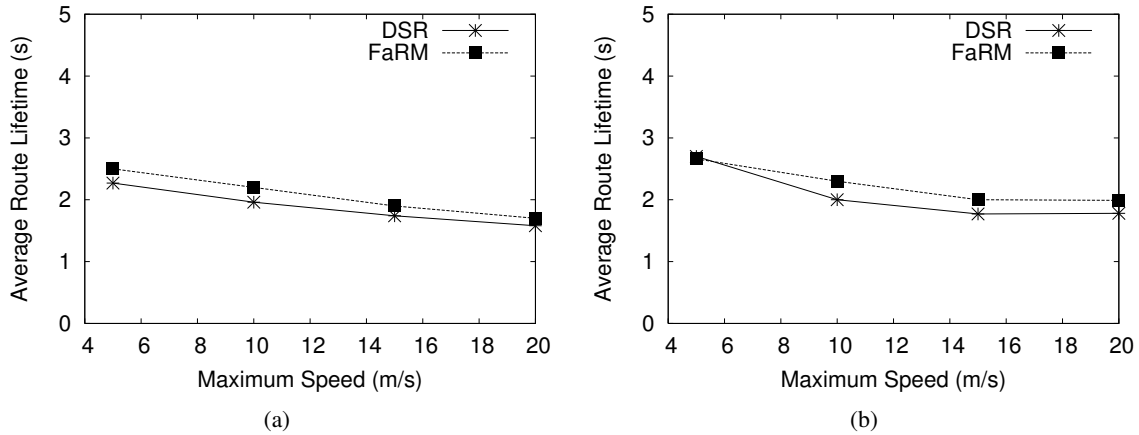


Figure 2.6: Average Route Lifetime: (a) Pause Time=5s (b) Pause Time=30s

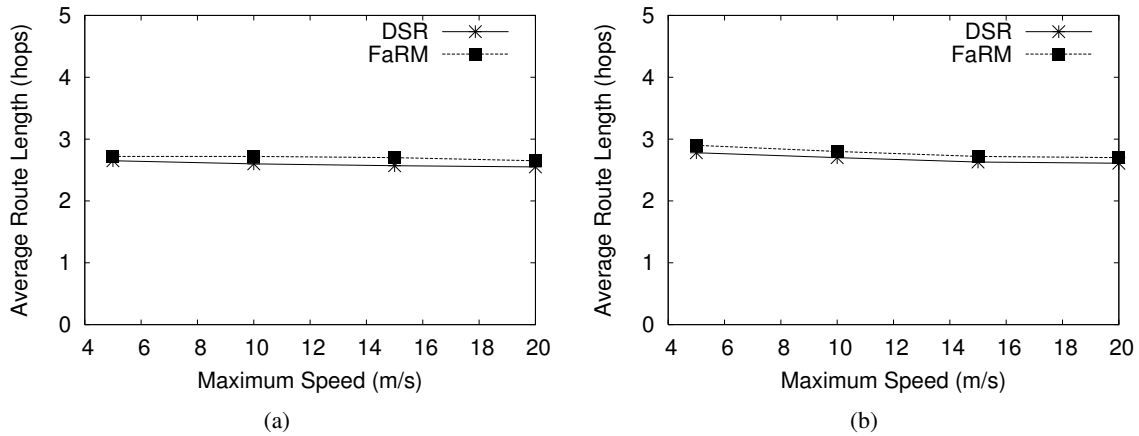


Figure 2.7: Average Route Length: (a) Pause Time=5s (b) Pause Time=30s

2.3.2 Local Recovery Techniques: Usefulness

In order to evaluate the effectiveness of various local route recovery techniques, we compared the usefulness of WAR, NSR and FaRM. The usefulness is measured as the percentage of successful local recoveries in the total number of local recovery requests. In WAR, each link has been assigned with a witness node and this witness node will broadcast any undeliverable packet if a packet loss has been detected. In NSR, each node maintain its two-hop neighborhood information by periodic topology updates. A detour is found by the intermediate node using its local topology information. In FaRM, we use a cone-shaped recovery zone to search the available detour near the faulty nodes. The traffic pattern used consisted of 20 CBR flows at a rate of 2.5KB/s. Figure 2.8-a and b show the percentage of successful recoveries under scenarios with different maximum moving speeds. Our results indicate that WAR has the most successful recovery rate between 25% to 35% because WAR requires additional resources allocated for each link as witness nodes. Without the assistance of witness nodes, FaRM achieves second-best successful recovery rate which is about 25%. It is because that the coverage of RRReqs in FaRM depends on the node density and the angle of recovery zone. For NSR, its successful recovery rate is only about 10% on average and decreases with increasing node mobility, which demonstrates the inefficiency of keeping a consistent local view using periodic local topology updates under dynamic topology changes.

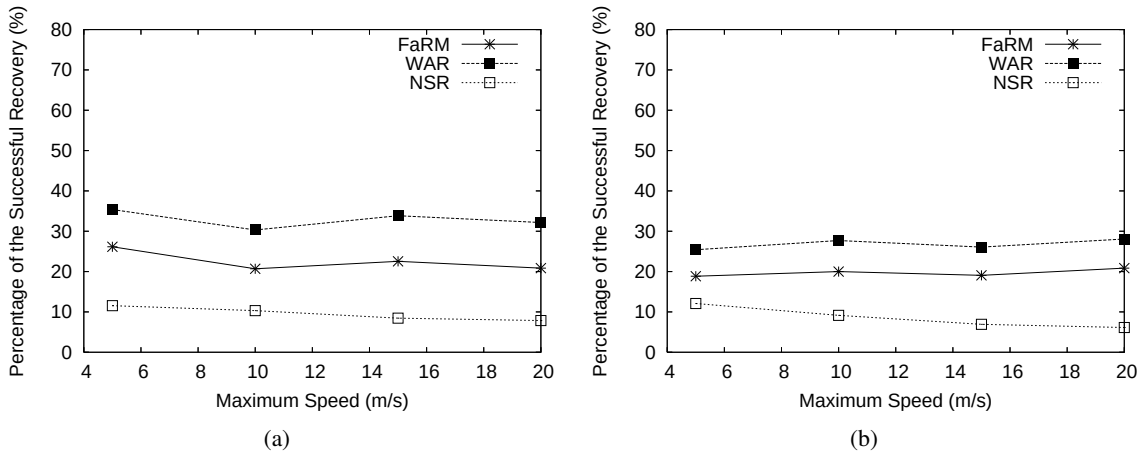


Figure 2.8: Usefulness: (a) Pause Time=10s (b) Pause Time=20s

2.3.3 Throughput and Overhead

In this experiment, we implemented two versions of FaRM: FaRM-DF uses density-first route selection technique and FaRM-SP uses shortest path routing. Both versions of FaRM uses local route recovery during their route maintenance. To evaluate FaRMs' performance, we used standard metrics *i.e.* throughput and overhead, and compared them with NSR, WAR, and DSR. 20 CBR connections with a rate of 2.5 KB/s were used in this experiment. Figure 2.9-a and b show results for the throughput of all five protocols under different scenarios in which the maximum mobility is varied between 5m/s to 20m/s while the pause time is kept constant as 10 seconds and 20 seconds. Among the five routing protocols compared, FaRM-DF outperforms other protocols. The throughput of FaRM-SP is lower than that of FaRM-DF especially under low node mobility, which indicates the density-first route selection technique is more efficient when mobility becomes a less dominating factor in a route's lifetime. WAR's average throughput is about 13.2KB/s, which is second to FaRMs' performance. The throughput of NSR and DSR are on average 15% lower than those of FaRMs and with increased node mobility, their performances degrade even more. The improvement of FaRM-DF's throughput over other protocols is due to its route selection and local recovery technique. Although the density-first route selection will introduce more overhead during route discovery, however, by choosing routes with higher density, it benefits route maintenance by avoiding transient routes and improve successful route recovery.

The routing overhead measures the routing related control packets. Since the overhead of WAR includes

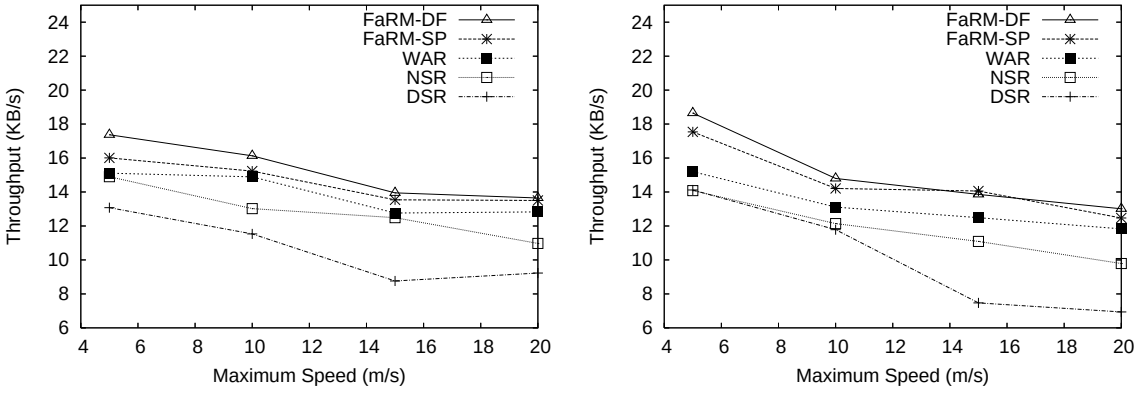


Figure 2.9: Throughput: (a) Pause Time=10s (b) Pause Time=20s

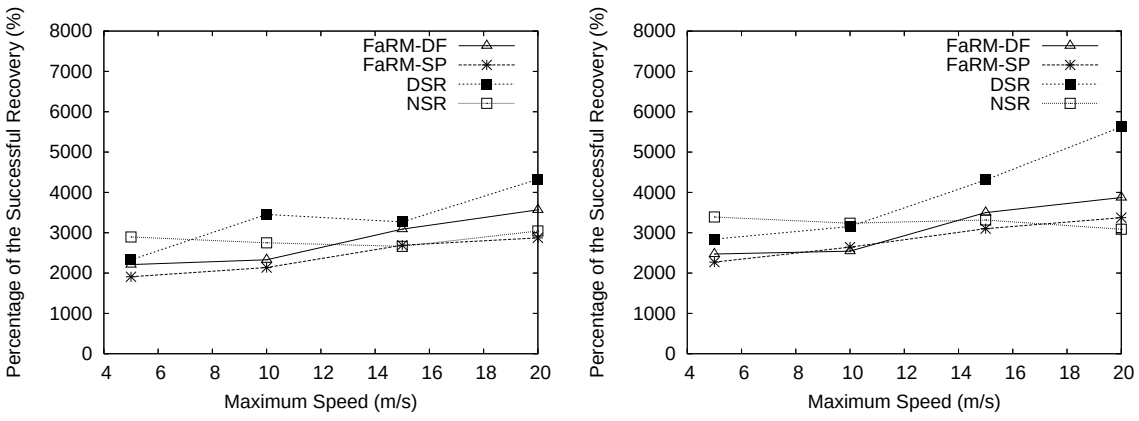


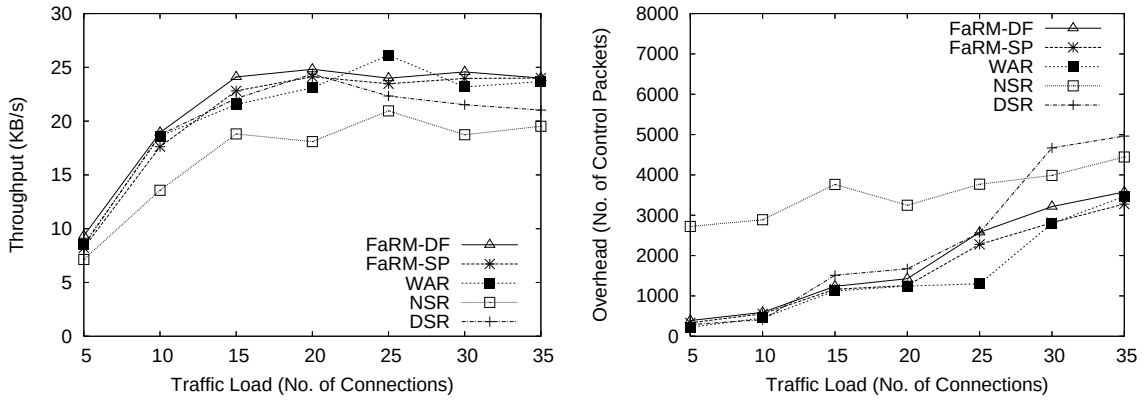
Figure 2.10: Overhead: (a) Pause Time=10s (b) Pause Time=20s

allocation of witness nodes in addition to control packets, we believe that it is unfair to compare WAR with FaRMs and other protocols just in terms of the number of control packets involved and hence, do not provide such a performance comparison. Figure 2.10-a and b show the overhead of the routing protocols by varying nodes' maximum moving speed with a pause time of 10 and 20 seconds. Since increased mobility causes more route breakups, the increasing trend of routing overhead can be observed for FaRMs and DSR as the maximum moving speed is increased from 5 to 20m/s. FaRM-SP shows the least routing overhead, which is at about 12.5 pkt/s on average. The reason of the improvement is because the route recovery avoids unnecessary source-initiated route discovery and the cooperative searching process based on cone-shaped recovery zone is very efficient. Furthermore, although FaRM-DF introduces more overhead during route discovery than FaRM-SP and DSR, its overhead is very close to FaRM-SP and much lower than DSR. This is because FaRM-DF is capable of selecting more robust routes which significantly reduces the unnecessary route discoveries. Due to periodic topology updates, NSR tends to have constant overhead; however, as the mobility increases, both throughput and usefulness of NSR degrades dramatically, which indicates its inefficiency in maintaining consistent topology information of demanding environments with high mobility.

2.3.4 Scalability and Stability

The scalability of the routing protocols is evaluated with different traffic loads. All cases use the same mobility as 10m/s maximum moving speed and 5s pause time. The traffic load is increased from 5 CBR connections to 35 CBR connections (each connection has a rate of 2.5KB/s). From Figure 2.11-a, FaRM-SP, FaRM-DF and WAR reach their highest throughput at about 25 and 30 connections; after these points, their throughput is almost constant. We believe that at those point the network becomes saturated, so more traffic generated to the network will not increase the system's throughput. A similar trend can be observed in Figure 2.11-b in which the throughput of FaRM-DF, FaRM-SP and WAR has an obvious increase after 25 connections. For NSR and DSR, the network becomes overloaded after 15 and 20 connections. The lack of scalability of DSR and NSR under the heavy traffic can be explained by the overhead generated from frequent route discovery and periodic topology updates.

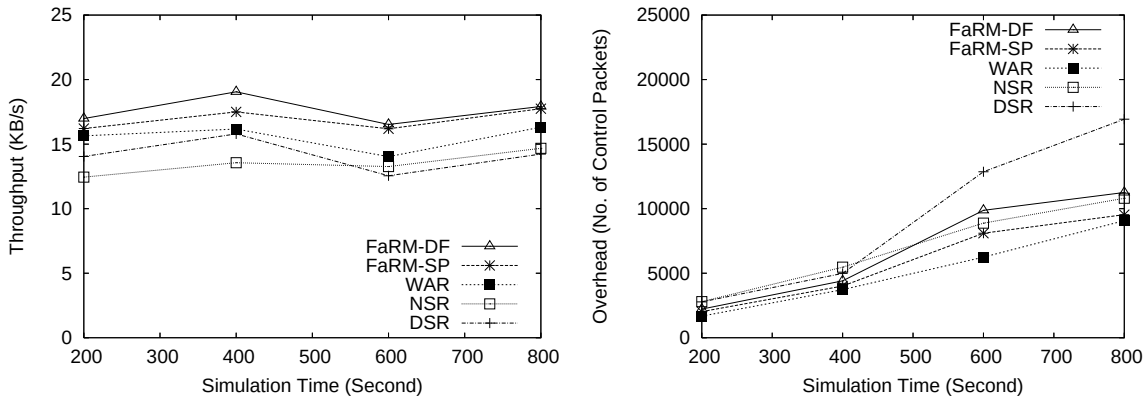
To evaluate a long time behavior of these routing protocols, we examine the throughput and overhead with different simulation times. The mobility model used in this experiment has 10m/s maximum moving speed and 5 second pausing time and the traffic consists of 20 CBR connections with 2.5KB/s rate. All of



(a) (b)

Figure 2.11: Scalability: (a) Throughput (b) Overhead

these protocols demonstrate good stability as shown in Figure 2.12-a and b. In Figure 2.12-a, all protocols have almost constant throughput at different time, while in Figure 2.12-b, all protocols show almost linear increases which means that a constant number of routing related control packet is generated during any time intervals.



