Energy-Aware Topology Control and Data Delivery in Wireless Sensor Networks

A Thesis
Presented to
The Academic Faculty

by

Seung-Jong Park

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
July 2004

Energy-Aware Topology Control and Data Delivery in Wireless Sensor Networks

Approved by:

Raghupathy Sivakumar, Advisor

Jennifer E. Michaels

John A. Copeland

Samit Soni

Henry L. Owen

Date Approved: 7 July 2004

To my companion for life: Young-Jin Juhn.

ACKNOWLEDGEMENTS

I first would like to express my sincere gratitude to my thesis advisor, Dr. Raghupathy Sivakumar. He has always advised me on not only the direction of research but also the way of campus life. I will miss the days and nights having brainstorming which enables me to finish my thesis. I also would like to thank to the members of dissertation committee, Dr. John Copeland, Dr. Henry Owen, Dr. Jennifer Michaels, and Dr. Samit Soni, who have also given many constructive comments during the proposal and final dissertation process. For the helpful discussion about simulation and the thesis, I would like to give special appreciation to Dr. George Riley and Dr. Chuanyi Ji. Thanks are also due to fellow graduate students at GNAN research group. Above all, I would like to express my gratitude in writing to my family, Young-Jin, Katie, and Jason, for their loving support and sacrifice.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

The objective of the proposed research is to address the problem of energy conservation in wireless sensor networks. Since wireless sensor networks typically consist of tiny sensors with a scarce energy resource, energy conservation is a critical issue to be addressed. This thesis addresses the energy conservation issue by tackling two fundamental problems: network topology construction and data delivery.

We first address energy-aware topology control taking into account throughput per unit energy as the primary metric of interest. Although there exists a myth that the minimum transmission power required to keep the network connected minimally results in the optimal topology for energy conservation, we motivate the consideration of trade-offs between energy consumption and throughput in determining the optimal topology. Through both experimental observations and analysis, we show that the optimal topology is really a function of the load in the network. We then propose a new topology control scheme, Adaptive Topology Control (ATC), which increases throughput per unit energy. Based on different coordinations among nodes, we proposed three ATC schemes: ATC-CP which coordinates common power among all nodes, ATC-IP which allows each node to use independent power without coordination, and ATC-MS which coordinates a power only between every two nodes. Through extensive simulations, we show that three ATC schemes outperform static topology control schemes, and particularly the ATC-MS has the best performance under all environments.

Secondly, we explore energy-aware data delivery that is robust to data losses. Because of the distinctive characteristics of point-to-multipoint communication for downstream vs. multipoint-to-point communication for upstream, the data delivery problem in sensor networks can be seen as consisting of two sub-problems: downstream (from a sink to sensors) and upstream (from sensors to a sink) data delivery. Although we address the problems as two independent ones, we eventually solve those problems with two approaches: GARUDA-DN and GARUDA-UP which share a common structure, the minimum dominating set.

For the downstream data delivery, we consider reliability as well as energy conservation since unreliable data delivery can increase energy consumption under high data loss rates. To reduce energy consumption and achieve robustness, we propose GARUDA-DN which is scalable to the network size, message characteristics, loss rate and the reliable delivery semantics. To form the basis of GARUDA-DN on the optimization perspective, we design the core structure that approximates the solution of the minimum set cover problem, which is the optimal solution of loss recovery server designation. We then solve new challenges of downstream data delivery by proposing Wait-for-First-Packet (WFP) pulses for reliable short-message delivery; candidacy-based solution for the different reliable delivery semantics; and two-phase NACK based recovery process. From ns2-based simulations, we show that GARUDA-DN performs significantly better than the basic schemes proposed thus far in terms of latency and energy consumption.

For the upstream data delivery, we address an energy efficient aggregation scheme to gather correlated data with theoretical solutions: the shortest path tree (SPT), the minimum spanning tree (MST) and the Steiner minimum tree (SMT). To approximate the optimal solution in case of perfect correlation among data, we propose GARUDA-UP which combines the minimum dominating set (MDS) with the shortest path tree (SPT) in order to aggregate correlated data. To reduce the redundancy among correlated data and simplify the synchronization among transmission, the aggregation takes two stages: local aggregation among sensors around a node in the MDS and global aggregation among sensors in the MDS. From discrete event simulations, we show that GARUDA-UP outperforms the SPT and closely approximates the centralized optimal solution, the SMT, with less amount of overhead and in a decentralized fashion.

CHAPTER I

INTRODUCTION

The advances in the integration of micro-electro-mechanical system (MEMS), microprocessor, and wireless communication technology have enabled the deployment of large-scale sensor networks. A wireless sensor network consists of two kinds of entities [89]: (1) the sink that queries and collects information; and (2) the sensor that senses environmental phenomena and reports data to the sink. In general, a few sinks and a huge set of small untethered sensors are randomly deployed in a multi-hop and ad-hoc fashion[1] to sense a physical phenomenon cooperatively. The physical phenomena include temperature, humidity, light, noise levels, and the presence or movement of certain objects.

Sensor networks have a wide range of applications [4]: military applications [101] to detect and track hostile objects in a battle field; environmental research applications [7] to monitor a seismic tremor, a tornado or a flood; industrial applications [3] to guide and diagnose robots or machines in a factory; and educational applications [82] to monitor developmental childhood or to create a problem-solving environment.

This wide variety of applications has spurred a tremendous amount of research in sensor networks over the past few years [4]. These research works have spanned over all layers of the network protocol stack. At the physical and the data link layer, energy-efficient and robust schemes, such as SMAC [102] and UWB (Ultra Wide Band) [15], have been proposed. Above these layers, new routing and transport protocols, such as Directed diffusion [36], SPIN [32], PSFQ [92], and CODA [93], have been proposed to fit the requirements of sensor networks, which include data-centric processing, in-network processing, and attribute-based naming.

Notwithstanding the fact that these research works have different goals, we notice that all of them have a common concern of minimizing energy consumption. Since wireless sensors are battery driven, and their sizes are too small to accommodate a large battery, they are constrained to operate

[1] The deployment in an ad hoc fashion does not require a centralized infrastructure. Each sensor can go through several number of hops to report data to a sink.

using an extremely limited energy budget. For instance, the total stored energy in a *smart dust mote* is only 1 J [62]. Since this small amount of energy is the only power supply to a sensor node, it plays a vital role in determining the lifetime of sensor networks. Therefore, energy conservation is a significant issue at all layers.