(a) (b)

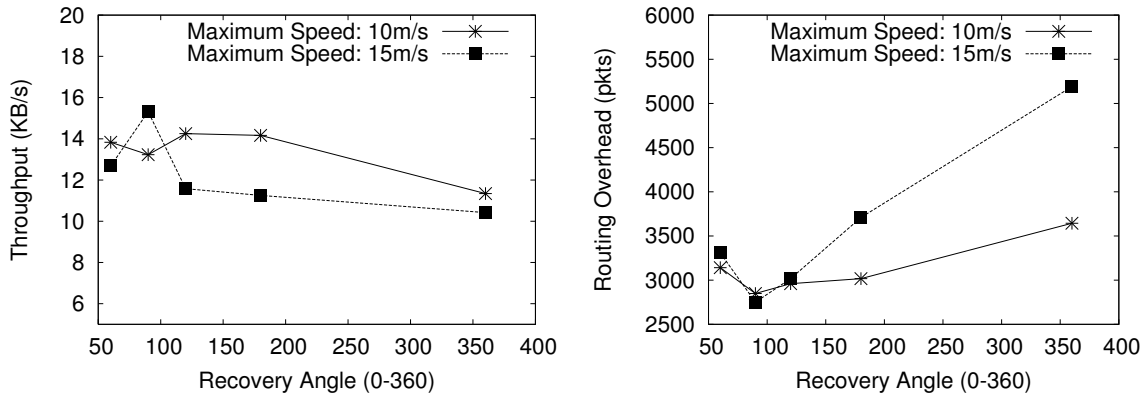
Figure 2.12: Stability: (a) Throughput (b) Overhead

2.3.5 Recovery Angle

The optimal angle for a cone-shaped recovery zone is obtained by balancing the cost of searching overhead and the benefit of successful route recovery. We vary the degree of the angle made by the cone-shaped recovery zone in the following increments: $\pi/3$, $\pi/2$, $2\pi/3$, π, and 2π. For each adjustment made, we

compared the overhead and throughput with the maximum moving speeds of 10m/s and 15m/s, and 20-second pause time.

In Figure 2.13-a, the throughput reaches its peak when the degrees of the recovery zone angle are increased to $2\pi/3$ and $\pi/2$ with maximum moving speeds of 10m/s and 15m/s, respectively. In both cases, the throughput is stable where the degree of the recovery zone is less than π, but it slightly declines when the angle is increased. A similar phenomenon can be observed in Figure 2.13-b. In essence, a smaller angle introduces less overhead, but has a lower self-recovery rate due to a smaller searching area; conversely, a larger angle introduces more overhead as it requires more nodes in the process of forwarding RRReqs, but obtains a better route recovery rate. Our simulation indicated that the optimal recovery zone angle is between $\pi/2$ and $2\pi/3$, with a maximum moving speed of 10 to 15m/s.



(a) (b)

Figure 2.13: Optimal Angle: (a) Throughput (b) Overhead

CHAPTER THREE

A MONITOR-BASED TRANSPORT PROTOCOL FOR WIRELESS SENSOR NETWORKS

3.1 Related Work

Transport protocols typically exist on top of the routing layer in the traditional network stack model. Such protocols provide multiplexing, rate control and congestion control on an end-to-end flow. Traditional TCP [48] is adequately efficient in reliable data delivery and congestion control by using an ACK- and AIMD-based mechanism; however, TCP is not suitable for WSNs for several reasons: (1) TCP is biased toward sensors that are closer to the sink because of the delays caused by ACK; (2) TCP's propagation of ACK packets incurs additional network congestion; (3) TCP's end-to-end recovery mechanism causes considerable delays and an excessive number of retransmissions; and (4) TCP misinterprets packet losses as a signal of congestion and thus, affects throughput in wireless networks. In order to provide a reliable data delivery within a bounded delay, several transport protocols for WSNs have been proposed and can be categorized as either downstream or upstream.

The downstream transport protocols usually provide packet-level reliability for delivering information pertaining to re-tasking, querying, or reconfiguration. Downstream protocols are similar to IP multi-casting, but are faced with the additional problems associated with WSNs, such as resource-constraints and unpredictable environments. In PSFQ [18], a packet is distributed from a source node by forwarding data at a relatively slow speed, but nodes that experience data-loss are allowed to aggressively fetch missing segments from immediate neighbors. PSFQ uses a NACK-based hop-by-hop data-recovery and hence, requires in-sequence data delivery. GARUDA [57] is another downstream transport protocol and is based on a two-tier, two-state loss-recovery method. In GARUDA, a core is constructed based on the *Minimal Dominating Set*, and caches packets by acting as a collection of loss-recovery servers.

The upstream transport protocols are designed to provide reliable delivery of sensory data from the event center (sensor) to the sink. In ESRT [52], authors propose an event-level reliability, which tolerates signal packet losses as long as the event fidelity can be achieved at the sink. ESRT also detects the current status of networks and uses an end-to-end rate control scheme based on broadcast. In RMST [23], authors proposed

a selective NACK- and timer-driven mechanism for both loss detection and notification; however, RMST is dependent on Directed Diffusion [12] and does not address reliability issues with congestion and unreliable sensors. RBC [32] uses a hop-by-hop loss recovery and a windowless block acknowledgment based on IACK (Implicit ACK) in order to reduce control-related overhead.

In addition to reliable data delivery, congestion control is another important aspect of WSN transport protocols. Authors of CODA [8] propose a complete congestion control scheme for transient and consistent congestion. The CODA scheme consists of an open-loop, hop-by-hop control for back-pressure of transient congestion, and a closed-loop, multi-source regulation for consistent congestion. Ee and Bajcsy discuss a congestion control mechanism from the perspective of fairness [21]. In their protocol, each node is assigned a fair rate based upon the respective node's routing tree; thus, an equal amount of data will be received from each sensor at the sink.

Due to redundant deployments of WSNs, *Topology Control* [7, 11, 28, 55] is generally used in link layer protocols and coverage algorithms in order to improve energy-efficiency. For example, ASCENT [11] uses sleep/wake-up scheduling to allow each sensor to determine its sleeping period based on the number of active nodes in the network and per-link data-loss rate. Hsin and Liu [28] propose a scheduling scheme based on low duty-cycle nodes to obtain energy-efficient coverage by activating only a minimal number of sensors. Our approach is distinct from the aforementioned research because it utilize a topology control technique to obtain improved reliability in data delivery.

3.2 Protocol Design

Traditional transport protocols for WSNs use hop-by-hop loss recovery and cache packets only at nodes that are involved in the forwarding process. Because sensors are resource-constrained devices, caching packets at forwarding nodes can cause congestion when the queue is almost full. Furthermore, this approach does not allow quick loss detection and recovery if the next-hop becomes unreachable due to congestions or sudden node-failures. To address these problems, we propose a reliable sensor-to-sink transport protocol that provides link-monitoring and packet-loss recovery by dynamically scheduling inactive nodes that are not involved in forwarding.

3.2.1 Design Considerations

Due to application requirements, traffic pattern, resource-constraints, and error-prone radio communication, following issues have to be considered during the process of designing an upstream data transport protocol for WSNs.

Types of Reliability. Unprocessed sensory data are generally considered to be redundant and correlated; thus, only event-level reliability is commonly considered for upstream data transport protocols. However, as in-network data aggregation techniques become more widely used in sensor networks, the redundant sensory data will be reduced. With that in mind, packet-level reliability is necessary for delivering aggregated data in WSNs.

Causes of Packet Loss. Four types of events can contribute to packet-losses: (I) link errors, (II) transmission conflicts, (III) queue overflow, and (IV) node failure. Link errors, such as self-interference and signal attenuation, can corrupt a packet at the receiver; conflicts may result from two neighboring nodes trying to transmit simultaneously; packets will be dropped at a receiver if the queue overflows due to congestion; and occurrences such as a power outage can lead to node-failure and cause the loss of all queuing packets. Generally, type I and II data losses caused by link errors and transmission conflicts are mitigated by physical and MAC-based layer protocols, whereas type III and IV losses resulting from events such as queue overflow and power outage can be more easily addressed by upper-layer protocols, such as transport protocols.

Loss Recovery. In traditional wired networks, end-to-end loss recovery is more efficient and reliable; however, due to the exponential accumulation of link errors (p^n where p is the link error rate and n is the number of hops), WSNs prohibit end-to-end recovery because of their error-prone wireless links. Therefore, hop-to-hop recovery is more commonly used for reliable data transport in sensor networks.

Loss Detection. Hop-by-hop packet detection can detect packet losses based on receivers, senders, or other non-forwarders (nodes that are not involved in forwarding). Receiver-based detection requires an in-sequence delivery thus, a packet loss is detected when an out-of-sequence packet is received. Sender-based detection uses acknowledgments (ACK/NACK) from receivers or by overhearing the retransmission of the packet at the receiver. A packet loss can also be detected by other non-forwarders. For example, in WAR [36], loss-detection at asymmetrical links is carried out by assigning a witness node to each link.

Scalability. Scalability is required for large-scale WSNs to provide reliable data transport among thousands of sensors. Traditional transport protocols rely on forwarding nodes to cache data for loss recovery; however, this approach is only effective in handling losses of type I and II. In the case of packet loss from congestion and sudden node-failures, this loss recovery scheme will cause severe congestion, delay, and unnecessary retransmissions by sending more packets to an unreachable next-hop. To address these issues, we propose a monitor-based approach to dynamically configure inactive nodes as monitors for enhanced scalability in heavy traffic and congestion.

3.2.2 Minimum Vertex Set Cover

To improve reliability in upstream data delivery, our approach dynamically configures the topology based on all active flows. As in Figure 3.1, data are generated at several event-centers, and all sensory data containing the correlated information are aggregated locally. To forward the aggregated data to the sink, a subset of inactive nodes are initiated as *monitors* to assist in loss detection and recovery. Our goal is to initialize a minimum number of monitors for energy-efficiency and keep all forwarding nodes monitored. As in Figure 3.1, since any inactive node can be selected as a monitor for its one-hop neighbors, {A, B, C, D, E} is a minimum set of monitors for monitoring three active paths. However, determining the minimum set of monitors for an arbitrary topology is non-trivial. We will formally define the problem as a *Minimum Vertex Set Cover* problem and prove its NP-completeness. Then, we will address the problem by presenting a greedy heuristic algorithm in order to solve it efficiently.

Definition 2 ***Minimum Vertex Set Cover (MVSC):*** *Given an undirected graph $G :< V, E >$ and a vertex set $V' \subset V$, a MVSC: V^* of V' is a subset of $V - V'$ such that (1) $\forall v \in V'$, $\exists u \in V^*$ and $uv \in E$; and (2) $|V^*|$ is minimum.*

The MVSC problem is to find a minimum set cover for a given set $V' \subset V$. It differs from the Dominating Set problem [44] in that a Dominating Set always exists with the size of $|V|$, but a MVSC only exists for a given vertex set V' with the size strictly less than $|V|$.

Definition 3 ***Vertex Set Subgraph (VSS):*** *Given an undirected graph $G :< V, E >$ and a vertex set $V' \subset V$, the corresponding $G_{VSS} :< V_{VSS}, E_{VSS} >$ is a subgraph of G such that (1)$V_{VSS} = V' \bigcup V''$ and $V'' = \{v | v \in V\text{-}V' \bigwedge (\exists u \in V', uv \in E)\}$; and (2) $E_{VSS} = \{uv | u \in V' \bigwedge v \in V'' \bigwedge uv \in E\}$.*

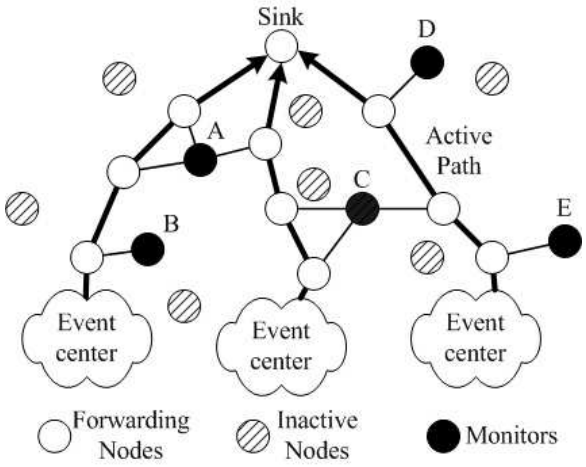


Figure 3.1: Monitor-Based Data Delivery

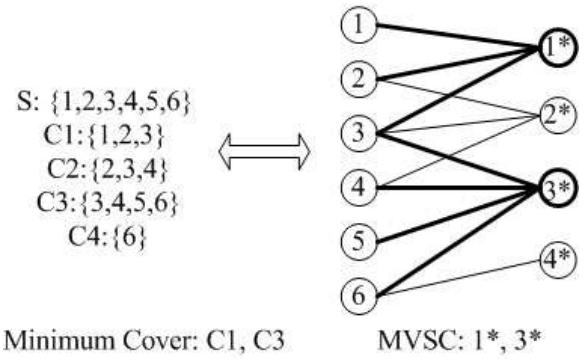


Figure 3.2: Deduction from Minimum Set to MVSC

By definition, a VSS of G is a bipartite graph with two partite sets of V' and V''. Next, *Lemma 4* shows that for any vertex set V', the problem of finding MVSC of graph G is equivalent to the problem of finding MVSC of G's corresponding Vertex Set Subgraph VSS.

Lemma 4 *Given a graph $G :< V, E >$ and a vertex set $V' \subset V$, G has a MVSC of size k if and only if its induced subgraph VSS has a MVSC of size k.*

Proof: For G:$< V, E >$ and its VSS:$< V_{VSS}, E_{VSS} >$, the sufficient condition holds because VSS is a subgraph of G. For the other direction, assume the MVSC of G is V_1 ($|V_1| = k_1$) and the MVSC of VSS is V_2 ($|V_2| = k_2$). If the necessary condition is not true, then $k_1 < k_2$. According to *Definition 2*, $V_1 \subset V_{VSS}$. Then, V_2 cannot be MVSC of VSS because $|V_1| < |V_2|$. Therefore, it is a contradiction.

Based on *Lemma 4*, we will prove that the problem of finding the MVSC of a vertex set $V'(V' \subset V)$ is NP-complete for a given graph $G :< V, E >$. ■

Theorem 5 *For graph $G :< V, E >$ and a vertex set $V' \subset V$, finding the MVSC of V' is NP-complete.*

Proof: According to *Lemma 3*, we only need to show that the problem of finding MVSC for subgraph $G_{VSS} :< V_{VSS}, E_{VSS} >$ is NP-complete. First, this problem is in NP because it can be verified in polynomial time. Next, we will show that finding MVSC is NP-hard by presenting a polynomial time reduction from the Minimum Cover problem [44] to this problem. Assume a set S and a collection of subset $C = \{C_i | C_i \subset S\}$. Then, we construct $V' = \{v | v = s \bigwedge s \in S\}$ and $V'' = \{v | v = i^* \bigwedge C_i \in C\}$. Therefore, $V_{VSS} = V' \bigcup V''$. Also, we construct $E_{VSS} = \{e | e = vi^* \bigwedge v \in C_i\}$ (e.g. Figure 3.2). According to our construction, if there is a Minimum Set C_{min}, then there exists a $V_{MVSC} = \{i^* | C_i \in C_{min}\}$. Similarly, in the other direction, if there is a MVSC V_{MVSC}, then there exists a Minimum Set $C_{min} = \{C_i | i^* \in V_{MVSC}\}$. Since the construction of this transformation can be finished in polynomial time ($O(n)$), the MVSC problem is NP-complete. ■

Because the problem of finding MVSC is NP-complete, we propose a greedy heuristic algorithm to efficiently obtain a nearly-optimal result. This algorithm will first construct the VSS based on V', then at each iteration, a vertex from V'' that can cover the maximum number of nodes in V' is selected until all vertices in V' have been covered (**Algorithm 1**).

Algorithm 1 A Centralized MVSC Heuristic Algorithm

1. Construct $G_{VSS} :< V_{VSS}, E_{VSS} >$ for V';
2. $V^* = \phi$, $V'' = V_{VSS} - V'$
3. Choose $v \in V''$ and v has maximum degree;
4. $V^* = V^* + \{v\}$;
5. $V'' = V'' - \{v\}$;
6. FOR each edge $uv \in E_{VSS}$
$V' = V' - \{u\}$,
$E_{VSS} = E_{VSS} - \{uv\}$;
7. IF ($V' \neq \phi$ and $V'' \neq \phi$)
GOTO 3,
else END.

3.2.3 Protocol Design

Monitor-configuration

Due to the resource-constraints and scalability issues in sensor networks, the centralized heuristic algorithm to find the MVSC (**Algorithm 1**) is infeasible in a real implementation; hence, we propose a distributed heuristic algorithm that dynamically configures monitors with only one-hop neighbor information (**Algorithm 2**). Algorithm 2 activates a minimal number of monitors when a new flow starts and de-activates the redundant monitors when a flow stops. The $Rank$ of each node is first initialized to 0. When an active flow starts on a path, all forwarders on that path will broadcast a $RankIncrement$ message to their one-hop neighbors. A $RankIncrement$ message carries a unique flow ID and any node receiving that message will increment its $Rank$ by one if the flow ID is new. Similarly, after a flow is finished on a path, all nodes on that path will broadcast a $RankDecrement$ message with a flow ID and any one-hop neighbors receiving that message will decrement their $Ranks$ by one if the flow ID is new.

In Algorithm 2, monitors will be initialized to cover the new flow as the first data packet is propagated to the sink; thus, there is no additional latency added to the process of monitor-configuration. At each node, Algorithm 2 can be summarized as follows:

1. If a data packet from a new flow is received and no monitor is available, go to Step 2); otherwise go to Step 4);

2. Identify the neighbor with the highest $Rank$ and request that neighbor to be initialized as a new monitor. The process of monitor initialization uses a two-way handshake with $Request$ and $Agree$

messages;

3. After a new monitor has been initialized, make an announcement of this new monitor by sending a $Notification$ message. Any neighbor that receives the $Notification$ message will update its $Rank$ or monitor if necessary;

4. Hand the data packet to its routing layer for forwarding.

Details on Step 2) and Step 3) are shown in **Algorithm 2**. In order to better illustrate this algorithm, an example is presented in Figure 3.3. In Figure 3.3-a, a new flow starts on the path $ABCDE$; there are 6 nodes in the path's vicinity with each node's $Rank$ indicated on the graph. When the first data packet propagates to node A, A will choose G as its monitor because G has the highest rank among all of A's neighbors. Then, A will send a $Request$ message to G. After G receives the $Request$ message, it will initialize itself as the new monitor and send an $Agree$ message to A (Figure 3.3-b). When A receives the $Agree$ message, it will record the identity of the new monitor and will then notify its neighborhood by broadcasting a $Notification$ message. If a forwarder (node B in Figure 3.3-c) that hasn't previously been covered by a monitor receives the $Notification$ message, node B will check whether it can be covered by the new monitor ("is G in my one-hop neighborhood?"). If so, node B will record G as its new monitor; otherwise, it will simply ignore the $Notification$ message. If a non-forwarder (node F in Figure 3.3-c) receives the $Notification$ message, it will decrement its $Rank$ by 1. After the entire neighborhood has been notified of the new monitor, the data packet will be forwarded to the next-hop, which is B. Since B has already recorded a monitor for itself (node G), it continues to forward the data packet without sending a new request. Then, after C receives the data packet, it chooses node J as its monitor (Figure 3.3-d) and broadcasts a $Notification$ message in order to update its neighboring nodes (Figure 3.3-e). By repeating the same process, all nodes on the path will be covered with a monitor as the first data packet reaches the sink.

Monitor/Sender-based loss detection

As discussed in Section 3.2.1, there are several drawbacks to sender- and receiver-based hop-by-hop loss detection. With those drawbacks in mind, our transport protocol combines the use of sender- and monitor-based detection. We categorize the packet losses into four types as in Section 3.2.1. Losses of type I and

Algorithm 2 A Distributed MVSC Heuristic Algorithm

Each node V do
IF (A data packet is received)
IF (It is from a new flow) **AND** (no monitor is available)
Choose an inactive neighbor V_{max} with max. $Rank$
Send $Request$ packet to V_{max}
Wait for $Agree$ packet till timeout
IF (Timeout)
Request for monitor is failed
ENDIF
Hand the data packet to the routing layer
ENDIF
IF (It is from an old flow) **OR** (a monitor is available)
Hand the data packet to the routing layer
ENDIF
ENDIF
IF (a $Request$ packet is received)
Record the sender's ID
Send $Agree$ packet
ENDIF
IF (an $Agree$ packet is received)
Record the Monitor's ID
Send $Notification$ packet with the Monitor's ID
Stop the timer
ENDIF
IF a $Notification$ packet of monitor V^* is received
IF (V is a forwarder) **AND** (V can be covered by V^*)
Send $Request$ packet
Wait for $Agree$ packet till timeout
IF (Timeout)
Request for monitor is failed
ELSE
record the new Monitor's ID
ENDIF
ENDIF
IF V is a non-forwarder
Decrement $Rank$
ENDIF
ENDIF

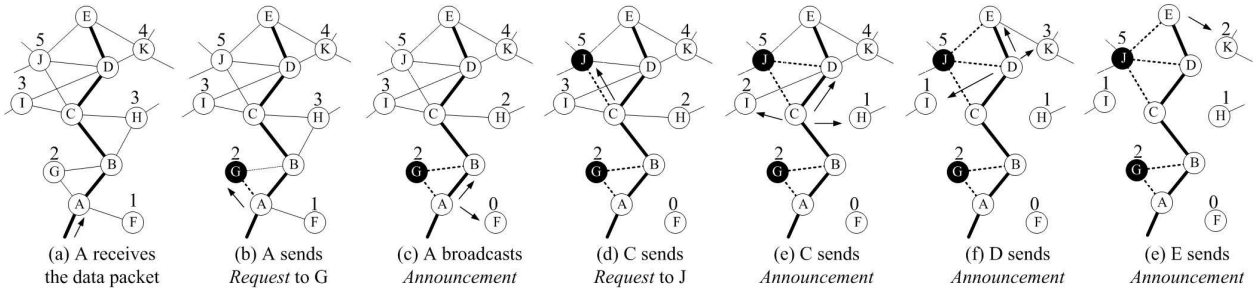


Figure 3.3: Monitor-initialization Processes

II can be detected from the lower-layer's feedbacks and; losses of type III and IV can be notified by using monitors to constantly overhear the shared radio for retransmissions. This combined approach allows for an easy identification of link failures without creating additional overhead and a quick response to losses caused by congestion or node failures. In short, the advantages of our approach include:

- Elimination of ACK/NACK-related overhead
- Out-of-sequence forwarding capability
- Robustness in the case of flows with small numbers of packets
- quick responses to various types of losses

Monitor-Aided Loss Recovery

If losses are caused by congestion or node-failure, retransmission of a packet to the same node is less likely to be successful, thus wasting energy and causing unnecessary delay. Instead of retransmitting an undeliverable packet to its original next-hop, our protocol forwards the packet to its recorded monitor. The monitor will then use *Geographic Routing* [3] to forward the packet to the node that is closest to the sink. The advantages of using monitor-aided loss recovery can be summarized as follows:

- Avoidance of failed nodes and quick adaptations to changes in topology
- diversion of traffic from congested "hot" spots
- independence from any underlying routing structures

3.3 Performance Evaluation

We used the *ns-2* simulator to implement our monitor-based transport protocol. To evaluate its performance, we also implemented three other related transport protocols which can provide packet-level reliability for sensor-to-sink data transport (Table 3.1). WAR [36] is originally designed as a routing protocol for mobile ad hoc networks with asymmetric links. WAR initializes a group of non-forwarding nodes as witness nodes, and each witness node overhears the data transmission on a link for loss detection and broadcasts the packet for loss recovery. Due to its similarity to our monitor-based approach, we implement WAR as a transport protocol for sensor networks. The hop-by-hop protocol uses the sender-based NACK mechanism for loss detection and retransmits the packet for loss recovery. Finally, the end-to-end protocol uses a TCP-like loss detection and recovery scheme. We didn't compare with other important upstream transport protocols, such as ESRT [52] and RMST [23], because they provide event-level reliability or are specifically designed for certain network architectures. Since our transport protocol focuses primarily on packet-level reliability for upstream data delivery, we use standard metrics, throughput and packet delivery rates, to evaluate their performances.

Our simulation consisted of three phases. First, we evaluated the performance of the four transport protocols under persistent congestion by continuously increasing the packet-generating rate at each source. We used 50 randomly-deployed sensor nodes, each with a maximum transmission range of 250 *meters* (m), and one fixed sink (coordinates: $(500m, 500m)$) in a $1000m \times 1000m$ grid. In the second phase, using the same topological configuration, we examined the scalability of the four protocols under transient congestion by increasing the number of event centers. In the third phase, using the same topology but, instead, with 200 sensor nodes, we compared the fault-tolerance of the four transport protocols by imposing randomly-faulty sensor nodes.

Table 3.1: Transport Protocols

Schemes	*Loss Detection*	*Loss Recovery*
Monitor	Sender&Monitor	Monitor-aided retransmission.
WAR	Witness	Broadcast-based
HopbyHop	Sender(NACK)	Hop-by-hop retransmission.
EndtoEnd	Sender(NACK)	End-to-end retransmission.

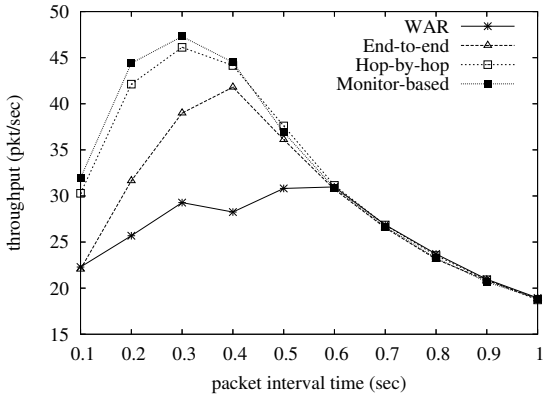


Figure 3.4: Throughput *vs.* Traffic Rate

3.3.1 Persistent Congestion

Figures 3.4 and 3.5 show the throughput and packet delivery rate of four transport protocols. We randomly chose 20 sources from 50 sensor nodes to generate CBR traffic. The interval time of the packet generator varied in each experiment. The size of each packet was 516 bytes, and rates of each connection ranged from 0.5KB/s (1-second intervals) to 5KB/s (0.1-second intervals). Figure 3.4 shows that throughput increases in all protocols when the traffic load increases (with no congestion). After the throughput reaches its peak (between 0.3 and 0.6 seconds intervals), it starts dropping dramatically due to congestion. Our monitor-based protocol proved to be more scalable than the other three protocols because congestion is not observed until the packet interval time is reduced to 0.3 seconds and, even after congestion appears, monitor-based protocol can still obtain a better throughput. This improved performance can be explained by the monitor's process of diverting part of the traffic to other under-loaded links. Furthermore, the hop-by-hop protocol out-performs the end-to-end protocol, which is due to its more efficient packet-recovery mechanism in error-prone wireless links. Lastly, the WAR protocol had the lowest throughput because of the significant overhead around the congested area caused by its broadcast-based recovery process. As shown in Figure 3.5, all protocols can achieve a delivery rate of nearly 100% with low traffic; however, the delivery rate dramatically drops when subjected to congestion. Our monitor-based protocol achieves the highest packet delivery rate among all other protocols even after being subjected to increasing congestion thereby proving its effectiveness in handling conditions of persistent congestion.

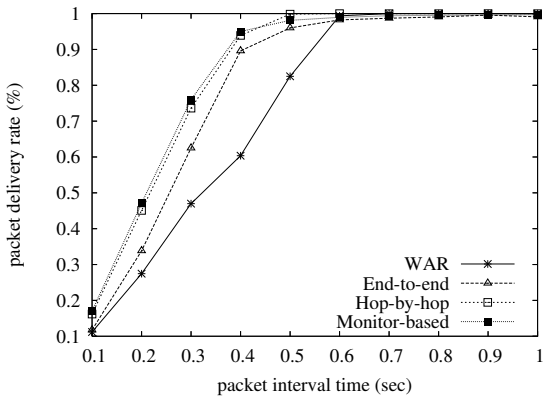


Figure 3.5: Packet Delivery Rate *vs.* Traffic Rate

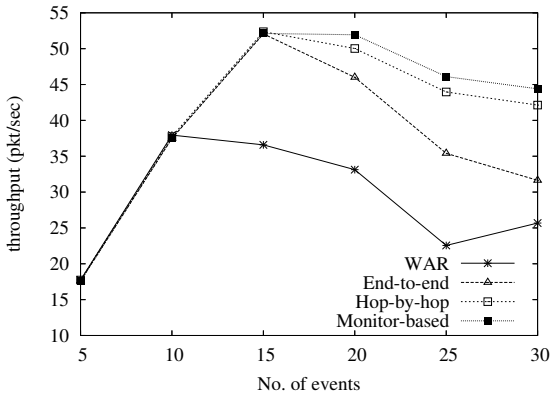


Figure 3.6: Throughput *vs.* Event Occurrence

3.3.2 Transient Congestions

To simulate transient congestions, we used short-term CBR flows with a rate of 2.5KB/s and increased the number of event centers from 5 to 30. In Figure 3.6, monitor-based, hop-by-hop, and end-to-end protocols reach their highest throughput at 15 connections; then, congestion causes throughput to decline. Conversely, WAR saturates the network with only 10 connections because of the significant overhead associated with its broadcast-based loss recovery. Moreover, after congestion starts to appear, the monitor-based protocol has higher throughput than the other three protocols. In Figure 3.7, a 100% delivery rate can be achieved for all protocols with sparse events happening ($\leq$ 10 event centers) whereas a sharp decline can be observed as more frequent events occur. However, the monitor-based protocol exhibits the highest packet delivery rate among all protocols.

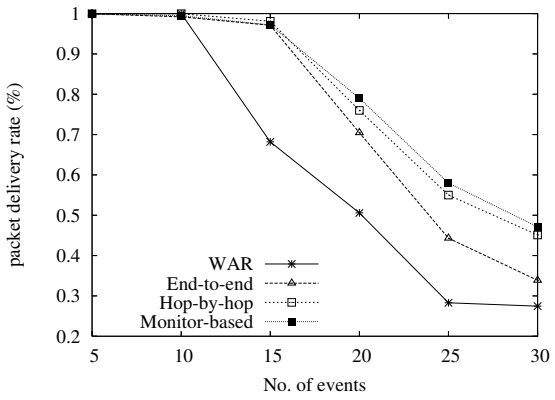


Figure 3.7: Packet Delivery Rate *vs.* Event Occurrence

3.3.3 Fault-Tolerance

Because sensor networks are generally deployed in harsh environments such as battlefields or natural disasters, fault-tolerance is vital in transport protocols. We examined the fault-tolerance of various protocols by randomly turning off sensors in a congestion-free environment (10 CBR connections at a rate of 0.5KB/s). Obviously, broadcast-based recovery in WAR is the most robust approach to provide fault-tolerance in congestion-free environments, so we only compared the fault-tolerance of monitor-based, hop-by-hop, and end-to-end protocols. Figure 3.8 illustrates how throughput among all three protocols decreases as more nodes are randomly turned off. Specifically, monitor-based protocol outperformed other protocols in handling such random failures with improved throughput. The monitor-based protocol's improved performance can be explained by its more effective detection of faulty nodes and its ability to forward undeliverable packets to nodes that are independent of the original routing structure. Furthermore, the monitor-based protocol also has the highest packet-delivery rate of all the protocols, as shown in Figure 3.9.

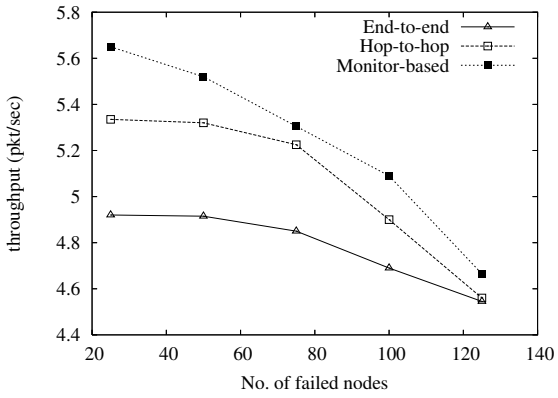


Figure 3.8: Throughput under Unreliable Sensors

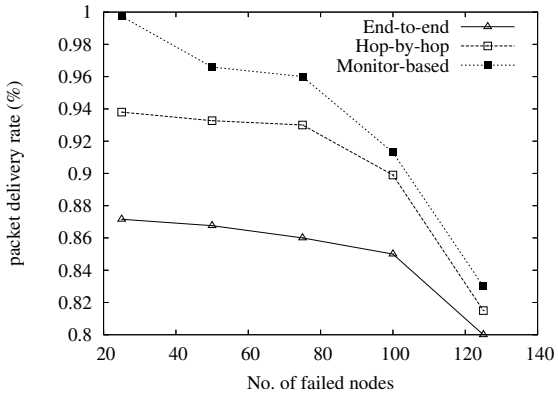


Figure 3.9: Data Delivery Rate under Unreliable Sensors

CHAPTER FOUR

ENERGY-EFFICIENT SENSING COVERAGE IN WIRELESS SENSOR NETWORKS

4.1 Related Work

Some of the earliest research on energy-efficient coverage was done by Slijepcevic and Potkonjak who proposed a NP-complete problem called *Set K-Cover* [55]. Their algorithm selects several mutually-exclusive sets of sensor nodes, with each set covering an entire area. Hsin and Liu [28] discuss node-scheduling and the tradeoff between random and coordinated sleep algorithms based on low-duty cycle sensors. Tian and Georganas [59] proposed a scheduling mechanism wherein a node becomes active only if its "sponsored area" is covered by its neighboring nodes; and TGim [24] extended Tian and Georganas' mechanism by considering the realistic signal propagation model. Zhang and Hou discuss a triangulation-based coverage, where equilateral triangulation achieves the best energy-efficiency in sensing [34]. Khan *et al.* proposed a Mobile Traversal Algorithm to form an equilateral triangulation-based coverage using mobile sensors [1]. Coverage has been approached from different perspectives by Meguerdichian *et al.* [56], who developed a technique that sought maximal breach and maximal support paths by using the Voronoi Diagram.

The use of *k*-coverage to improve accuracy and fault-tolerance is examined by various other researchers [25, 31, 68, 73]. Huang and Tseng developed a sufficient and necessary condition for *k*-coverage [31]. In order to configure the networks for *k*-coverage, Wang *et al.* proposed an eligibility algorithm to determine whether it is necessary for certain nodes to become active [68]. Gupta, Zhou, and Das designed a greedy *k*-coverage algorithm based on the "K-Benefit" value of each candidate path [73]. Hefeeda and Bagheri illustrated the same problem as a set system in which optimal hitting sets correspond to optimal *k*-coverage solutions [25].

Recently, researchers have recognized the need to develop integrated approaches for both coverage and connectivity. PEAS [71] addresses that challenge by using a "probabilistic probing." Shakkottai, Srikant, and Shroff [54] examine the probabilistic bounds that both coverage and connectivity can be attained if unreliable sensor nodes have been deployed on a given grid. Wang *et al.* [68] show that full coverage implies connectivity if the transmission radius is at least two times greater than the sensing radius.

To further reduce redundant coverage, sensors with variable-sensing radii have been used. Wu and

Yang [70] propose a coverage algorithm that uses sensors with maximum, medium, and small-sensing radii according to network topology. Cardei *et al.* [10] propose an algorithm and a model of continuously-adjustable sensing radii in order to identify mutually-exclusive sensor covers with optimal sensing ranges. Other approaches to this problem include those by Dhawan *et al.* [20] and Zhou *et al.* [75] based upon Linear Programming and the Voronoi Diagram, respectively.