To tackle the problem of energy conservation, new algorithms have been proposed at all protocol layers, and they can be classified into two main categories. The first type is topology control [13], which decides per-node transmission powers and hence constructs multi-hop paths between sensors to sinks, in order to minimize energy consumption. The second type is data delivery to reduce energy consumption ranging from the data link layer [102] and the network layer [32, 36] to the transport layer [67, 71, 92, 93]. While topology control impacts energy consumption, it is important for the data delivery between sources and sinks to be conducted in an energy efficient manner as well.

This thesis, therefore, tackles the problem of energy conservation by addressing the above two categories. In particular, the following are the two contributions: (i) optimal topology control schemes to construct sensor networks for optimal energy conservation and (ii) energy-aware data delivery framework with robustness to losses. While these two problems have the common objective of energy conservation, each has its own distinct objective to be accomplished. The secondary objectives of the topology control schemes and the data delivery framework are to maximize the total throughput and to minimize the latency of delivery, respectively.

Although it is difficult to control transmission powers in a distributed fashion, optimal topology control has been explored to prolong network lifetimes [64, 97]. To date, a minimally connected topology has been regarded as an optimal solution because of its ability to reduce interference to other nodes. However, we care to identify and exploit the trade-offs between energy consumption and throughput performance in order to construct the optimal topology. Specifically, this thesis shows that the optimal topology in terms of throughput per unit energy depends on node density and traffic load of sensor networks. An adaptive topology control (ATC) scheme for throughput and energy awareness is then proposed. The ATC scheme controls the transmission power in a decentralized fashion. Therefore, the proposed scheme achieves maximum throughput performance while minimizing energy consumption simultaneously.

Since enabling communication between sensors and sinks is the major role of sensor networks, many research works [32, 36] have investigated energy-aware data delivery. Because of the distinctive characteristics of multipoint-to-point communication vs. point-to-multipoint communication, the data delivery problem in sensor networks can be seen as consisting of two problems: downstream and upstream data delivery. Although those two delivery seem to be different, they are closely related to each other due to the relationship between a query from a sink and replies from sensors at the application level. Therefore, we propose two solutions, GARUDA-DN and GARUDA-UP, which share a common delivery structure, so that we can solve those problems with a common framework.

For the downstream, a direction from a sink to sensors, wireless sensor networks experience wireless and congestion losses more severely than other wireless networks because of the low capability to recover from losses and the high node-density. Therefore, robustness is also important to energy conservation since unreliable data delivery, which increases the probability of data retransmission under high loss rates, results in the consumption of a large amount of energy. Although the problem has been addressed by previous works [28, 103] in the context of wireless ad-hoc networks, such approaches cannot be directly applied to the sensor environment due to the resource constraints of sensors. Therefore, we propose a sink-to-sensors energy-aware data delivery scheme, GARUDA-DN[2], in order to solve the downstream problem while considering robustness simultaneously.

For the upstream, a direction from sensors to a sink, most of research works on upstream data delivery have considered the redundancy of data more importantly than the ene-to-end reliability because in-network processing called data aggregation modifies the redundant data and reduce the size of them. Therefore, we propose a energy efficient data aggregation scheme, GARUDA-UP, in order to tackle the upstream problem while utilizing a common delivery structure that GARUDA-DN constructs to solve the reliable downstream data delivery problem.

The rest of the thesis is organized as follows: In Chapter 2, the applications of wireless sensor networks are classified into three basic classes and the generic challenges are discussed. In addition, previous research works are presented based on the protocol layers. In Chapter 3, the problem definition and motivation of topology control are described. Then proposed adaptive topology control

[2] A mythological bird that "reliably" transported Gods.

(ATC) schemes: ATC-CP, ATC-IP, and ATC-MS are presented. By analyzing convergence and evaluating performance, the proposed schemes are shown as energy-efficient topology control schemes. In Chapter 4, the reliable upstream data delivery is defined and motivated with simulations. To solve the reliable delivery problem, an ideal solution and a practical solution are presented. After the detailed framework called GARUDA-DN is shown, we address the other semantics of delivery using the basic framework. Then GARUDA-DN's outperforming the basic frameworks is shown through extensive simulations. In Chapter 5, the energy efficient upstream correlated data delivery is described and addressed with theoretical solution for the problem. To approximate the ideal solution, we propose GARUDA-UP which shares the concept of the minimum dominating set from the basic GARUDA-DN proposed in Chapter 4. Then the energy efficiency of GARUDA-UP is compared with the shortest path tree and an approximation of the Steiner minimal tree which are the optimal solutions for zero correlation and perfect correlation cases, respectively. Finally, in Chapter 6, the contributions of this thesis and future works are discussed.

CHAPTER II

WIRELESS SENSOR NETWORKS

2.1 Applications

In the past decade, many wireless sensor networks have been deployed. Based on basic functionalities, we categorize the applications into three basic applications: (i) data collecting, (ii) event monitoring, and (iii) object tracking application. Most of wireless sensor networks will fall into one of these basic classes or hybrid class which combines more than two basic applications.

2.1.1 Data Collecting Applications

Scientists in environmental research areas want to collect different kinds of sensor readings, e.g., temperature, humidity, pressure, noise levels and the current characteristics of objects, from a set of different positions periodically in order to gather statistics and detect trends. Based on the collected data, they analyze offline. In addition to the environmental research, military and commercial applications require the basic functionality for collecting data distributed spatially or temporally.

Therefore, these data collecting applications are characterized by having a large number of source nodes periodically sensing and transmitting data back to sinks which request and process them. Moreover, the redundant deployment of sensor nodes produces a huge amount of spatial and temporal data. However, the limited energy and transmission capacity of WSNs requires the low transmission rate and long lifetime of networks. To satisfy the above requirements, it is necessary to find an energy efficient topology and optimal routing strategies at a network layer.

In general, typical data collecting applications use a tree-based topology where a routing tree is rooted at a sink node[16]. Depending on the location of sources which can be correlated with each other, the optimal structures of trees may be different; for example, the shortest path tree is optimal in case of non-correlation among data, or the Steiner minimal tree is optimal in case of perfect correlation among data.

After the topology or network is constructed, each sensor will sample environmental data, e.g.,

temperature and humidity, based on different time intervals. The sampling rate can be from several minutes to few days depending on the significance of data. However, since small time interval produces large amount of data than longer interval, it is necessary to decide the frequency of sampling data based on the significance of data and capacity of WSNs.

Basically, those two things, optimal gathering structure and sampling frequency, will decide the lifetime of WSNs for data collection applications. Since the major role of data collection application is to gather information over long period of time, the lifetime of WSNs is one of critical metrics for the evaluation of performance.