4.2 *1*-Coverage with Uniform Sensing Radius

4.2.1 MAX-k-Covered Mesh

Given an area with redundantly deployed sensors, there is a subset of sensors that can provide full coverage with minimum overlap. Suppose all sensors have a uniform sensing radius R, the lower-bound of energy consumption for full coverage of the given area is obtained when active sensors formalize an equilateral-triangular mesh where each triangle has its sides as $\sqrt{3}R$ [68]. To guarantee full coverage of a given area with circles, there are certain points that have to be redundantly covered by more than one sensor. We call an area to be MAX-k-covered if any point in this area is covered by no more than k sensors. Obviously, the area has to be at least MAX-3-covered in order to maintain full coverage.

The equilateral-triangular mesh has minimum overlapping coverage among all configurations of MAX-3-coverage, thus, it has the lower-bound of energy consumption (Figure 4.1-a). All intersecting points of circles in the equilateral-triangular mesh are covered by three sensors. For an unbounded area, the redundantly covered area is $\frac{2\pi}{3\sqrt{3}} - 1$ (about 21%). Similarly, the square mesh is MAX-4-covered (Figure 4.1-b) and the redundantly covered area is $\frac{\pi}{2} - 1$ (about 57%). However, during the random deployment of sensor networks, it is not possible to achieve the lower-bound of energy consumption by building the equilateral-triangular mesh. Furthermore, even if such a mesh exists, a continuous surveillance will overburden those selected sensors. Once those sensors die, the performance will be significantly degraded. Therefore, we are proposing a dynamic mesh building mechanism which can

- quickly construct the mesh in a distributed fashion,
- periodically rotate active nodes for energy-balancing, and
- approximate the equilateral-triangular mesh or square mesh based on the local topology.

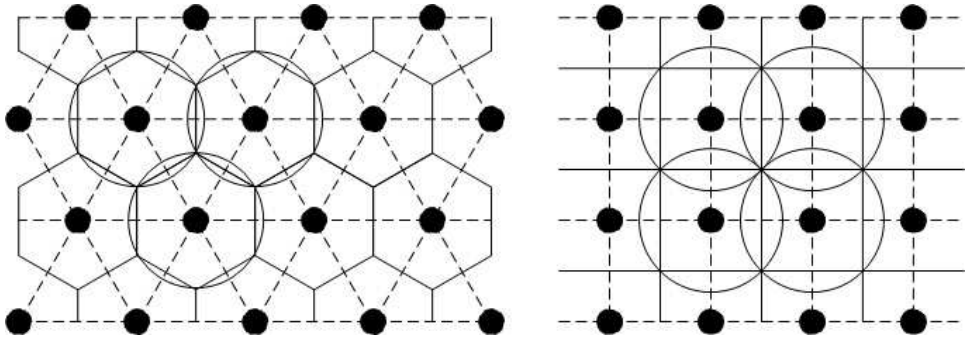


Figure 4.1: (a) Equilateral-Triangular Mesh (b) Square Mesh

4.2.2 Gossip-Based Mesh Construction

Our gossip-based mesh construction process dynamically builds a topology based on an equilateral-triangular or square mesh. In order to build a mesh that is self-adaptive to random topologies and capable of energy-balancing, a static mesh and a virtual mesh are used.

A static mesh is built based on an equilateral-triangular mesh or square mesh. Assume R is the sensing radius of all sensors. For an equilateral-triangular mesh, the target area is first statically divided into hexagons with each side as R. If a square mesh is used, the area is divided into squares with each side as $\sqrt{2}R$. Two cells are adjacent if they share a common side and two cells are neighbors if they share a common vertex or side. Each cell in an equilateral-triangular mesh has 6 adjacent cells and 6 neighboring cells, and each cell in a square mesh has 4 adjacent cells and 8 neighboring cells.

The sensors are clustered based on the static mesh. Within each hexagon or square cell, a sensor is selected as a cluster head based on its power and service time. The cluster head can communicate with the rest of the sensors in its cell and the cluster heads of its adjacent and neighboring cells.

The work proposed previously by [33, 70] uses a similar static mesh to activate sensors that are closest to the centroid of all cells. In order to provide the self-adaptiveness to local topology and achieve energy-balancing among all sensors, a virtual mesh is used to select appropriate sensors for each static cell. At the beginning, a sensor is randomly selected as an initiator. Once the initiator becomes active, it computes its virtual cell centered at its own location and its adjacent virtual cells. Then, the initiator requests the cluster heads of its adjacent cells to activate their own sensors based on the initiator's virtual mesh. If a sensor is already activated in an adjacent cell, the cluster head of this cell ignores the request. Otherwise,

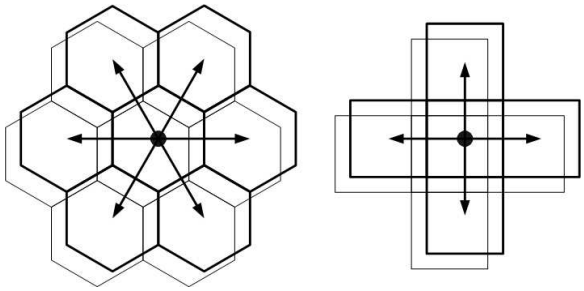
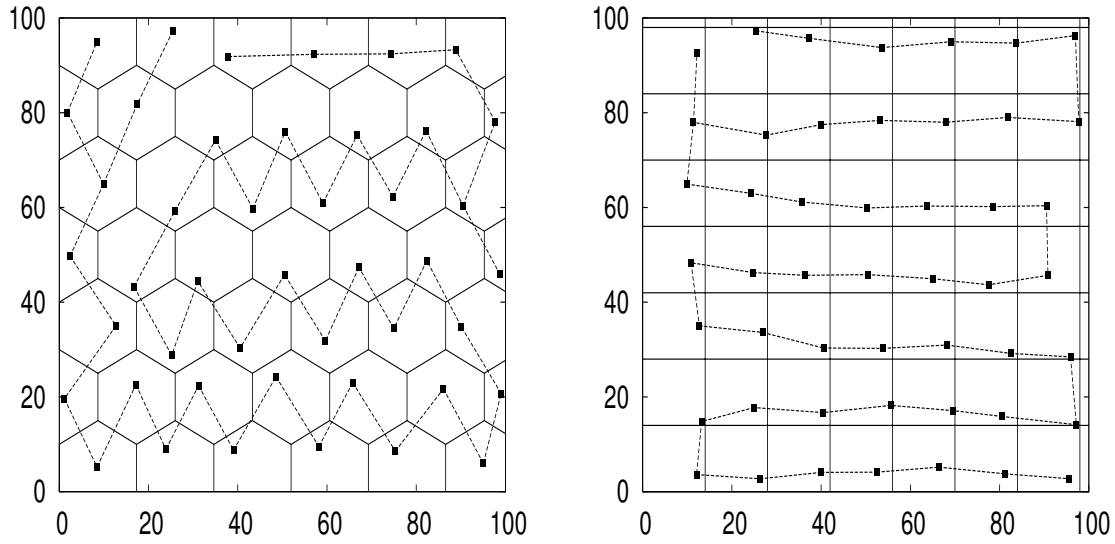


Figure 4.2: Static and Virtual Mesh

the cluster head activates a sensor that is closest to the centroid of the corresponding virtual cell. Once a node becomes active, it builds its own virtual mesh and starts the same process to activate new sensors for its adjacent cells. As in Figure 4.2, the light lines indicate the static mesh. Once a sensor is selected as an active node, it automatically obtains its virtual mesh (dark lines) based on its location. The shifting between a sensor's virtual cell and static cell is a local correction that will be applied to the newly activated sensors in order to maintain the property of the equilateral-triangular mesh or the square mesh. As the activation chain continues, the final mesh is constructed through the gossip propagation process where an erroneous shifting will be corrected at the next stage with a new virtual mesh.

As shown in Figure 4.2, the construction process of a square mesh offers the advantage of lower communication overhead than that of an equilateral-triangular mesh. This is due to the fact that each equilateral-triangular cell needs to communicate with 6 adjacent cells whereas a square cell needs to communicate with only 4 adjacent cells. Furthermore, an equilateral-triangular mesh is more complex than a square mesh, thus it is more likely to introduce erroneous shifting when the node density is low. However, the equilateral-triangular mesh has a much lower redundant coverage and much faster convergence time ($O(log_6^N)$) than the square mesh ($O(log_4^N)$).

Figure 4.3 demonstrates two actual construction cases generated with 500 random nodes in a $100m \times 100m$ square. All nodes have $10m$ sensing range and hence, the hexagon cells and square cells have their sides as $10m$ and $10\sqrt{2}m$ respectively. In each static cell, there is one sensor activated (the black point). The line between two active sensors indicates the activation chain. As shown in Figure 4.3, the final topology of active sensors can absorb the erroneous shifting and maintain a good mesh structure.



(a) (b)

Figure 4.3: Gossip-Based Mesh Construction (a) Equilateral-Triangular Mesh (b) Square Mesh

4.2.3 Coverage Improvement

If no sensor can be found at a selected location, the sensor that is closest to this location will be chosen and thus, an erroneous shifting will occur. An erroneous shifting causes active sensors in adjacent static cells to have disjointed virtual cells and therefore, may introduce uncovered holes. To provide full coverage, a hole detection and recovery mechanism is used as the second phase after the mesh construction is finished.

Hole Detection

To detect uncovered areas, we use a boundary detection technique proposed by Carbunar *et al.* [9]. The authors proved that an inner hole exists at those nodes such that their Voronoi cell cannot be completely covered by their sensing range. In our previous work [65], we developed a distributed algorithm which allows each sensor to independently compute its own Voronoi cell with one-hop neighbor information. For each static cell, its active sensor's Voronoi cell is computed. Once an active sensor detects that its coverage disc can not cover its Voronoi cell, the hole recovery phase will be initiated.

Hole Recovery

In *Theorem* 6, if an uncovered hole has been detected, there must be two virtual cells that are both neighbors and disjointed. Our hole recovery technique is used to find those virtual cells and activate an additional

sensor to cover the hole. In the ideal scenarios, 3 neighboring cells will join at one vertex for an equilateral-triangular mesh and 4 neighboring cells will join at one vertex for a square mesh. Due to the random topology, the neighboring cells may overlap or disjoint with each other. Since only disjointed cells may cause holes, an elimination process is applied to those cells that share a common vertex. After the elimination, an uncovered area is approximated as a polygon by computing the convex hull based on the vertices of the remaining virtual cells and then, a sensor will be activated at the centroid of the polygon.

Theorem 6 *If no two neighboring sensors have their virtual cells disjointed, the area is fully covered.*

Proof: Suppose the area is not fully covered, there is an uncovered point A. Since all virtual cells of adjacent sensors are either overlapped or jointed, A must reside in one of the virtual cells V_A. Therefore, A must be covered by the active node in V_A, which is impossible. Thus, the area is fully covered. ■

Figure 4.4 illustrates the process of recovering the uncovered holes under a square mesh. Once node 1 detects a hole using the technique described in Section 4.2.3, it starts checking each vertex of its virtual cell. For node 1's vertex V_1, node 2, 3, and 4 are its neighboring nodes and V_2, V_3, and V_4 are the corresponding neighboring vertices. If there is no disjointed cells at V_1, according to *Theorem 1*, the hole can not occur near V_1. For example, in Figure 4.4-a, any vertices $V_i(i \in 1..4)$ is contained by at least one of its neighboring cells $j(j \neq i \bigwedge j \in 1..4)$. In this case, no additional sensor is required to be activated. Otherwise, a hole may occur. To approximate the hole area, node 1 eliminates those neighbor vertices V_i that are contained by other cells $j(j \neq i)$. Then, the remaining vertices must have disjointed cells. To approximate the hole, a convex hull is constructed based on the remaining vertices and a sensor is activated at the centroid of the convex hull. For example, Figure 4.4-b illustrates four disjointed cells after the elimination and a quadrangle is computed. In Figure 4.4-c and -b, three and two disjointed cells remain, and thus a triangle and a segment are generated respectively. The hole recovery process of the equilateral-triangular mesh is similar to that of the square mesh except that each vertex in the equilateral-triangular mesh has three neighboring cells.

4.2.4 Energy-Balance

The performance of WSNs relies on a collective work of all sensors, therefore, the energy-balance has become a major concern of design. In our protocol, each cluster head is selected for communication between its adjacent and neighboring cells during the mesh construction and coverage improvement processes, therefore

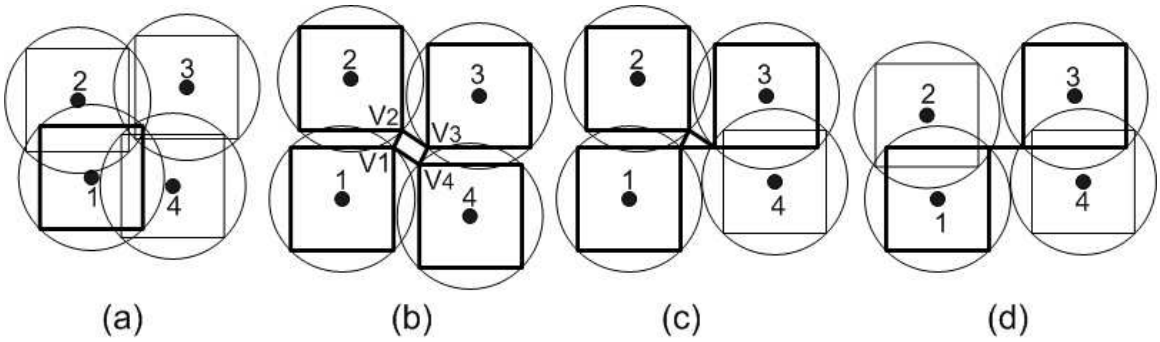


Figure 4.4: Hole Recovery

they require more energy consumption. To balance the energy consumption spent on cluster heads, cluster heads are rotated periodically and a sensor with higher energy storage and less service time is preferred to be selected as the new cluster head.

Sensing consumes energy at those active sensors. If the mesh is static, a small number of selected sensors will become overburdened and die quickly. Therefore, the mesh must be rebuilt periodically to avoid using the same set of nodes all the time. In our protocol, a new initiator is randomly selected to rebuild the mesh periodically. Since the new mesh solely depends on the local topology and the location of the initiator, different sensors will be selected during each period.

A strictly global synchronization is not required for the cluster head rotation and periodic mesh reconstruction processes. Once a sensor is selected as a head, it sets up a timer as its service period. The head broadcasts a service-expired message to its static cell when the timer expires. A sensor receives this message and responds with a rank calculated with its cumulative service time and energy storage. After gathering all ranking information of sensors in its cell, the cluster head decides the new head with a new service time. For mesh reconstruction, once a sensor has been activated, it will become active for at least one mesh-rebuilding-period. After that period expires, the sensor uses the current period number to decide if it should become the initiator for the next period. If so, a new mesh construction will be initiated; otherwise, it remains active until a new active sensor is selected for its cell.

4.3 *1*-Coverage with Variable Sensing Radii

With sensors capable of adjusting their sensing radii, energy consumption can be further reduced by selecting an appropriate size for each sensor's sensing disc based on the network topology. This section introduces a

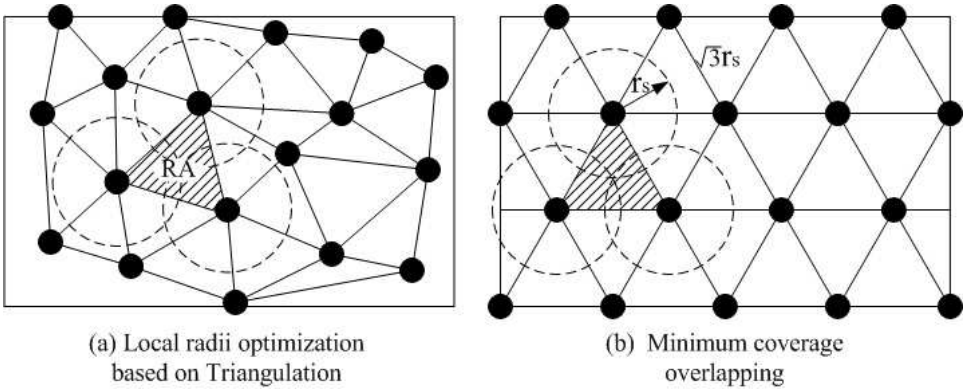


Figure 4.5: Triangulation-based Sensing Radii Optimization

technique to optimize sensing radii according to a Delaunay Triangulation of the network.

To optimize sensing radii locally, a triangulation is used to construct a planar graph based on sensor deployment. A *Responsible Area (RA)* is represented by a triangle in the graph (Figure 4.5-a). Each triangularly-shaped *RA* requires coverage to be provided by a single sensor at each of the three vertices of the triangle, for a total of three sensors per *RA*. With the triangulation available, the sensing radius is first optimized locally for each *RA* in order to ensure the local coverage. Then, each sensor will choose the maximum radius that can satisfy all of its adjacent *RAs* to provide a collective coverage of whole area.

To optimize sensing coverage locally based on each *RA*, triangulation is essential. Zhang and Hou [34] prove that the minimal redundant coverage is obtained with equilateral triangulation, where each edge is $\sqrt{3}r_s$ and sensors have an identical sensing radius as r_s (Figure 4.5-b). That ideal arrangement requires consistent distances between adjacent nodes, which may not be possible due to the random deployment of sensors. With that in mind, we chose Delaunay Triangulation to achieve a nearly-optimal result for energy-efficient coverage.

4.3.1 One-Hop Approximation of Delaunay Triangulation

Delaunay Triangulation (DT) [13] is the dual of the Voronoi Diagram and is known to exhibit the following characteristics:

- "Fat triangles," in the sense that the minimum angle of any Delaunay triangle is as large as possible;
- The Empty Circle Property, defined as a circle that runs through the vertices of any triangle with no

other vertex inside the circle.