On the other hand, the latency or delay is not a critical factor on the performance of networks because the data transmission can be compromised to improve the network efficiency. For example, delay caused by optimal transmission scheduling may be indispensable to the optimal data gathering.

For an instance of data collecting applications, environmental application[84] is a natural candidate for applying WSNs because the data to be collected are distributed over large region and long period of time. Environmental sensors are used to study vegetation response to climate trends and diseases; acoustic and imaging sensors can identify, track, and measure the population of birds and other species.

Commercial industry also has been interested in sensing as a means of lowering maintenance cost and improving performance of machines. Monitoring machines' health by collecting data for vibration or lubrication level is a typical example of industrial applications of WSNs.

2.1.2 Event Monitoring Applications

Another class of sensor network applications is an event monitoring. Compared to the data collecting applications, an event monitoring application tries to detect events, e.g., environmental anomaly, disaster, or security violation. In general, the event is different to data in terms of the spatial and temporal characteristics because the event will occur not periodically over global area but sporadically over local area. Although each node has to sense data periodically, it does not need to report data to sinks. However, the node which senses an anomaly should report the event immediately and reliably to sinks.

Once nodes detect events, they should notice the events to a set of sinks within limited time boundary, e.g., less than a few seconds. Other nodes which forward the events also should respond quickly. Therefore, at event monitoring applications, reducing the delivery latency of events is more significant than reducing the energy consumption of data delivery because of two reasons: (i) events are rare and (ii) the event delivery has time-constraint.

One of popular applications for monitoring events is a battlefield surveillance. Critical terrains, approach routes and paths can be rapidly covered with sensor networks and closely watched for the activities of the opposing forces. In case of chemical and biological warfare, WSNs can be deployed in the friendly region and used as a chemical or biological warning system to warn friendly forces so that they can reduce the casualties drastically.

Another event monitoring example is a forest fire detection. Since sensor nodes may be randomly and densely deployed in a forest, a WSN can detect a fire and relay the origin of the fire to a base station. The quick notice of a fire can prevent disaster. This application can also be installed in commercial and residential buildings to replace aged fire alarms with small amount of overhead for reinstallation.

2.1.3 Object Tracking Applications

The third class of WSN applications is to track mobile objects, e.g., vehicles (tanks and platoons) of opposing forces, natural monuments and valuable assets. The major goal of object tracking applications is to find the current location of objects or the trajectory of movement of mobile objects.

To track the objects in a simple way, a small tag which is similar to normal sensor node can be attached to the object. In WSNs, other sensor nodes are deployed to communicate directly to the sensor nodes attached to the mobile objects. These tags can receive the current location information from sensors and give their identification information to sensors so that their location can be tracked by WSNs.

If a mobile object is too hostile to attach a tag, sensor nodes or sinks need to have functionality to recognize the ID of mobile objects.

The major concern of object tracking applications is how to keep track of movements with less amount of information exchange among sensor nodes and sinks.

One of military applications to track objects is targeting. WSNs can be incorporated into guidance systems of the intelligent ammunition. In this case, each sensor should be equipped with functionality to identify mobile objects.

The tracking application can be used in commercial industry as a inventory control and management system. Sensor nodes are attached to each item in a warehouse so that users can find out the location of an item or the number of items within a specific area.

2.1.4 Hybrid Applications

Most of real applications have combined with more than two basic applications. For example, battle field applications require three basic functionalities: sensing fields, tracking objects and detecting dangerous objects.

In health areas, the WSNs can be used to track doctors and patients and collect health information of patients inside a hospital. Each patient has a small tag which has several functionalities: (i) sensing heart rate, (ii) detecting a heart attack, (iii) tracking location information, and (iv) exchanging information with doctors' sensor nodes.

To solve these multi-modal applications, we should address them with one of two approaches: a unified architecture to handle all three of applications and different architectures to address them individually. Depending on the given and limited characteristics of WSNs, one of two approaches will be chosen to solve the multi-modal applications.

2.2 Challenges in Wireless Sensor Networks

In general, wireless sensor networks share commonalities with existing wireless ad-hoc networks which use multi-hop communication without centralized coordinations. However, there are several factors which make WSNs different to wireless ad-hoc networks as follows:

- *Energy*

  Since the motto of sensor network is to develop tiny sensor nodes cheap enough to dispose without recharging battery, the energy conservation is a critical issue in WSNs. For instance, a sensor node, Mote, has a total stored energy on the order of 1J[62]. For wireless integrated

network sensors (WINS)[90], the total average system supply currents must be less than 30 mA to provide long operating life. Moreover, inaccessibility of sensor nodes after deployment makes energy consumption more critical in WSNs.

- *Scalability*

  To cover areas with sensors which have a short transmission distance due to frugal energy budget, WSNs have to scale to much large numbers (e.g., more than 10,000) of sensor nodes. Without any centralized coordination, scalability of WSNs makes itself different to wireless ad-hoc networks which have up to a few thousands of nodes.

- *Redundancy*

  Due to the frequent node failures and inaccessibility of failed nodes, WSNs are required to have high redundancy. Therefore, sensor nodes are normally deployed with a high degree of connectivity instead of minimal connectivity. Because of the high degree of redundancy, the failure of single node can be negligible. At the same time, the redundancy has negative effects on the performance of WSNs because it causes redundant transmission causing broadcast storm problem[54].

- *In-network Processing*

  In general, previous transport protocols used in wired and wireless have assumed the end-to-end approach guaranteeing that data from senders should not be modified by intermediate nodes until data reach a receiver. However, data at WSNs can be modified or reduced into smaller amount of data by intermediate nodes in order to remove redundancy of information inside data. Therefore, previous solutions cannot accommodate this new concept of in-network processing, called data aggregation or diffusion in WSNs.

- *Data Centric Processing*

  WSNs have a large scale in terms of number of nodes which cannot be assigned individually with unique identification, e.g., IP address. Therefore, sensor nodes cannot be no longer accessed by unique ID. Instead of addressing nodes with ID, it is more natural to access the data directly through content, attribute, e.g., location of node. The naming schemes in WSNs are often data-oriented.

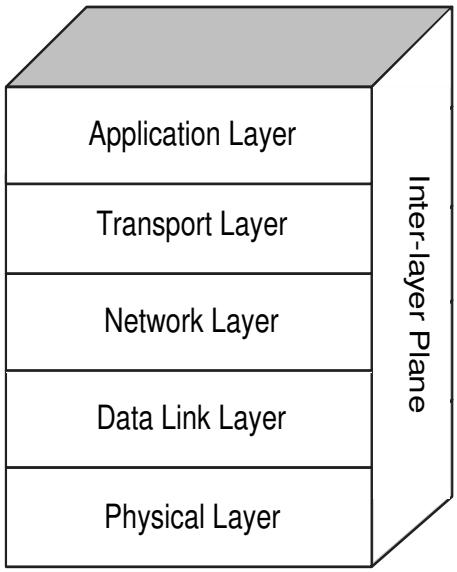


Figure 1: Protocol Layers of Nodes in Wired and Wireless Networks

For example, an environmental monitoring system requests the temperature readings through a query, such as“collect temperature readings in the region bound by the rectangle (x1,y1,x2,y2)”, instead of a query such as“collect temperature readings from a set of nodes of which addresses are x, y, and z.”

2.3 Research in Protocol Layers

In this section, we present previous research works on wireless sensor networks and classify them into different protocol layers that have been used in wired and wireless protocol classification [1] shown in Figure 1.

2.3.1 Research in Application Layer

In general, WSNs are application-specific networks. Depending on different application scenarios, there are different requirements that an application layer has to satisfy. Among a variety of research works at an application layer, query dissemination, localization, synchronization, and security are

[1] In general, WSNs cannot be classified exactly into the following layers because they require new architectures due to new challenges. We, however, use this category for the purpose of convenient explanation.

addressed.

2.3.1.1 Query Dissemination

The basic functionality of WSNs is to collect data by sending a request or command, called"*query*" through WSNs. There have been several research works [19, 77] which provide user applications with interfaces to issue queries, respond to queries, and collect incoming replies. As discussed before, these queries are generally not issued to particular nodes. Instead, attribute-based or location-based naming is preferred. Sensor query and tasking language (SQTL)[77] is proposed as an application that provides even a larger set of services. SQTL plays the role of a programming interface between sensor applications and a middleware. It is a procedural scripting language, designed to be flexible and compact, with a capability to interpret simple declarative query statements.

2.3.1.2 Localization

Localizing sensor nodes is one of popular research areas. There have been several research works including (i) exploiting received signal strength indicators, (ii) time of arrival, (iii) time difference of arrival, and (iv) angle of arrival. Despite adequate performance, techniques based on the above technologies are not quite applicable to WSNs because they require extensive hardware, infrastructure, and pre-deployment fine-tuning.

Given the inherent challenges of WSNs, a set of localization schemes are proposed. [9] assumed a heterogeneous network containing powerful nodes with established location information. And these nodes beacon their position to neighbors that keep an account of all received beacons. Using this proximity information, a centroid model is applied to estimate the listening nodes' location. In [55], DV-HOP consists of heterogeneous nodes: beaconing nodes which flood their location information throughout a network maintaining a running hop-count at each intermediate node and listening nodes which estimate their location based on the received beacon locations and the hop-count from the corresponding beacon. Additionally, there were other approaches that integrated beacons or anchor nodes with precise location information and used the iterative increase in precision by distributed algorithms in [66, 73, 76].

2.3.1.3 *Synchronization among Sensors*

Time synchronization is a critical issue for any distributed system. Since most of WSNs are operated in a distributed fashion, WSNs make particularly extensive use of synchronized time: for example, to integrate a time-series of proximity detections into a velocity estimate [11]; to measure the time-of-flight of sound for localizing its source; to distribute a beamforming array [100]; or to suppress redundant messages by recognizing that they describe duplicate detections of the same event by different sensors [36]. Sensor networks also have many of the same requirements as traditional distributed systems: accurate timestamps are often needed in cryptographic schemes, to coordinate events scheduled in the future, for ordering logged events during system debugging, and so forth.

2.3.1.4 *Security*

Security issue for WSNs is still a wide open area. Although several works seem to be directly adopted from ad-hoc networks, the principal threats and possible attacks to the correct functioning of WSNs are still missing a theoretical analysis due to the challenges: limited processing power, storage, bandwidth, and energy.

[60] introduces the Security Protocols for Sensor Networks (SPIN) comprised of Sensor Network Encryption Protocol (SNEP) and μTESLA. The function of SNEP is to provide confidentiality (privacy), two-party data authentication, integrity, and freshness. μTESLA is to provide authentication to data broadcasts. SPINS presents an architecture where the base station accesses nodes using source routing.

2.3.2 Research in Transport Layer

Although TCP is the most popular transport protocol in wired and wireless networks, it cannot be used in WSNs directly. TCP is the end-to-end protocol which guarantees reliable delivery without information modification. WSNs, however, have a new delivery concept called *data aggregation* which requires in-network processing instead of end-to-end processing[2].

And WSNs also have a point-to-multipoint (direction from a sink to sensors) or multipoint-to-point (direction from sensors to a sink) communications while normal TCP assumes point-to-point

[2] In [89, 4], they address that in-network processing combines transport layer with network layer in WSNs.

communication model. Therefore general approaches for point-to-point model cannot be directly adopted into WSNs. We address two major functionalities of transport layer: congestion control and reliability.

2.3.2.1 Congestion Control

Since upstream direction has more amount of data compared to downstream, most of works for congestion control concentrate on the upstream flows which result in multipoint-to-point communications. To address multipoint-to-point traffic of WSNs, ESRT[71] includes a congestion control component that serves the dual purpose of achieving reliability and conserving energy. The algorithms of ESRT mainly run on the sink, with minimal functionality required at resource constrained sensor nodes. ESRT protocol operation is determined by the current network state based on the reliability achieved and congestion condition in the network. If the event-to-sink reliability is lower than required, ESRT adjusts the reporting frequency of source nodes aggressively in order to reach the target reliability level as soon as possible. If the reliability is higher than required, then ESRT reduces the reporting frequency conservatively to conserve energy while still maintaining reliability.

2.3.2.2 Reliability

RMST[83] and PSFQ[92], both proposed as reliable transport layer solutions for sensor networks, attempt to provide end-to-end reliability at upstream (from sensors to a sink) and downstream direction (from a sink to sensors) with minimal cost, respectively. RMST acts as a filter in Directed Diffusion[36], tracking fragments so that receiver initiated requests can be instantiated when individual pieces of an application payload get lost. PSFQ caches packets along the path to the sender, initiating fragment recovery as required, starting with its local neighborhood.

2.3.3 Research in Network Layer

At a network layer of WSNs, we need to address “data aggregation” as well as routing problems. The routing problems in WSNs also have a new feature changing address-centric routing to data-centric routing.

2.3.3.1 Data-centric Routing

There are a set of routing schemes proposed in wireless ad-hoc networks, such as DSR [8] and AODV [59]. Most of these routing schemes use a flooding method to discover a path to a destination. However, WSNs cannot use the flooding method because they may contain at least thousands of nodes and even up to millions of nodes. The flooding over large scale and high density sensor networks will cause broadcast storm problem[54] which results in significant energy consumption and finally a network breakdown.