Li *et al.* [39] and Liebeherr, Nahas, and Si [40] discussed distributed algorithms to compute DT based on the T-spanner graph and the locally equiangular property; however, their method posed certain limitations. Their algorithms required sensors to have knowledge of nodes located multi-hops away, or had a slow convergence time. Therefore, we proposed a simple, distributed algorithm to locally approximate DT using one-hop neighbor information.

Our one-hop approximation of DT is based on the centralized edge-flipping algorithm whereby all non-locally Delaunay edges are flipped to become locally Delaunay [13]. In our algorithm, each node maintains a list of its one-hop neighbors ($NeighborList$). After an arbitrary triangulation is constructed (Figure 4.6-a), each node independently tests its adjacent triangles to determine whether they all satisfy the Empty Circle Property. If an adjacent triangle cannot satisfy the Empty Circle Property, the corresponding edge is flipped. For example, in Figure 4.6-b, $\triangle ACD$ is, at first, a non-Delaunay Triangle because point B lies inside of $\triangle ACD$'s circumcircle. Then, AC is flipped to DB, and point C is deleted from A's $NeighborList$. The result is the formation of the Delaunay Triangle $\triangle ABD$ as shown in Figure 4.6-c. In Figure 4.6-d, $\triangle ADE$ is identified as a non-Delaunay Triangle and, similarly, to make the conversion, AE is flipped to DF and point E is eliminated from A's $NeighborList$. The final result is the creation of $\triangle ADF$, with no other points located inside its circumcircle as shown in Figure 4.6-e. The edge-flipping process continues until A's adjacent triangles can all be classified as Delaunay Triangles (Algorithm 3).

Algorithm 3 Construction of Delaunay Triangles

Each node N^* in the network, *DO:*
  Create $NeighborList(NL)$ of N^*'s one-hop neighbors;
  Initialize $TriangleList(TL)$ based on NL;
  While ($\exists\triangle_i$ in *NL*) and ($\triangle_i$ is not a Delaunary Triangle)
    Update TL by flipping the corresponding edge;
    Eliminate the unused point from NL;
  ENDWhile.

The one-hop approximation of DT can be implemented easily on sensors with low communication and computation overhead; however, with only one-hop information, the resulting triangulation may differ from the traditional DT. *Theorem* 7 shows that our local approximation of DT is equivalent to the traditional DT, provided that: (1) the area can be completely covered by the maximum sensing radius; and (2) the sensors

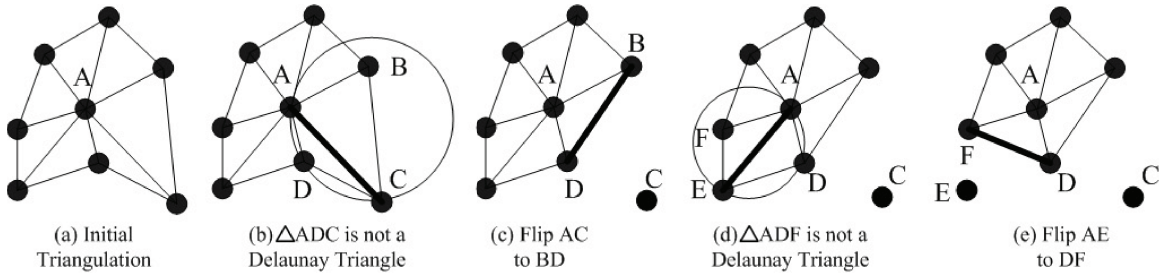


Figure 4.6: Construction of DT Based on One-hop Neighbors

satisfy $2R_s \leq R_x$, where R_s and R_x represent the maximum sensing radius and the maximum transmission radius, respectively.

Theorem 7 *If a two-dimensional area $\Re^2$ is completely covered by sensors with a maximum sensing range of R_s and a transmission range of R_x ($R_x \geq 2R_s$), the one-hop approximation of Delaunay Triangulation is the same as the traditional Delaunay Triangulation.*

Proof: According to *Theorem* 9, if $\Re^2$ is completely covered by a uniform maximal sensing range, it can be covered by the Delaunay-Triangulation-based heuristics too. Hence, each $Responsible\ Area$ (triangle) is covered by its three adjacent vertices with sensing radii less or equal to R_s. Obviously, the longest edge of any triangle e_{max} cannot exceed $2R_s$, and according to $R_x \geq 2R_s$, it means $e_{max} \leq R_x$. Because the distance between any two adjacent nodes in Delaunay Triangulation is within the transmission range of R_x (in other words, they are actual one-hop neighbors), the approximation using the one-hop neighbor generates the same triangulation as traditional Delaunay Triangulation. ■

If the first condition (complete coverage) does not hold, which means there exists some *natural holes* that can not be covered, the heuristics based on one-hop approximation of Delaunay-Triangulation may cause a different radii assignment. For example, in Figure 4.7-b, B and C are out of transmission range of A, so A will be unaware of these two points when generating its approximation of Delaunay Triangulation. As a result, instead of taking $\triangle_{ABC}$ as its $Responsible\ Area$ (there is a circle passing through A, B, and C, and no other point is inside of the circle), it chooses $\triangle_{ADE}$ and hence creates a smaller sensing radius.

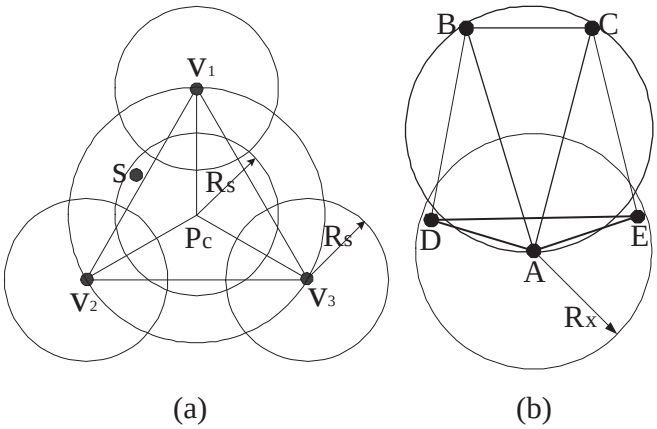


Figure 4.7: (a) Complete Coverage (b) Coverage with Natural Hole

4.3.2 DT-Based Sensing Radii Optimization

We use a local heuristic to optimize sensing radii based on each triangle. *Theorem* 8 uses a quadratic energy model and shows that the energy consumption of the three sensors at the vertices of each triangle is minimized when their sensing discs intersect at the circumcenter of the triangle. The same principle can also be applied to other energy models.

Theorem 8 *For a quadratic sensing energy model (kr_s^2 where r_s is the sensing radius and k is a constant), to cover a triangle by its vertices v_j:$(x_j, y_j), j\in\{1..3\}$, the energy consumption is minimum if the sensing discs of the vertices intersect at c : $(\sum_{j=1}^{3} x_j/3, \sum_{j=1}^{3} y_j/3)$.*

Proof: Since the triangle is completely covered by its adjacent vertices, it is necessary for all three sensing discs to intersect at one point for minimal energy consumption; otherwise, by shrinking the sensing disc we can get a sensing radius assignment with less energy consumption.

Suppose three sensing discs intersect at c with coordinates as (x_0, y_0). The total energy consumption is

$$E = \sum_{j=1}^{3} k \cdot r_j^2 = k \cdot (\sum_{j=1}^{3}((x_0 - x_j)^2 + (y_0 - y_j)^2))$$

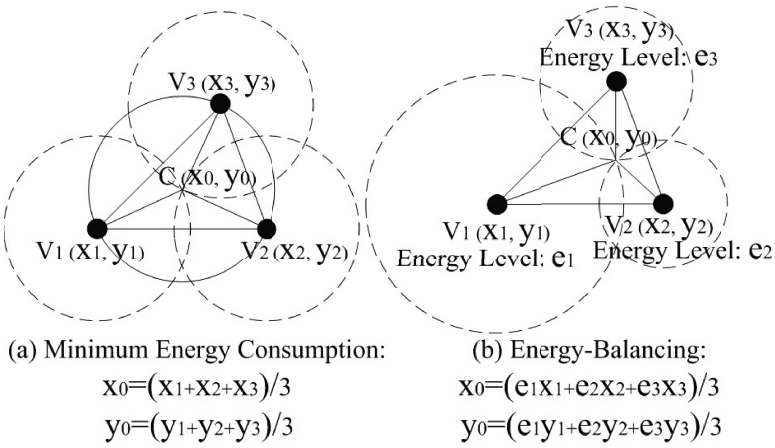


Figure 4.8: Local Optimization for Sensing Radii

To find the minimal value of E, we have $\frac{\partial E}{\partial x_0} = 0$ and $\frac{\partial E}{\partial y_0} = 0$, which give us $x_0 = \sum_{j=1}^{3} x_j/3$ and $y_0 = \sum_{j=1}^{3} y_j/3$. Hence, the optimal sensing radii for vertices v_j should be $\delta(v_j, c)$. ■

Based on *Theorem* 8, our first heuristic is to minimize the total energy consumption, which is specified as follows:

- *Step 1:* Each sensor calculates the optimal sensing radii for each of its adjacent Delaunay Triangles;
- *Step 2:* Each sensor chooses the largest optimal sensing radius among those calculated in *Step 1*.

In *Step 1*, three adjacent nodes collaborate to determine the optimal radii for their Delaunay Triangle based on *Theorem* 8 as shown in Figure 4.8-a. In *Step 2*, each sensor independently adjusts its sensing radii in order to maintain full coverage of all of its adjacent triangles. If, incidentally, the triangulation is equilateral, optimal coverage for energy-efficiency can be obtained [34]; otherwise, due to the "Fat Triangle" characteristic of DT, a nearly-optimal assignment for the sensing radii can be achieved.

For sensor networks, surveillance and data gathering require collective efforts among all sensors. However, the energy storage of each sensor may vary due to environmental conditions, antenna positions and traffic load. By assigning different sensing areas according to the sensor's energy level, sensors with less power can reduce their power consumption in sensing and communication, and hence extend the lifetime of networks for long-term services.

Similar to the first heuristic illustrated in Figure 4.8-a, our second heuristic, shown in Figure 4.8-b, incorporates energy-awareness to sensor with variant energy storage. In step 1, we choose the center of gravity as the intersecting point of three vertices' sensing discs. Therefore, the one with the higher energy level will be assigned with a larger sensing radius for more area to cover. Suppose a triangle $\triangle_i$ has three vertices v_j at $(x_j, y_j), j \in \{1..3\}$ and the energy level of each vertex is $e_j, j \in \{1..3\}$. Then, the center of gravity is at c_g with coordinates (x_0, y_0), where $x_0 = \sum_{j=1}^{3}(e_j \cdot x_j) / \sum_{j=1}^{3} e_j$ and $y_0 = \sum_{j=1}^{3}(e_j \cdot y_j) / \sum_{j=1}^{3} e_j$, and the sensing radius of v_j is $\delta(v_j, c_g)$. For example in Figure 4.8-b, v_1 and v_2 has the highest and lowest energy level, hence the assignment gives v_1 and v_2 the largest and smallest sensing discs respectively.

Full coverage is a fundamental requirement for reliable surveillance. In *Step 2* of our heuristics, local coverage is attained by selecting the largest optimal radius among all adjacent triangles; however, that selection does not necessarily ensure full coverage across the target area. *Theorem* 9 shows that our DT-based radii optimization can guarantee full coverage if there is no "hole" in the initial deployment.

Theorem 9 *If a two-dimensional area $\Re^2$ is fully covered by nodes with a maximum sensing range R_s, it is still fully covered after applying the Delaunay-Triangulation-based heuristic for sensing-radii optimization.*

Proof: If it is not a complete coverage using Delaunay Triangulation Based Coverage Heuristics, $\exists p$ and $\exists \triangle_i$, such that p is an uncovered point and $p \in \triangle_i$. Since p is an uncovered point in $\triangle_i$, the center p_c of $\triangle_i$ is not covered either, according to our heuristics (Algorithm 3). Hence, $|v_1 p_c| > R_s$. Because it is a complete coverage using uniform maximal sensing range, p_c can be covered by some sensor s. Hence, $|p_c s| < R_s$ (Figure 4.7-a). So s is located within the circle passing through v_1, v_2, and v_3, which is impossible according to the Empty Circle Property of Delaunay Triangulation. So no such p exists and the area is completely covered. ■

4.4 Group-Based *k*-Coverage

For applications in military surveillance and emergency response, *k*-coverage ($k > 1$) is usually required in order to obtain more accurate data and better fault-tolerance. In this section, we propose a generalized technique to extend a *1*-coverage technique to *k*-coverage. Features of our technique includes:

- compatibility with any existing *1*-coverage algorithm;

- capability of maintaining coverage and energy-efficiency;
- easily configurable with k.

The general concept of our group-based technique involves separating all sensors into k mutually exclusive groups. Each group uses *1*-coverage algorithm to optimize its sensing range or choose its sleep/wakeup schedule. Then, by layering the k groups, *k*-coverage can be achieved. For example, if Delaunay-Triangulation-based heuristics are used for *1*-coverage, each node will only keep in its $NeighborList$ the neighbor nodes with the same group I.D. as itself during DT construction; then, all k layers of *1*-coverage can be simultaneously generated by k groups through the DT-based optimization. In order to easily adjust k for various service requirements, a simple, distributed grouping technique is necessary. Furthermore, in order to maintain energy-efficiency and load-balancing, all k groups should have the same number of sensors and the same distribution across the target area.

4.4.1 Probability-Based Approach

The probability-based algorithm is a straightforward approach to form k independent groups. It allows each sensor to select any group I.D. between 1 and k with the same probability of $1/k$. To study the distribution of each group, we assume that sensors are deployed according to the Homogenous Poisson Point Process (HPPP) with density λ (*Definition 1, Section 2.2.1*). Then, we will show that each group of sensors follows the same distribution with density denoted by λ/k and the expectation of the group size equals n/k (*Theorem 10 & 11*).

Lemma 1: If the original point process is an HPPP, for group $l(l \in 1..k)$, $N_l(A_i)$ and $N_l(A_j)$ are independent for any disjoint A_i and A_j.

Due to the HPPP, $N(A_i)$ and $N(A_j)$ are independent for any disjoint A_i and A_j. Because each node randomly joins the group, $N_l(A_i)$ and $N_l(A_j)$ $(l \in 1..k)$ are still independent for any disjoint A_i and A_j; hence, *Lemma 1* holds.

Lemma 2: If the original point process is an HPPP with density λ, then $N_i(A)(A \in \Omega)$ is the Poisson Distribution with density λ/k for group $i(i \in 1..k)$.

Proof: We assume that $1 - p$ is the probability of any point in the original distribution joining group i;

therefore:

$$
\begin{aligned}
P_r(N_i(A) = n) &= \sum_{j=0}^{\infty} (P_r(N(A) = n + j) \binom{n+j}{n} (1-p)^n p^j) \\
&= \sum_{j=0}^{\infty} (\frac{e^{-\lambda|A|}(\lambda|A|)^{n+j}}{(n+j)!} \frac{(n+j)!}{n!j!} (1-p)^n p^j) \\
&= \frac{e^{-\lambda|A|}(\lambda|A|)^n}{n!} (1-p)^n \sum_{j=0}^{\infty} \frac{(\lambda|A|p)^j}{j!} \qquad (4.1) \\
&= \frac{e^{-\lambda|A|}(\lambda|A|)^n}{n!} (1-p)^n e^{\lambda|A|p} \\
&= \frac{e^{-(1-p)\lambda|A|}((1-p)\lambda|A|)^n}{n!} = \frac{e^{-\rho|A|}(\rho|A|)^n}{n!}
\end{aligned}
$$

According to Equation (4.1), $N_i(A)$ is a Poisson Distribution with density ρ where $\rho = (1-p)\lambda$. Because $1/k$ is the probability of each sensor joining group i, $N_i(A)$ is a Poisson Distribution with the density of λ/k for group i. ■

Theorem 10 *If the original point process is an HPPP with density λ and each sensor randomly joins a group $i \in 1..k$ with the probability of $1/k$, then group $i (i \in 1..k)$ is also an HPPP with density λ/k.*

Proof: According to *Definition 1*, a point process is an HPPP if and only if conditions (1) and (2) are satisfied. Therefore, based upon *Lemma 1 & 2*, group $i (i \in 1..k)$ is an HPPP with density λ/k. ■

Theorem 11 *If sensors randomly join group $i (i \in 1..k)$ with a probability of $1/k$, then all groups have the same expectation of group size.*

Proof: $1/k$ is the probability of each sensor joining group i. Suppose ξ is the random variable representing the size of group i, and the number of points in Ω is N. According to Binomial Distribution, $P_B(\xi = x) = \binom{N}{x}(1/k)^x(1-1/k)^{N-x}$, and hence $E(\xi) = N/k$. So, the expectation of the group size is the same for all groups. ■

The probability-based algorithm ensures the same group size and distribution probabilistically by assuming an HPPP for initial sensor deployment; however, it may not be feasible to place sensors in a certain regular manner (*i.e.* HPPP). Furthermore, information on the initial sensor placement may not be available and the topology of sensor networks may change dynamically into any random formation due to various

environmental conditions and unbalanced work-loads. Additionally, in order to obtain better performance in energy-efficiency and load-balancing, more deterministic properties in group size and distribution are preferred.

4.4.2 Grid-Based Approach

Different from the complete random group-formation in the probability-based approach, an alternative way of forming groups is by using full coordination among all sensors. However, such an approach is impractical in sensor networks due to resource constraints and scalability issues. Thus, we propose a grid-based approach, which utilizes limited coordination among one-hop neighbors in order to obtain a more deterministic property under arbitrary network topology.