Data-centric routing and location awareness of sensor nodes motivate the location-based routing schemes because location is of interest of most of applications rather than specific nodes' IDs.

MFR [85] proposed a routing scheme that forwards a packet to the node that makes the most progress toward a destination. [23] proposed a greedy geographic forwarding protocol with limited flooding to circumvent the voids (a special region having a node with the Euclidean shortest distance to a destination, which does not have a path to the destination) inside a network. And GPRS [40] uses the perimeter forwarding method to get around voids.

2.3.3.2 Data Aggregation

Since WSNs deploy large number of sensor nodes with high density, data from sensors have high level of redundancy. To gather these data from sensors to a sink efficiently, data aggregation techniques are proposed. Most of previous works focus on application dependent data aggregation techniques, in which aggregation depends on the application layer information.

A recent work[16] starts to consider the data aggregation problem in a viewpoint of optimization that minimizes delivery cost by reducing redundancy among data. [16] proposes a simplified information model and tries to solve the problem of gathering by aggregating correlated data with two simple heuristics. It considers a correlation model and proposes two simple heuristics for a given correlation factor ($0 < \rho < 1$): (1) leaves deletion heuristic: the resulting spanning tree is considered as a good approximation of the optimum minimum cost spanning tree; (2) balanced shortest path tree and multiple traveling salesman problem (TSP) tree: This solution is based on the assumption that the optimum solution should be a combination of partial SPT and partial TSP.

[35] proposes a simple modification to the directed diffusion to come up with a structure that

encourages aggregation. Briefly, it uses a greedy incremental tree to aggregate the information from the different sources where the sources try to reach the aggregation tree that is already constructed in the least number of hops.

2.3.4 Research in Data Link Layer

The data link layer is responsible for medium access control (MAC) and error control. It ensures reliable point-to-point and point-to-multipoint connections in a link level.

2.3.4.1 Medium Access Control

MAC is an effective methodology that allows devices on a network to share their interconnecting media. Although there are large number of works on MAC for wired, wireless cellular and ad-hoc networks, they cannot be used in WSNs efficiently.

Since WSNs use battery-operated computing and sensing devices, we expect sensor networks to be deployed in an ad-hoc fashion, with individual nodes remaining largely inactive for long periods of time in order to save energy. If something is detected, then they should become active. These characteristics of sensor networks and applications motivate a MAC that is different from traditional wireless MACs such as IEEE 802.11 in almost every way: energy conservation and self-configuration are primary goals, while per-node fairness and latency are less important.

To reduce energy consumption, S-MAC [102] uses three techniques: (i) periodic sleeping mode, (ii) virtual clusters formed by neighboring nodes to synchronize on sleep schedules, and (iii) in-channel signaling to reduce contention latency.

In [99], a CSMA (carrier sensing multiple access) based MAC scheme for WSNs is presented. Traditional CSMA based schemes are deemed inappropriate because they assume stochastically distributed traffic which is independent to each other. However, [99] found that two components, listening mechanism and backoff scheme, are energy efficient and robust to collision.

2.3.4.2 Error Control

Wireless channels provide error rates that are typically around 10^{-2}. Such high error rates result from multi-path fading which characterizes mobile radio channels. To increase the apparent quality of a communication channel there exist two distinct approaches: (i) forward error correction (FEC)

which employs error correcting codes to combat bit errors by adding redundancy (henceforth parity bits) to information packets before they are transmitted and (ii) automatic repeat request (ARQ) wherein only error detection capability is provided and no attempt to correct any packets received in error is made; instead it is requested that the packets received in error be retransmitted.

To the best of our knowledge, these two approaches are not explored enough at WSNs. In [78], convolutional coding effects for FEC have been considered. They found that FEC is generally inefficient if the decoding is performed using a micro-processor and recommended an on-board dedicated Viterbi decoder.

2.3.5 Research in Physical Layer

The physical layer is responsible for frequency selection, carrier frequency generation, signal detection and modulation.

2.3.5.1 Frequency

For a frequency of radio links, there are several options: (i) industrial, scientific, and medical (ISM) bands, which offer license-free communication in most countries; (ii) infrared which also is license-free and robust to interference from electrical devices; and (iii) optical medium.

According to [61], certain hardware constraints and trade-off between antenna efficiency and power consumption limit the choice of a carrier frequency for such transceivers to the ultrahigh frequency range, such as 433MHz or 914MHz in ISM band. Most of the current hardware for sensor nodes is based on RF circuit design. For instance, the low-power sensor device in [99] uses a single channel RF transceiver operating at 916MHz.

Infrared based transceivers are cheaper and easier to build. Many of laptops, PDAs and mobile phones offer an infrared data association interface. However, it has a major drawback which between a sensor and a receiver, requires a line of sight (LOS).

The smart dust mote in [96] uses optical medium for transmission with two schemes: (i) passive transmission using a corner-cube retroreflector (CCR) and (ii) active communication using a laser diode and steerable mirrors.

2.3.5.2 Modulation

Since modulation schemes affect the reliable communication and energy consumption of WSNs, there are several research works on modulation schemes.

[78] compared binary modulation to M-ary modulation schemes and argued that under start-up power dominant condition, the binary modulation scheme is more energy efficient.

Ultrawideband (UWB) has been used for baseband pulse radar and ranging systems and has recently drawn considerable interest for communication applications[15]. Since UWB uses baseband transmission, it does not require intermediate carrier frequencies. Low transmission power and simple transceiver circuitry make UWB an attractive modulation scheme of WSNs.

2.3.6 Research in Inter-Layer Planes

There are a few inter-layer planes, which help sensor nodes to coordinate the sensing task and reduce the overall energy consumption: topology control and mobility management.

2.3.6.1 Topology Control

The topology control plane manages the transmission of sensor nodes so that the topology of networks can be maintained to minimize energy consumption and maximize throughput performance.

There exist considerable previous works addressing the topology control problem of minimizing transmission power, guaranteeing network connectivity. For example, [97] proposed a fully distributed algorithm that only relies on directional information between nodes. [64] presented a centralized topology control algorithm, along with a distributed heuristic.

Unlike the above deterministic guarantee of connectivity, [72] analyzed the connectivity of a sensor ad hoc network using a probabilistic approach in order to find out the minimum transmission power to be used at all nodes. The lower and upper bound on the probability of network connectivity are derived for certain transmission range assignments.