In the grid-based approach, we apply the randomized group assignment on a much smaller scale than the probability-based approach. The basic idea is to divide the area into contiguous cells where each cell contains k or fewer sensors. Then, we randomly assign group I.D.s (from 1 to k) to sensors within each cell. The group size is N/k for a total of N sensors, if all cells contain exactly k sensors. Furthermore, because WSNs are generally assumed to be densely-deployed, each cell will be small enough to allow the grid-based algorithm to attain the same distribution for each group. For example, as illustrated in Figure 4.9-a, where sensors are more densely-deployed, the cells are much smaller than Figure 4.9-b. Then for any arbitrary area $\mathbb{A}$ (*i.e.* the circle in Figure 4.9-a and b) and two groups G_i & G_j, the relative variation between those two group sizes in $\mathbb{A}$ is $\frac{|G_i|_{\mathbb{A}} - |G_j|_{\mathbb{A}}}{|G_i|_{\mathbb{A}}}$. Because only those cells that intersect $\mathbb{A}$'s border (shaded cells) will contribute to the variation of group sizes, the sizes of all groups in $\mathbb{A}$ are almost equal in Figure 4.9-a. Therefore, as long as the network is densely-deployed, the grid-based approach can obtain desirable properties in group size and distribution under any sensor topologies.

Our grid-based algorithm consists of two phases. The first phase is to construct the grid with each cell containing no more than k sensors. In the second phase, a random method or heuristic is used to assign group I.D.s based on each cell. Obviously, if dividing is continued until each cell contains zero or one sensor, the grid-based approach becomes similar to the probability-based approach.

Grid construction by division has two drawbacks. First, it is slow because the initial cell (the entire target area) has to be divided repeatedly into smaller cells until each cell contains only k or fewer than k sensors. Secondly, it is not feasible to distributively implement division-based construction with local

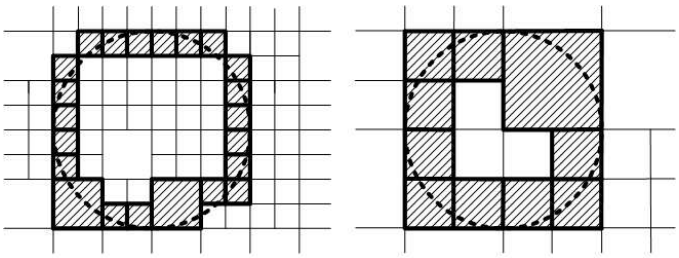


Figure 4.9: Grid Formation with Different Node Densities

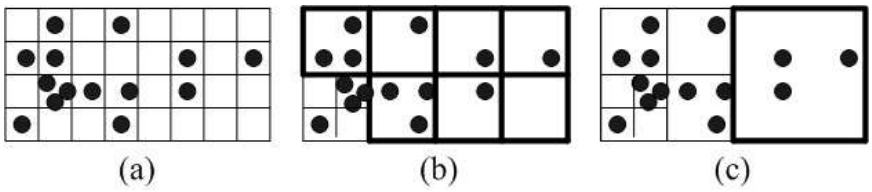


Figure 4.10: Cell-Merging Process ($k = 3$)

topology information. To alleviate those problems, we use a merging process in order to distributively construct the grid in a short time from one-hop information. The grid is initially pre-defined with cells of the same size. Those cells should be small enough that none of them contain more than k sensors (Figure 4.10-a). Each sensor identifies its residual cell based on its location information which is available through GPS or other localization mechanisms (for example, see [41]). If the number of sensors in four quadrants is fewer than or equal to k, then those four quadrants are merged into one rectangular cell (Figure 4.10-b). By continuously merging four smaller quadrants into one larger cell, the number of sensors contained in all cells in the final grid will exactly equal k, or nearly k (Figure 4.10-c). If the network is densely deployed, cells will stop growing before their sizes exceed one-hop range; thus, the merging process can quickly compute the grid with local information.

After the grid is formed, each cell will contain exactly k or fewer than k sensors. In the second phase, each sensor's group I.D. will be determined within its rectangular cell. For a cell S and $|S|$ representing the number of sensors in S,

1. If $|S|=k$, $1..k$ is randomly assigned to k nodes in S;

2. If $|S|<k$, all sensors in S join groups from $1..k$ with a probability that is inversely proportional to the

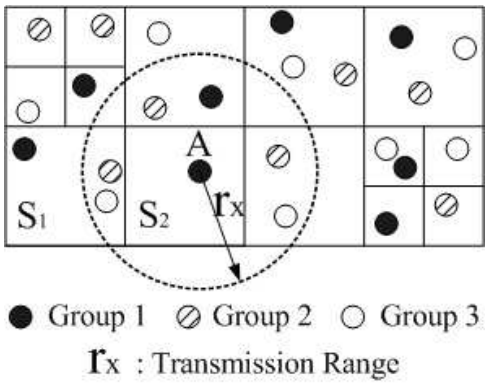


Figure 4.11: Grid-Based Group Assignment for 3-Coverage

size of each group in their one-hop neighborhood.

In Figure 4.11 (*k=3*), cell S_1 has three sensors, thus it randomly assigns each sensor with a group I.D. from $1..3$ according to rule (1). Cell S_2 has one sensor A, hence sensor A uses rule (2) to join group 1 based upon group sizes in its neighborhood. Within sensor A's one-hop range, there is one sensor in group 1, three sensors in group 2, and two sensors in group 3; so sensor A will have the highest probability of joining group 1.

4.5 Performance Evaluation

4.5.1 1-*Coverage with Uniform Sensing Radii*

We implemented the gossip-based topology control techniques, *i. e.* dynamic-square and dynamic-triangle, based on a square mesh and an equilateral-triangular mesh. To study their performance, we also implemented the static triangular mesh method discussed in [70] and a similar static square mesh method. The two static methods, static-square and static-triangle, choose active sensors based on a static mesh and thus, are not capable of shifting correction and energy-balancing. We simulate all techniques in a $100m \times 100m$ square with a size of the network ranging from 300 to 1000 sensors. All sensors are randomly deployed and have a $25m$ transmission range and $10m$ sensing range. As a result, each cell in the square mesh has its sides as $10\sqrt{2}m$ and each cell in the equilateral-triangular mesh has its sides as $10m$. Since we only focus on coverage, we use percentage of coverage ($\%$) and number of active nodes as the metrics to study the performance of our techniques. All data points are obtained by averaging the results from 20 randomly

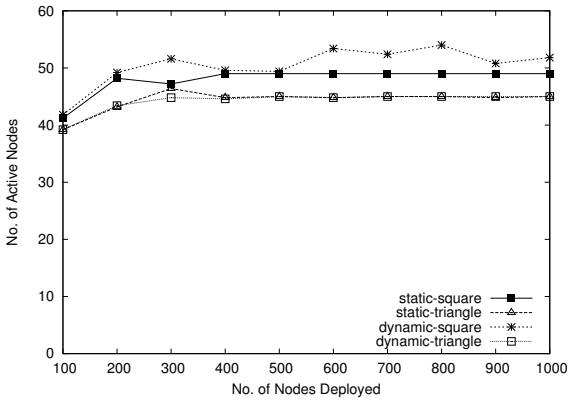


Figure 4.12: Energy Consumption

generated topologies.

Gossip-Based Mesh for Energy Efficient Coverage

We studied the performance of our gossip-based mesh construction technique by varying the size of the network. In Figure 4.12, the dynamic techniques have almost the same number of active sensors as the static techniques. Furthermore, all four techniques have a nearly consistent mesh size when the network size is larger than 300. This is because there are a fixed number of static cells and only one sensor will be initiated for each cell. Fewer active sensors in a lower density (< 300) network is observed because some cells may not have any sensors in it. Also, square mesh techniques have more active sensors than the equilateral-triangular mesh techniques because the size of cell is smaller than the square mesh. In Figure 4.13, the dynamic-square and dynamic-triangle techniques have an improved percentage of coverage over static-square and static-triangle techniques. As expected, the advantage of using virtual mesh and dynamic shifting correction becomes more obvious with lower network densities. This is because the lower network densities make it harder to find sensors at preferred locations and thus, an erroneous shifting will occur more likely. Without an appropriate adjustment of those preferred locations based on the local topology, the performance of the two static approaches degrades significantly in their coverage rate. Therefore, with an equal number of active sensors, a better coverage can be achieved by using dynamic mesh construction techniques.

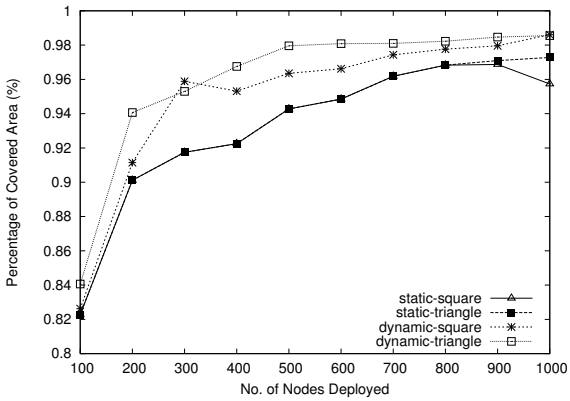


Figure 4.13: Quality of Coverage

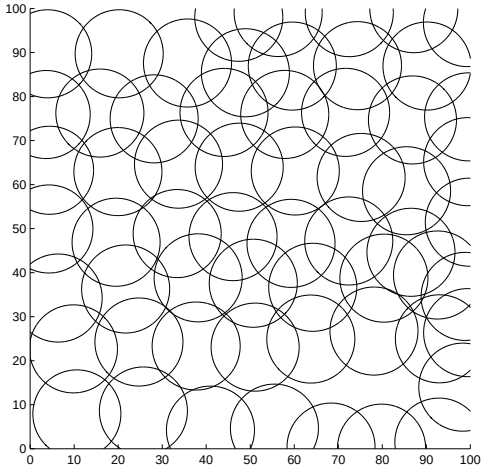


Figure 4.14: Square Mesh without Coverage Improvement

Dynamic Mesh with Coverage Improvement

Figure 4.14 and 4.16 are generated with a network with size of 500 nodes. Both figures show obvious uncovered area. Figure 4.15 and 4.17 are generated based on the same network topology but applied with our coverage improvement technique. Figure 4.15 and 4.17 visually demonstrate that the uncovered holes are precisely detected and additional sensors are initiated to improve the coverage.

Figure 4.18 shows the additional energy consumption after applying the coverage improvement technique to our dynamic mesh. After the coverage improvement process is finished, the average size of the square mesh and the equilateral-triangle mesh increases from 47 and 54 to 62. However, Figure 4.19 demonstrates that the dynamic meshes with coverage improvement have a significant improvement of coverage to

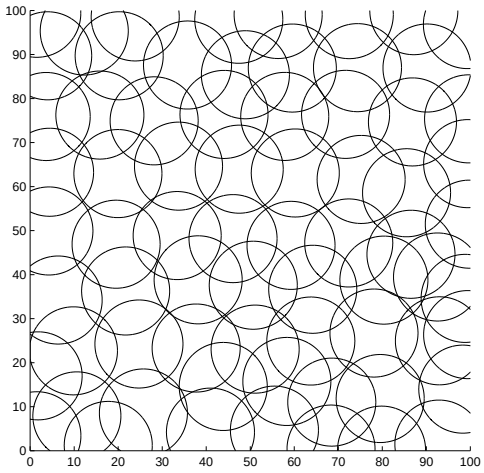


Figure 4.15: Square Mesh with Coverage Improvement

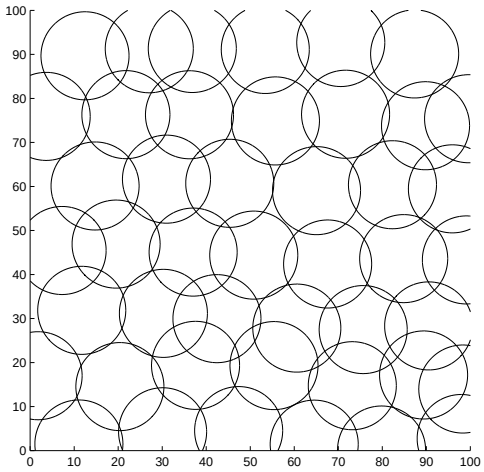


Figure 4.16: Equilateral-Triangular Mesh without Coverage Improvement

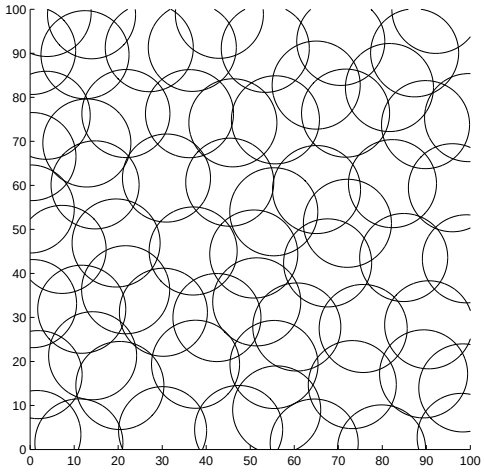


Figure 4.17: Equilateral-Triangular Mesh with Coverage Improvement

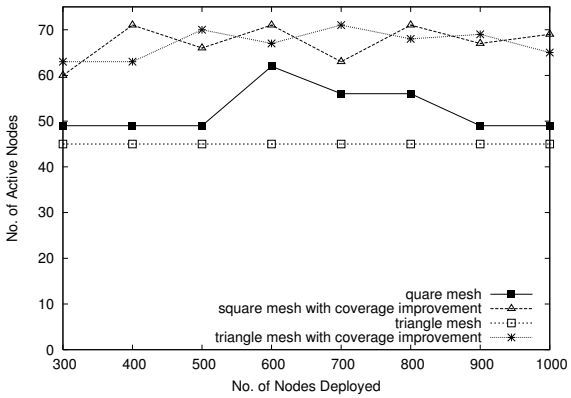


Figure 4.18: Energy Consumption with Coverage Improvement

almost 100% under all network densities. In lower density, the coverage improvement is more obvious (about 10%).

Our coverage improvement technique identifies an area of uncovered holes and tries to activate a sensor at the centroid of this area. However, it is possible that there is no sensor that is close to the selected locations. To preserve energy, only sensors within a certain range of the selected location can be activated to cover the hole. We study the accuracy of selecting new active sensors in Figure 4.20 and 4.21. We use a network size of 500 sensors. In Figure 4.20, as the accuracy-range increases, more sensors are selected to cover the holes and as a result, improved coverage is shown in Figure 4.21. Furthermore, Figure 4.21 demonstrates that a better accuracy-range is between $4m$ to $6m$. The accuracy-range that is less than $4m$

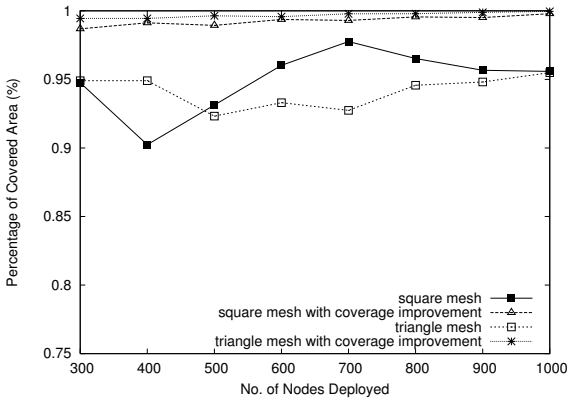


Figure 4.19: Quality of Coverage with Coverage Improvement

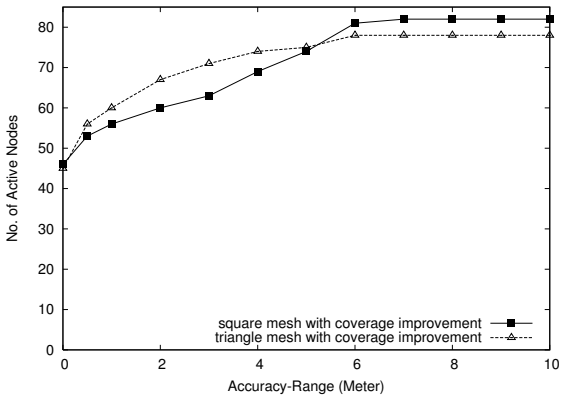


Figure 4.20: Energy Consumption with Varying Accuracy Range

does not allow sufficient numbers of additional sensors for hole recovery, while the accuracy-range that is larger than $6m$ does not provide more benefit in coverage. However, the optimal accuracy-range depends on the network density. For lower density, a larger accuracy-range is required in order to obtain full coverage.

4.5.2 1-*Coverage with Variable Sensing Radii*

We implemented our two Delaunay-Triangulation-based sensing coverage heuristics using the ns-2 simulator. To evaluate their performance, we also implemented Wu's Variable Sensing Range algorithm in ns-2 (We refer it as VSR in our paper), because its ns-2 results are not available in the original paper [70]. The metrics we used for comparison include average sensing radius ($Meter$), average sensing power consumption ($Watt$) and coverage ratio (%). Additionally, for energy-balancing, we also evaluated our two Delaunay-Triangulation-based heuristics in terms of average energy level ($Joule$) and number of failed

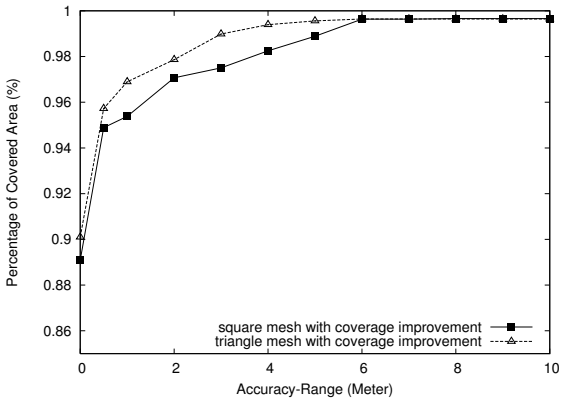


Figure 4.21: Quality of Coverage with Varying Accuracy Range

sensor nodes.