CHAPTER III

ATC: ADAPTIVE TOPOLOGY CONTROL

3.1 Problem Definition

The battery power in mobile nodes is a premium resource in sensor networks. Not surprisingly, energy-aware protocols have been proposed at several layers of the protocol stack [12, 52, 63, 68, 75, 79, 80]. More recently, using effective topology control to optimize energy usage in the network has come into focus. Briefly, the topology control problem can be defined as the determination of the optimal transmission range (and hence the transmission power) to be used by the mobile nodes in the network - with the choice of the transmission range directly impacting the effective network topology.

Several related works have investigated the problem of topology control from the perspective of optimal throughput per unit energy performance [1, 43, 51, 64]. The common thesis of the aforementioned works is that the transmission range used by mobile nodes should be the minimum required to keep the network connected. We refer to such a topology as the *minimally connected topology* in the rest of the thesis.

The optimality of the minimally connected topology can be explained intuitively as follows. When the transmission range is decreased, the average hop-count for the paths traversed by flows increases linearly. However, the transmission power per hop decreases super linearly (given that the path loss exponent typically ranges from 2 to 4). Hence, the overall energy consumption in the network for the same amount of data transferred is minimized in a minimally connected topology. In [1, 43, 51, 64], the authors also propose distributed algorithms to achieve approximations of a minimally connected topology.

While the optimality of the minimally connected topology in terms of throughput per unit energy

performance is indeed correct for a generic[1] sensor network, there is a key but straightforward phenomenon that changes the optimality. Specifically, when the transmission range in a network is decreased, the average hop-count of flows in the network increases, which in turn increases the total number of one-hop flows[2] in the network, thus increasing the aggregate *induced load* in the network[3].

However, a decrease in the transmission range also increases the *spatial-reuse* in the network because of the smaller interference ranges, thus increasing the network capacity (total number of bits that can be transmitted in the network in unit time). It can be shown that, with a decrease in transmission range, while hop-count increases linearly, the spatial-reuse in fact increases quadratically since the inhibition area of a transmission is proportional to the square of the transmission range (assuming omni-directional transmissions). Hence, any increase in the aggregate induced load in the network is easily offset by the higher spatial-reuse in the network, thus leaving the throughput unaffected for the same basic load in the network. Alternately, it can be argued that decreasing the transmission range allows for the basic load to be increased further.

Based on this intuitive reason, most of research works for topology control have concentrated on the minimally connected topology to minimize the energy consumption as well as the throughput.

3.2 Motivation

3.2.1 Terminology and Models

In this section, we define the terminology and describe simulation model used in the following preliminary observation.

3.2.1.1 Terminology

- Topology Control vs. Power Control

  The topology of a sensor network consists of sensors and wireless links between the sensors.

[1] A generic sensor network is not restricted to the node density [29]. Therefore, it can consist of several thousands of nodes or more.

[2] We define such flows as mini-flows later in the thesis.

[3] We distinguish the basic load offered by the sources of the flows from the induced load that is a basic load multiplied by the average number of hops traversed by flows.

In a wireless sensor network, a link between two nodes (sender and receiver) is determined by the transmission power used at the sender. While the topology will be inevitably affected by mobility or interference, it can also be controlled intentionally by adapting the transmission power. In this thesis, we focus on topology control through adaptation of transmission power in order to acquire a desired topology. To distinguish power control [52] from topology control, we confine the terminology of power control to a scheme that tunes a power to reduce energy consumption without changing the topology.

- Typical Sensor Network vs. Generic Sensor Network

  In this work, we define a *typical sensor network* as a sensor network with a few hundred nodes distributed in an area of a few square miles. This definition of the typical sensor network is also consistent with the focus of a large body of related work including [12, 44, 64].

- Mini-channel

  We define a *mini-channel* as a maximal sub-section of the network where at most one transmission can take place for any given transmission slot. For example, in IEEE 802.11, when a transmission is in progress, the neighbors of the source and the destination cannot be involved in another transmission simultaneously.

- Mini-flow

  We refer to a one-hop transmission within a mini-channel as a mini-flow. Note that an end-to-end flow will consist of *hop-count* number of mini-flows.

- Spatial Re-use Factor

  We define *spatial-reuse factor* as the number of simultaneous transmissions observed during a transmission slot. It can be measured as the total number of transmissions divided by the maximum number of transmissions within a mini-channel during the simulation period. Conceptually, the spatial-reuse factor will be a measure of the number of mini-channels in the network.

- Contention Time

The *contention time* is defined as the average of the sum of the back-off times and the transmission times experienced by packets traversing a node.

- Utilization of CSMA/CA

  We define *utilization* as the total throughput of a mini-channel. We study the utilization as a function of the number of mini-flows contending in a mini-channel. Figure 4 is an illustration of the utilization curve of a typical MAC protocol including CSMA/CA.

- Static Topology Control using Constant Transmission Power

  We define *a static topology control* as an approach wherein *all stations in the network* use the same constant transmission power. Further, the stations *do not adapt the transmission power* based on network conditions.

3.2.1.2 Models

- Simulation Model for Energy Consumption

  To emulate a realistic environment, we measure power by monitoring three components: (i) transmission power required to send a packet, (ii) receiving power required to receive or listen to a packet, and (iii) idle power required to stay awake. The transmission power includes both the power required to drive the circuit and the transmission power from the antenna. The power required to drive the circuit is set to 1.1182W [12], while the antenna transmission power is computed based on the transmission range using the two-ray ground reflection model, and is equal to $7.2 * 10^{-11} * d^4$W for a transmission range of d meters [12]. The receiving and idle power values are assumed as 1W and 0.83W respectively [12].

- NS2 Environment

  The wireless physical layer in *ns2* is based on the IEEE 802.11 DS/SS specifications. The signal propagation model is a combination of the free space propagation model (for distances less than 100m) and the two-ray ground reflection model (for distances greater than 100m) [18, 20]. The data rate of the underlying channel is 2Mbps. To exclude the effect of transport layer protocol, we use a constant bit rate traffic over UDP for the sources. The packet size is set to 512 bytes. Source destination pairs are randomly chosen from the network stations.

The IEEE 802.11 protocol in the distributed coordination function mode (CSMA/CA) is used at the MAC layer. Dynamic source routing (DSR) [8] is used as the routing protocol. Transmission range is varied from the minimum range required to keep the network connected to the maximum range required to make the network fully connected.

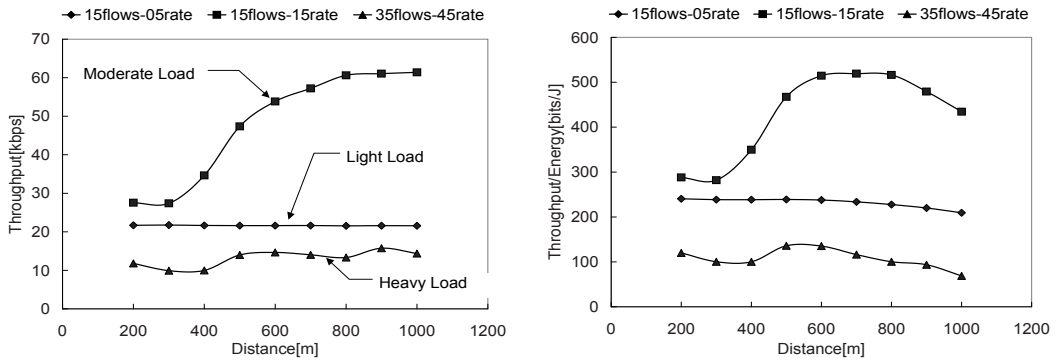
3.2.2 Preliminary Observations

In this section, we first use simulations to study the throughput per unit energy performance of a sensor network under different load conditions, and when using different transmission ranges. In the process, we observe that the minimally connected topology using minimum transmission power could not obtain optimal performance under higher traffic loads. Finally, we provide a simple analysis to explain the observed results and motivate the need for adaptive topology control that reacts to the load conditions in the network.

We simulate three different kinds of environments: (i) low traffic load, (ii) moderate traffic load, and (iii) heavy traffic load. For three different load conditions, we use 5 flows, 15 flows and 35 flows with packet transmission rates of 5, 15, and 45 packets (512 bytes packet size) per second, respectively. We observe throughput and throughput per unit energy as a function of the transmission distance. Extensive results for several other environments, such as different node densities and different network sizes, are presented in [56, 57].

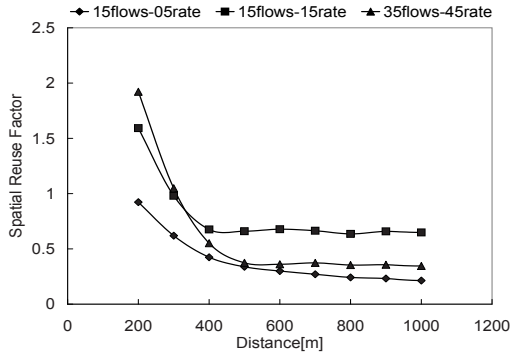
We present the following metrics for all the simulation results: (a) Per-flow throughput measured in Kbps, (b) Per-flow throughput per unit energy measured in bps/Watt, (c) Spatial-reuse factor, (d) Average per-flow hop-count measured in hops, (e) Contention time, and (f) Utilization of the IEEE 802.11 MAC protocol.

From Figure 2(a), it can be observed that (i) for the lightly loaded scenario, the maximum per-flow throughput is achieved at a transmission range of 300m (Because transmission range of 200m cannot guarantee the connectivity at all scenarios, 300m is considered as a minimum transmission power.); (ii) for the moderately loaded scenario, the maximum per-flow throughput is achieved at a transmission range of approximately 800m, and (iii) for the heavily loaded scenario, the utilization is poor and the maximum throughput is achieved approximately at 1000m (the throughput curve is relatively flat for this scenario and close to maximum throughput is achieved at 500m).

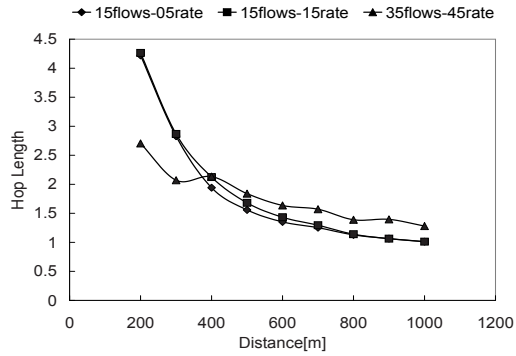


(a) Throughput

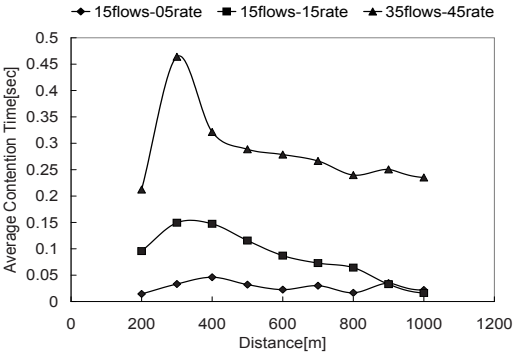
(b) Throughput Per Unit Power



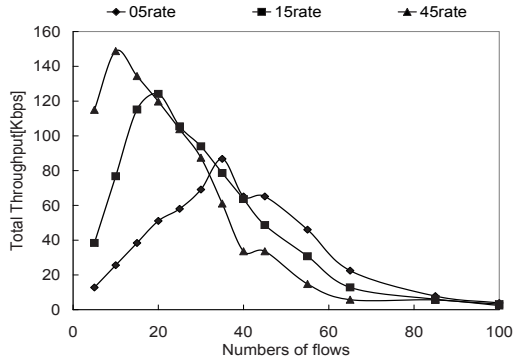
(c) Spatial Reuse



(d) Hop Count



(e) Contention Time



(f) IEEE 802.11 Utilization

Figure 2: Simulation Results in case of Varying Traffic Load, Fixed Number of Nodes, and Fixed Network Size (100 nodes are located in 1000m$\times$1000m area. All figures show a function of transmission distance except Figure (f).)

This illustrates the fact that for a given topology, the optimal transmission range (in terms of throughput performance) is variable, and is a function of the load in the network. It is important to note that merely observing throughput does not reveal the true performance as a transmission range of 1000m (for the scenarios considered) will achieve a high throughput, but will also have a transmission power that is approximately 18 times more than the transmission power required for a transmission range of 500m. Hence, we also present the throughput per unit energy results for the scenarios. The peaks of this result are at 300m, 600m, and 500m for the lightly loaded, moderately loaded, and heavily loaded scenarios, respectively.

We now highlight the impact of the transmission range on the different factors that affect performance, namely spatial-reuse, hop-count, and MAC contention time (which is representative of the induced load relative to the network capacity). Figures 2(c), 2(d), and 2(e) show the variation of the above factors with transmission range. The spatial-reuse factor stays below 2 for all scenarios while the hop-length goes up to 3 for the minimal transmission range (300m).