Network Size

The average sensing radius, average sensing power consumption, and coverage ratio are computed to compare the performances of three variable sensing radii assignment techniques. The average sensing radius and sensing power consumption are calculated by averaging the current sensing radii and sensing power consumption among all sensor nodes for every five seconds; the coverage ratio is calculated by dividing the given area into a 1000×1000 grid and for each cell, it is considered to be covered if its central point is within some sensor's sensing range. Additionally, we used the sensing energy model as $E = k \times r^2$ and k is 0.001 for all simulation runs. The network topology is a 50×50 grid and is randomly deployed with a varying number of sensor nodes from 50 to 250.

In Figure 4.22, our two Delaunay-Triangulation-based heuristics achieve almost the same performance. As the number of nodes increase, their average sensing radii decreases from 9.8m to 4.7m due to nodes getting closer which causes sharp triangles to occur less. For VSR, the sensing range tends to be constant between 6m to 7m as the density of sensor nodes increases. The reason is because VSR uses static strategy based on pre-chosen locations for sensing radii assignment, therefore its average sensing range is not affected by node density. Furthermore, although VSR shows better performance with low density (< 0.06 $sensor/m^2$), as node density increases, Delaunay-Triangulation-based heuristics tend to achieve better performance due to its dynamic radii assignments.

In Figure 4.23, because sensing power consumption is directly related to a sensor's sensing range, a

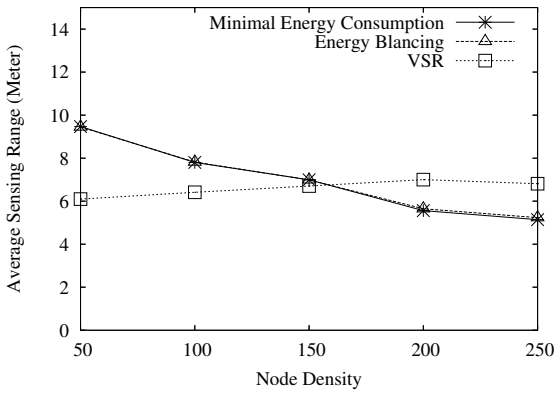


Figure 4.22: Average Sensing Radius

similar trend can be observed as in Figure 4.22. Specifically, our Delaunay-Triangulation-based heuristics drop average power consumption from 0.0009w to 0.0003w, while VSR keeps a consistent average power consumption between 0.0004w to 0.0005w as node density increases from $0.02\ sensor/m^2$ to 0.1 $sensor/m^2$.

Since wireless sensor networks are generally used for emergency and life-critical systems, complete coverage for reliability is foremost. Although VSR shows better average radius assignments with a low level of node density ($< 0.06\ sensor/m^2$) in Figure 4.22, it can only achieve 85% to 95% coverage (Figure 4.24). The reason for the low coverage ratio of VSR with low density is because with less available sensors in the given area, finding sensors that are close enough to pre-chosen locations becomes impossible. However, Delaunay-Triangulation-based heuristics show consistent performance with a coverage ratio larger than 95% for all densities (Figure 4.24), because our technique uses dynamic radius assignment and can better adapt to different levels of node density.

Energy Balancing and Lifetime

We proposed two Delaunay Triangulation based heuristics for minimal energy consumption and energy balancing. To evaluate the performances of these two heuristics, we compared their average energy level (J) and number of failed nodes, where average energy level is calculated by averaging the current energy storage of all sensors. Additionally, our sensing model is chosen as $E = k \times r^2$ with k as 0.01. The initial energy level was randomly chosen between $40J$ to $50J$. The simulations ran on a 50×50 grid with two different levels of node densities: 100 nodes and 200 nodes.

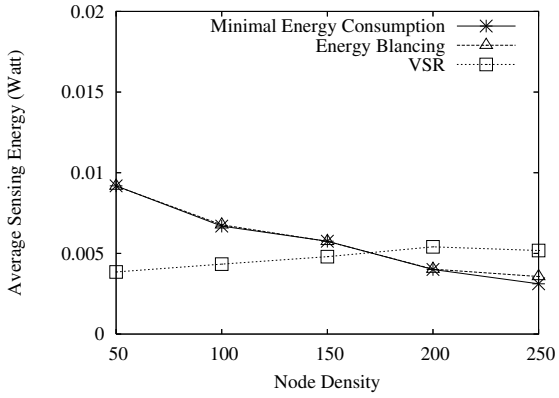


Figure 4.23: Average Sensing Energy Consumption

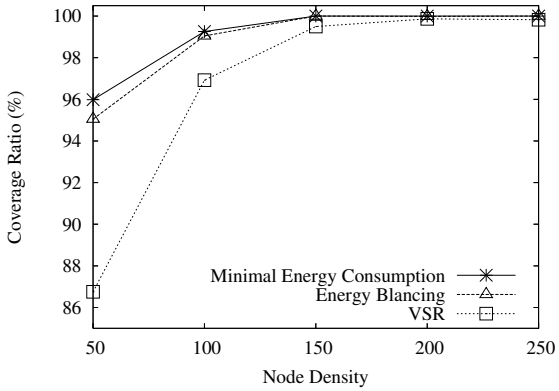


Figure 4.24: Coverage Ratio

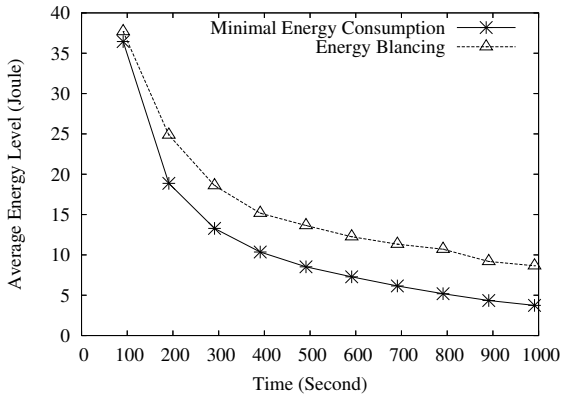


Figure 4.25: Average Energy Level (100 nodes)

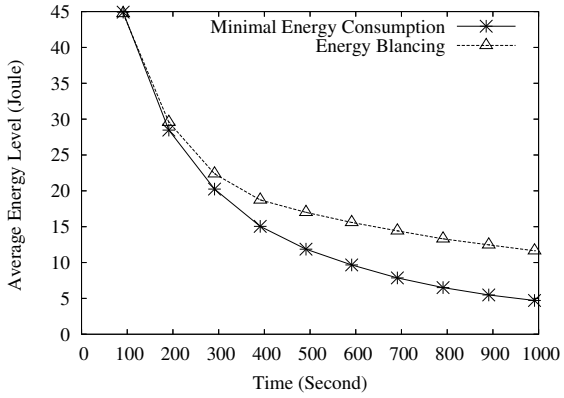


Figure 4.26: Average Energy Level (200 nodes)

Figure 4.25 displays the average energy level of sensor nodes as time progresses. From 100s to 1000s, the trend of decreasing energy can be observed for both heuristics as expected. However, the heuristic for energy balancing performs better with an average of 30% improvement, because this technique minimizes the difference of energy level among sensor nodes. In Figure 4.26, with higher node density, higher energy levels can be observed due to lower sensing radius assignments for sensor nodes. Additionally, a similar decreasing trend can be observed as the heuristic for energy-balancing outperforms the heuristic for minimal energy consumption.

Figure 4.27 and Figure 4.28 show the number of failed nodes as time progresses. For low node density, the nodes fail more quickly due to a larger sensing range. In Figure 4.27, it can be observed that the energy balancing heuristic outperforms the other heuristic by a 15% reduction of failed nodes on average. In Figure 4.28, although the heuristic for minimal energy consumption shows better performance before 400s, as time goes on, the heuristic for energy-balancing shows obvious improvements in slowing down the early node failures and extending the lifetime of sensor networks.

4.5.3 k-Coverage Algorithms

Probability VS. Grid-Based Approaches

Group-based *k*-coverage technique is applied to DT-based *1-coverage* in order to evaluate the probability and grid-based approaches. Our evaluation metrics included the coverage ratios and the averages of the sensing radii and sensing energy. In Figure 4.29, a correlation is observed between the levels of coverage and sensing radii, where a higher level of coverage requires larger sensing radii. This is because node density decreases

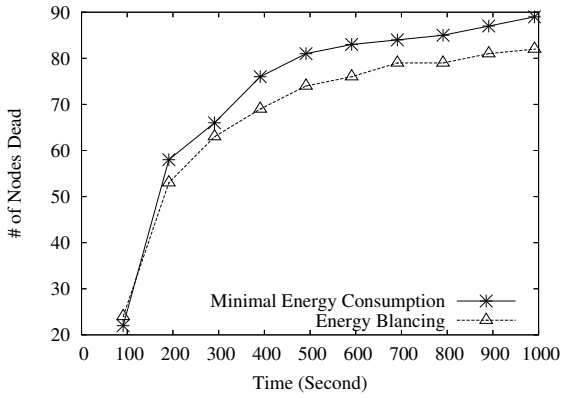


Figure 4.27: Number of Failed Nodes (100 nodes)

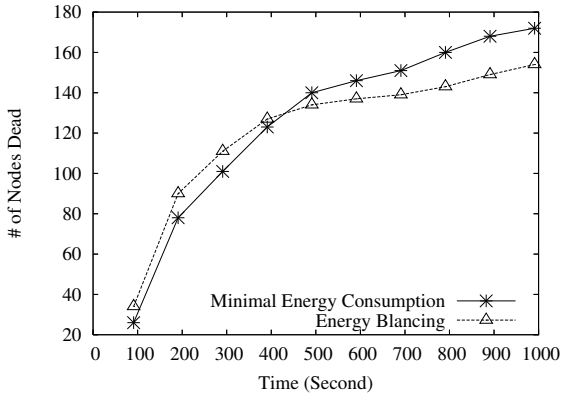


Figure 4.28: Number of Failed Nodes (200 nodes)

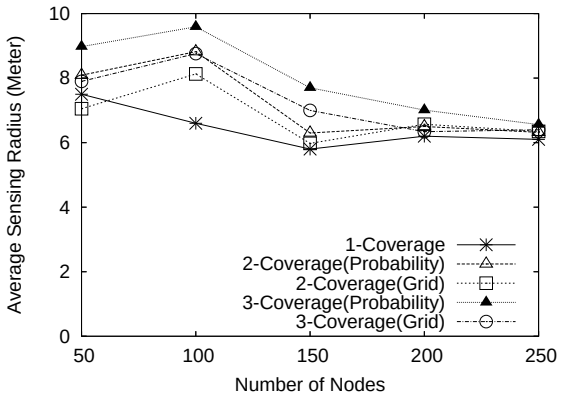


Figure 4.29: *k*-Coverage: Probability *VS.* Grid-Based Approaches (Average Sensing Radius)

in each group with a higher level of coverage. Additionally, in the process of reaching 2-coverage, the grid-based algorithm resulted in a better average radius than the probability-based algorithm. The grid-based algorithm obtained an average sensing radii that is about 6% lower than the probability-based algorithm. Similarly, in obtaining 3-coverage, the grid-based algorithm outperforms the probability-based algorithm and its average sensing radius is about 11% lower. The explanation lies mainly in the fact that grid-based approach uses a local collaboration mechanism to attain more balanced group formation and hence, improves the radii assignment in randomly deployed networks. Figure 4.30 illustrates the average sensing energy consumption of 2 and 3-coverage and shows a behavior similar to that which is described above. Therefore, with sufficient sensors deployed, the grid-based approach offers improved optimization of sensing radii based on local topology. Furthermore, as shown in Figure 4.31, surveillance quality was evaluated by comparing the coverage ratio, which shows that both probability and grid-based mechanisms can achieve almost the same coverage ratio.

Extensibility

Our grouping technique extends any 1-coverage algorithms to multiple coverage and preserves their original energy efficiency and reliability. We applied the group technique (Grid) to both VSR and DT-based algorithm. As in section 4.5.2, we have shown DT-based 1-coverage can adapt to the lower density with better reliability and higher density with more energy-efficiency. After applying the grouping technique to *1*-coverage algorithm, the energy consumption and the quality of coverage should follow the similar trend as in Fig. 4.22-4.24. In Fig. 4.32, Grouping technique using DT-based 1-coverage has its average radius

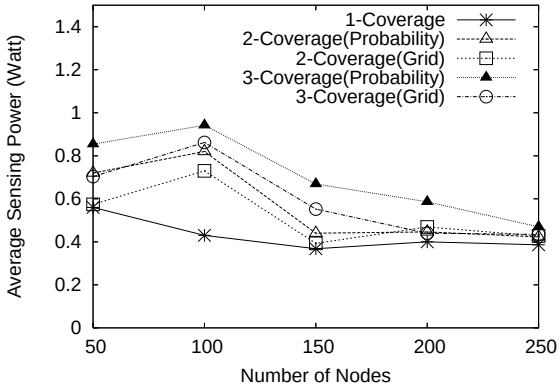


Figure 4.30: *k*-Coverage: Probability *VS.* Grid-Based Approaches (Average Sensing Power)

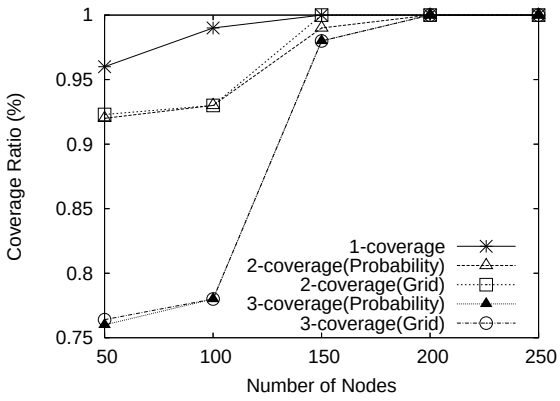


Figure 4.31: *k*-Coverage: Probability *VS.* Grid-Based Approaches (Coverage Ratio)

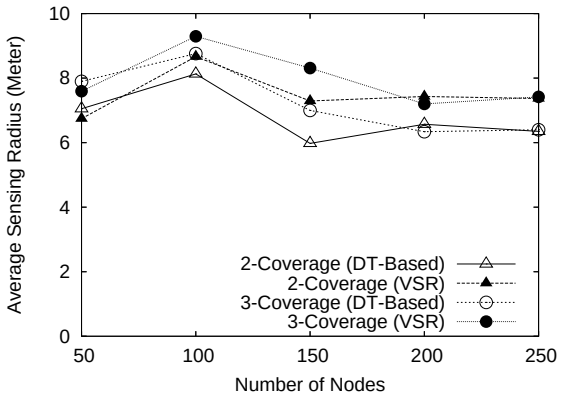


Figure 4.32: *k*-Coverage: Extensibility (Average Sensing Radius)

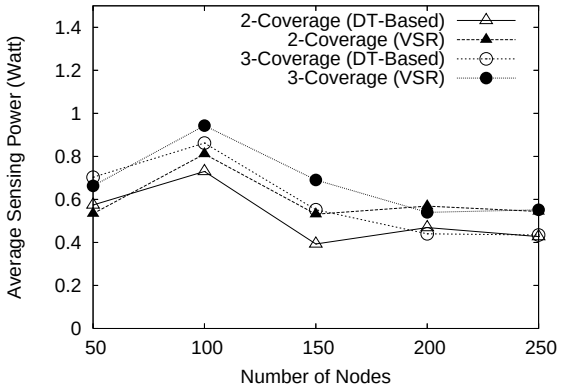


Figure 4.33: *k*-Coverage: Extensibility (Average Sensing Energy)

about 3-5% lower than VSR, which is about the same difference that demonstrated in Fig. 4.23. Similar performance is also shown in Fig. 4.33 where the average energy level is calculated based on each sensor's radius. In addition, Fig. 4.34 shows a significant coverage-ratio degradation of VSR when the node density is low; however, as higher nodes density increases, both techniques reaches almost 100 % coverage. Comparing to the results from 4.5.2, grouping technique preserves both energy efficiency and coverage ratio for these original 1-coverage algorithms.

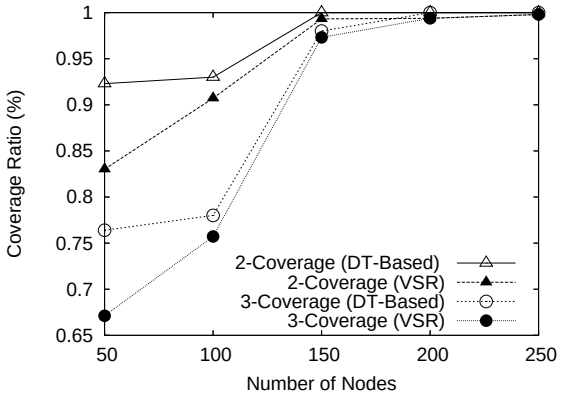


Figure 4.34: *k*-Coverage: Extensibility (Coverage Ratio)

CHAPTER FIVE

CONCLUSIONS

This thesis is comprised of three research topics in wireless ad hoc and sensor networks. First of all, a fault-resilient routing protocol is proposed for MANET to improve routing stability, throughput, and robustness in highly mobile and error-prone environments. Secondly, a monitor-based transport protocol is developed for upstream data delivery in WSN to improve packet-level reliability. Finally, several techniques for energy efficient coverage were designed to dynamically control the topology for guaranteed QoS and energy efficiency.

5.1 FaRM Routing Protocol

Fault-Resilient MANET (FaRM) routing protocol is based on source-initiated routing. It selects routes with better quality and allows for quick local-recovery when a route is partially broken. We believe that, due to its fault-resiliency, FaRM is best-suited for disaster recovery, emergency response, and battlefield scenarios. The following are two key characteristics of our protocol:

- The density-first routing selection technique not only chooses more stable routes during route discovery, but also helps to improve the successful local recovery rate since more alternative routes will be available near nodes with high densities.
- The local route salvaging process for route maintenance is designed to have a minimal effect on other traffic and, reduce the delay and overhead caused by dynamic topology changes.

We compared the performance of FaRM with NSR, WAR, and DSR based on the *ns-2* simulator. Our simulation experiments confirmed that using node density as a metric during route discovery can efficiently improve the routing stability. Additionally, FaRM exhibited an improved throughput and reduced routing overhead. Finally, in our research, we concluded that the optimal recovery zone angle is between $\pi/2$ and $2\pi/3$, provided that a medium moving speed (10m/s - 15 m/s) is used.

5.2 Monitor-Based Transport Protocol

The monitor-based transport protocol for reliable upstream data delivery is designed for in-network processing techniques, such as data-aggregation to provide packet-level reliability. By utilizing inactive non-forwarding nodes as monitors, our protocol can relieve forwarders of the need to cache packets for recovery, alleviate congestion by diverting the traffic, and assist in quick loss detection and recovery in the event of heavy traffic and unreliable nodes. Key features of our protocol include:

- A monitor- and sender-combined mechanism for reliable and fast loss detection and;
- A monitor-aided approach to packet loss recovery in order to handle scenarios with congestion or unreliable sensor nodes.

In order to ensure energy-efficiency and reliable data transport, it is necessary to identify a minimum set of monitors that cover all current flows; however, this problem is NP-complete. Therefore, we propose a distributed heuristic algorithm to address the problem efficiently. When compared to other upstream transport protocols, our *ns-2*-based simulations confirm that monitor-based transport protocol obtains a higher data-delivery rate and improved throughput in the presence of dynamic topology changes and intermittent heavy traffic.

5.3 Energy Efficient Coverage

Three different techniques were developed for energy efficient coverage for different applications and environments. For *1*-coverage with uniform sensing radius, we proposed a mesh construction and coverage improvement technique. The primary contribution of our technique is to provide energy-balancing and energy-efficiency of sensor networks with guaranteed full coverage. To provide energy-balancing, the mesh is reconstructed periodically by randomly picking up an initiator from the network. To avoid redundant coverage, the virtual mesh and shifting correction technique is applied to the activation chain between neighboring sensors during the mesh construction. As a result, a set of active sensors are selected to best approximate the square mesh or equilateral-triangular mesh regardless of the location of the initiator and local shifting errors. To guarantee full coverage, a hole detection and recovery process is applied locally to neighboring cells.

For *1*-coverage with variable sensing radii, the Delaunay Triangulation-based technique was proposed. This technique includes the use of a light-weight, distributed algorithm to approximate Delaunay Triangulation as a basis for estimating the optimum radii to obtain *1*-coverage and energy-efficiency. Based on the assumption that $R_x > 2R_s$, our one-hop approximation of DT yields accurate Delaunay Triangulation. Furthermore, we prove that both of the DT-based heuristics can reach the global optimality in energy-efficiency as well as complete coverage. These two DT-based heuristics are implemented using ns-2 simulator and compared with VSR [70]. The experiment results confirm improved coverage, reduced sensing energy consumption and longer lifetime of systems with various levels of node density.

To satisfy various QoS requirements for accuracy and fault-tolerance, a group-based *k*-coverage technique is proposed to extend any *1*-coverage algorithm into *k*-coverage. For the two method used in group division, *i. e.* probability-based and grid-based, the grid-based approach is a better-suited algorithm for *k*-coverage because it involves limited coordination among one-hop neighbors which allows it to attain more desirable features with arbitrary sensor-deployment. Finally, our *ns-2*-based experimentation shows that our algorithms can reduce sensing radii and energy consumption while preserving overall coverage with various node densities.

5.4 Future Work

The future research will consider more scenarios for our mobility model. Our results for an average route-lifetime is based on the worst scenarios. An accurate mathematic model for mobility should also consider a node's moving direction. Furthermore, our routing decision can be further improved in two possible aspects. First of all, in our Matlab results (Fig. 2.1), higher density do not significantly improve route-lifetime if the density level is higher than 0.5 $node/m^2$. A more accurate mobility model will assist us to identify this density threshold. Once the node density reach the threshold, the routing decision can be solely based on each node's movement. Secondly, the moving speed is considered as a scalar in our routing decision, therefore, a route with nodes of higher moving speed will always be less preferable. However, when the signal between two nodes is strengthened, those nodes are moving closer. By considering the trend of signal gain and attenuation, a better routing decision can be obtained.

Another future research area is the data transport and routing protocols in WSN. In our data transport

protocol, we make the same effort to provide a reliable end-to-end delivery of all packets. This is based on the assumption that packets are aggregated and thus, a reliable delivery is required. However, based on different aggregation level, each packet could carry different priorities. A packet that aggregated from 1K packets should have higher priority than the one aggregated from 10 packets. By differentiate priorities at the data transport level, a packet recovery process can be more efficiently tailed to meet the constrains in the network. Furthermore, to provide the most efficient data aggregation, the routing structure must be self-adaptive to the changes in traffic patterns and network topologies. Thus, an appropriate routing structure is required in order to aggregate data with enhanced reliability and energy efficiency. In addition, since most applications of WSNs have real-time constraints, the data centric routing has to consider the tradeoff between data aggregation and end-to-end delay. As the data aggregation technique provides the only scalable way to handle the intermittent traffic and many-to-one communication pattern of WSNs, research in those areas will become the cornerstone for future designs of upper layer protocols and applications.

BIBLIOGRAPHY

[1] A.Khan, C.Qian, P.Sharma, and S.K.Tripathi. An energy-efficient mobile triangulation-based coverage scheme. In *ICC 2007*.

[2] A.Nasipuri and S.R.Das. On demand multipath routing for mobile ad hoc networks. In *IEEE ICCCN*, 1999.

[3] A.Rao, C.Papadimitriou, S.Shenker, and I.Stoica. Geographic routing without location information. In *MobiCom*, 2003.

[4] G. Asada, T. Dong, F. Lin, G. Pottie, W. Kaiser, and H. Marcy. Wireless integrated network sensors: Low power systems on a chip, 1998.

[5] A.Valera, W.Seah, and S.V.Rao. Champ: A highly-resilient and energy-effcient routing protocol for mobile ad hoc networks. In *Proc. of 4th IEEE MWCN*, 2002.

[6] B. Awerbuch, D. Holmer, C. N.-R., and H. Rubens. An on-demand secure routing protocol resilient to byzantine failures. In *WiSe*, 2002.

[7] Chen B, Jamieson K, Balakrishnan H, and Morris R. Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *In 7th Annual Int. Conf. Mobile Computing and Networking*, 2001.

[8] C.-Y.Wan, S.B.Eisenman, and A.T.Campbell. CODA: congestion detection and avoidance in sensor networks. In *SenSys'03*, 2003.

[9] B. Carbunar, A. Grama, and J. Vitek. Distributed and Dynamic Voronoi Overlay Maintenance for Coverage Detection and Distributed Hash Tables in Ad Hoc Networks. In *IEEE ICPADS*, Jun 2004.

[10] M. Cardei, J. Wu, M. Liu, and M. Pervaiz. Maximum network lifetime in wireless sensor networks with adjustable sensing ranges. In *IEEE WiMob*, 2005.

[11] A. Cerpa and D. Estrin. ASCENT: Adaptive Self-Configuring sEnsor Network Topologies. In *INFOCOM*, 2002.

[12] C.Intanagonwiwat, R.Govindan, and D.Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Mobile Computing and Networking*, 2000.

[13] C.L.Lawson. Software for c^1 surface interpolation. In *Mathematical Software*, 1977.

[14] Diane Cook and Sajal Das. *Smart Environments: Technology, Protocols and Applications*. New York: John Wiley, 2004.

[15] C.Perkins. Ad-hoc on-demand distance vector routing. In *MILCOM'97*, Nov 1997.

[16] C.Perkins and P.Bhagwat. Highly dynamic destination sequenced distance vector routing (DSDV) for mobile computers. In *ACM SIGCOMM'94*, pages 234–244, 1994.

[17] C.Wang, K.Sohraby, B.Li, M.Daneshmand, and Y.Hu. A survey of transport protocols for wireless sensor networks. *IEEE Network*, 2006.

[18] C.Y.Wan, A.T.Campbell, and L. Krishnamurthy. PSFQ: A reliable transport protocol for wireless. sensor networks. In *WSNA'02*, 2002.

[19] D.B.Johnson and D.A.Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In *Mobile Computing*. 1996.

[20] A. Dhawan, C. T. Vu, A. Zelikovsky, Y. Li, and S. K. Prasad. Maximum Lifetime of Sensor Networks with Adjustable Sensing Range. In *IEEE SNPD'06*, 2006.

[21] Cheng Tien Ee and Ruzena Bajcsy. Congestion control and fairness for many-to-one routing in sensor networks. In *SenSys*, 2004.

[22] E.Royer and C.Toh. A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks. In *IEEE Personal Commun.*, 1999.

[23] F.Stann and J.Heidemann. RMST: Reliable data transport in sensor networks. In *SNPA'03*, 2003.

[24] A. Gallais, F. Ingelrest, J. Carle, and D. Simplot-Ryl. Preserving Area Coverage in Sensor Networks with a Realistic Physical Layer. In *INFOCOM*, 2007.

[25] M. Hefeeda and M. Bagheri. Randomized k-coverage algorithms for dense sensor networks. In *INFOCOM*, 2007.

[26] H.Luo, Y.Liu, and S.K.Das. Routing corelated data with fusion cost in wireless sensor networks. *IEEE Mobile Computing*, 5, 2006.

[27] H.M.Taylor and S.Karlin. *Random point processes in time and space (2nd ed.)*. New York: Springer-Verlag, 1994.

[28] C-F. Hsin and M. Liu. Network coverage using low duty-cycled sensors: random and coordinated sleep algorithms. In *IPSN '04*, 2004.

[29] Y.-C. Hu, D.B. Johnson, and A. Perrig. Sead: Secure efficient distance vector routing for mobile wireless ad hoc networks. In *IEEE Wksp. Mobile Com. Sys. and Apps*, 2002.

[30] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *MobiCom*, 2002.

[31] C. Huang and Y. Tseng. The coverage Problem in a Wireless Sensor Network. In *WSNA*, Sep 2003.

[32] H.Zhang, A.Arora, Y.Choi, and M.Gouda. Reliable bursty convergecast in wireless sensor networks. In *MobiHoc'05*, 2005.

[33] H.Zhang and J.C.Hou. Maintaining scheme coverage and connectivity in large sensor networks. In *Technical Report, UIUC*, 2003.

[34] H.Zhang and J.C.Hou. Maintaining sensing coverage and connectivity in large sensor networks. In *NSF Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, 2004.

[35] I.Aron and S.Gupta. A witness-aided routing protocol for mobile ad-hoc networks with unidirectional links. In *Proc. of First Intl Conf on Mobile Data Access (MDA99)*, Dec. 1999.

[36] I.Aron and S.Gupta. A witness-aided routing protocol for mobile ad-hoc networks with unidirectional links. In *MDA'99*, 1999.

[37] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: Mobile networking for ”smart dust”. In *International Conference on Mobile Computing and Networking (MOBICOM)*, pages 271–278, 1999.

[38] K.Paul, S.Bandyopadhyay, A.Mukherjee, and D.Saha. Communication aware mobile hosts in ad-hoc network. In *Proc. of IEEE ICPWC*, 1999.

[39] X.-Y. Li, G. Calinescu, P.-J. Wan, and Y. Wang. Localized Delaunay Triangulation with Application in Ad Hoc Wireless Networks. *IEEE TPDS* , 14:1035–1047, Oct 2003.

[40] J. Liebeherr, M. Nahas, and W. Si. Application-layer multicasting with Delaunay triangulation overlays. *IEEE JSAC* , 20:1472–1488, Oct 2004.

[41] M. Medidi, R. A. Slaaen, Y. Zhou, C. Mallery, and S. Medidi. Scalable localization in wireless sensor networks. In *HiPC 2006*.

[42] Sirisha Medidi and Jiong Wang. A fault resilient routing protocol for mobile ad-hoc networks. In *In Proceedings of 3rd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2007.

[43] J.Li M.Jiang and Y.-C.Tay. Cluster based routing protocol (CBRP) functional specification (Internet-Draft). Aug 1998.

[44] M.R.Garey and D.S.Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H.Freeman and Company, 1979.

[45] M.Spohn and J.J.Garcia-Luna-Aceves. Neighborhood aware source routing. In *Proc. of ACM MobiHOC'01*, 2001.

[46] U.S. Department of Defense and Ivan Getting. Global Positioning System (GPS).

[47] R.Dube, C.Rais, K.Wang, and S.Tripathi. Signal stability based adaptive routing (SSA) for ad hoc mobile networks. In *IEEE Personal Commun.*, Feb 1997.

[48] R.Jain and K.K.Ramakrishnan. Congestion avoidance in computer networks with a connectionless network layer: concepts, goals, and methodology. In *IEEE computer networking symposium*, 1988.

[49] R.K.Ganti and M.Haenggi. Regularity in sensor networks. In *International Zurich Seminar on Communications*, 2006.

[50] S.-J.Lee and M.Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. In *IEEE ICC*, 2001.

[51] S.Agarwal, A.Ahuja, J.P.Singh, and R.Shorey. Route-Lifetime Assessment Based Routing (RABR) Protocol for Mobile Ad-Hoc Networks. In *ICC (3)*, pages 1697–1701, 2000.

[52] Y. Sankarasubramaniam, O. Akan, and I. Akyildiz. ESRT: Event-to-sink reliable transport in wireless sensor networks. In *MobiHoc'03*, 2003.

[53] S.Floyd and V.Jacobson. Random early detection gateways for congestion avoidance. *In ACM/IEEE Trans. on Networking*, 3(1), 1993.

[54] S. Shakkottai, R. Srikant, and N.B. Shroff. Unreliable sensor grids: coverage, connectivity and diameter. In *INFOCOM*, Mar 2003.

[55] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In *ICC 2001*, May 2001.

[56] S.Meguerdichian, F.Koushanfar, M.Potkonjak, and M.B.Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *INFOCOM*, 2001.

[57] S.Park, R.Vedantham, R.Sivakumar, and I.Akyildiz. A scalable approach for reliable downstream data delivery in wireless sensor networks. In *MobiHoc'04*, 2004.

[58] Andrew S. Tanenbaum. *Computer Networks*. Prentice Hall PTR, Upper Saddle River, New Jersey, 1996.

[59] D. Tian and N. Georganas. A coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks. In *WSNA*, Sep 2002.

[60] C.-K. Toh. Associativity-based routing for ad hoc mobile networks. *Wirel. Pers. Commun.*, 4(2):103–139, 1997.

[61] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *Wireless Networks*, 2002.

[62] V.Park and S.Corson. Temporally ordered routing algorithm (TORA), functional specification. IETF Internet Draft, 1998.

[63] Jean Walrand and Pravin Varaiya. *High-Performance Communication Networks*. Morgan Kaufmann Publishers, San Francisco, California, 2000.

[64] P.-J. Wan, C.-W. Yi, F. Yao, and X. Jia. Asymptotic critical transmission radius for greedy forward routing in wireless ad hoc networks. In *MobiHoc*, 2006.

[65] J. Wang and S. Medidi. Energy efficient coverage with variable sensing radii in wireless sensor networks. In *IEEE WiMob*, 2007.

[66] Jiong Wang and Sirisha Medidi. Density-first ad-hoc routing protocol for manet. In *In Proceedings of 3rd IEEE International Conference on Communication System Software and Middleware (COMSWARE)*, 2008.

[67] L. Wang and Y. Xiao. A Survey of Energy-Efficient Scheduling Mechanisms in Sensor Networks. *MONET*, 11:723–740, Otc 2006.

[68] X. Wang, G. Xing, Y. Zhang, R. Pless C. Lu, and C. Gill. Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks. In *Sensys*, Nov 2003.

[69] W.C.Chung and D.S.Ha. An accurate ultra wideband ranging for precision asset location. In *Int. Conf. UWB Systems&Technologies*, 2003.

[70] J. Wu and S. Yang. Coverage issue in sensor networks with adjustable ranges. In *IEEE ICPPW'04*, 2004.

[71] F. Ye, G. Zhong, S. Lu, and L. Zhang. PEAS, A robust Energy conserving Protocol for Long-lived Sensor Networks. In *ICNP,* , 2002.

[72] Y.Ko and N.H.Vaidya. Geocasting in mobile ad hoc networks:Location-based multicast algorithms. In *In Proc. IEEE 2nd WMCSA*, 1999.

[73] H. Gupta Z. Zhou, S. Das. Connected k-coverage problem in sensor networks. In *ICCCN*, 2004.

[74] B.Y. Zhao, J.D. Kubiatowicz, and A.D. Joseph. Tapestry: An Infrastructure of Fault-resilient Wide-area Location and Routing. Technical Report UCB CSD-01-1141, 2001.

[75] Z. Zhou, S. Das, and H. Gupta. Variable Radii Connected Sensor Cover in Sensor Networks. In *SECON*, Oct 2004.

[76] Z.Ye, A.A.Abouzeid, and J. Ai. Optimal policies for distributed data aggregation in wireless sensor networks. In *INFOCOM*, 2007.