An equally revealing result is that the contention time shows a peak at around 300m for the moderate and heavy load scenarios (the decrease in contention time at 200m is due to the partitioned condition of the network at that transmission range). Such an increase in contention time reveals the higher induced load in the network when the transmission range is decreased. Since magnitude of the basic load by itself is high in these scenarios, such an increase in the induced load can push the operating point of the MAC protocol (see Figure 2(f)) down to the over-utilized region. We illustrate the relationship between the contention time and the MAC utilization in Chapter 3.

Under low load conditions, the contention time does not show any significant increase as the induced load due to the transmission range decrease is easily absorbed by the MAC utilization scalability. In other words, since the operating point in the utilization curve to start with is in the under-utilized region, the increase in induced load merely pushes the operating point up the MAC utilization curve.

The adaptive topology control algorithms that we present in Chapter 3 are based on *achieving the contention time equivalent to non-over-loaded network conditions*.

3.3 Related Works

- [43, 44] conceptualize the power control problem and provide a protocol which suggests that low common transmission power maximizes throughput capacity, extends the battery life, and reduces the contention at the MAC layer. Although their inference is valid in a general sense, we show that the low common transmission power cannot always provide the optimal throughput in typical ad-hoc environments.

- In [64], the authors proposed two transmission power control algorithms to create 1-connected and 2-connected (bi-connected) topologies. The results presented show that such minimally connected topologies improve the throughput and power consumption significantly. While the 2-connected topology delivers better performance, the authors do not study further degrees of connectivity or the impact of the different factors considered in this paper.

- [97] proposes a distributed power control algorithm based on directional information. Each station increases transmission power until it finds a neighbor node in every cone of angle α, where $\alpha <= 2 * \pi/3$, to guarantee a maximum connected node set. The resulting network topology increases network lifetime by reducing transmission power and reduces traffic interference by having low node degrees. The work infers its results from an average degree of stations in the network, and does not use traffic to study either throughput or throughput per unit energy.

- In [70], authors study the effects of transmission range on AODV's multicast performance at varying transmission ranges. They show that increasing the transmission range has pros and cons in an AODV multicast environment; they conclude that the transmission range should be adjusted to meet the targeted throughput while minimizing battery power consumption.

- In [52], authors propose a power control method at the MAC layer that finds the lowest power level required between two communicating nodes. The method reduces total interference and hence increases throughput of the wireless network. The mechanism assumes that the underlying topology is decided by an external approach. After the topology is pre-determined, their method increases throughput by reducing interference. We, however, show that the

underlying topology can itself be adapted to achieve optimal performance.

3.4 Theoretical Analysis

This section provides a theoretical reason of the observed results.

3.4.1 Per-flow Throughput

Several related works have analyzed the capacity and the performance of wireless sensor networks. In [30], the authors derive the end-to-end throughput of a sensor network as a function of the number of nodes n, to be $O(\frac{1}{\sqrt{n}})$. In this section, since transmission range is the variable of interest, we derive the end-to-end per-flow throughput of a sensor network as a function of the transmission range.

To derive the per-flow throughput, we first define the capacity of a wireless multi-hop network. The capacity is the maximum bound that can be achieved only if all flows are one-hop mini-flows. Considering the impact of the 802.11 (CSMA/CA) MAC protocol on the throughput, we can assume that a wireless network consists of several independent regions (mini-channels) in which transmissions can occur simultaneously. Hence, (1) defines the total capacity C of a multi-hop network as the sum of the capacity Γ_{mc} of each mini-channel. σ is a spatial reuse factor representing a number of mini-channel in a network.

$$C = \sigma \times \Gamma_{mc} \tag{1}$$

While the capacity aggregates mini-flows' throughput in different mini-channels, the per-flow throughput averages them belonging to a same flow. Since mini-flows share a mini-channel, the per-flow throughput is derived as like

$$\Gamma = \frac{C}{f_{mf}} = \frac{\sigma \times \Gamma_{mc}}{f \times h}, \tag{2}$$

where f_{mf} is the total number of mini-flow in a network, f is the number of flows, and h is the average hop-count for per flow.

To evaluate the impact of transmission power on throughput, we assess the impact of transmission power on each of the parameters in (2). Since the number of flows f is an uncontrollable

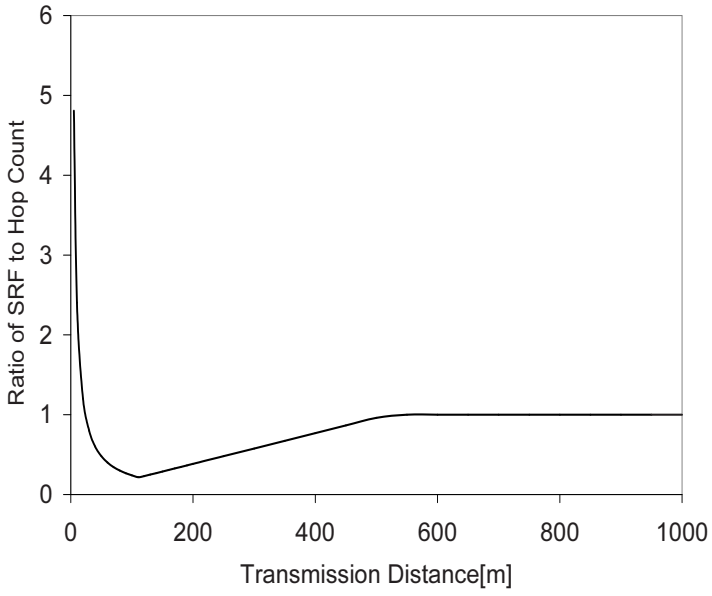


Figure 3: The Ratio of Spatial Reuse Factor to Hop Count as a Function of Transmission Distance

parameter, σ, h, and Γ_{mc} are the parameters of interest. At first, the ratio of σ to h is derived as a function of transmission power and the impact of the ratio on throughput is evaluated. Then the impact of Γ_{mc} is analyzed.

3.4.2 Spatial-Reuse Factor and Hop-Count

In generic sensor networks, the increase of spatial-reuse factor is what allows the higher induced load due to transmission range decrease, to prevent any lowering of the throughput. However, as observed in 2 (c) and (d), σ does not increase at the same rate as hop-count, and further does not exceed the hop-count in terms of the absolute value. Hence, the induced load in the network is bound to increase with decreasing transmission range.

To verify the assumption about spatial-reuse factor and average hop-count per flow, we approximate the two factors as:

$$\sigma = \frac{D^2}{\tau * r^2} \tag{3}$$

and

$$h = \frac{ED}{r} = \frac{0.52 * D}{r}, \tag{4}$$

where τ is a function of the inhibition region of the MAC protocol used ($\tau \approx 5$ for CSMA/CA), D is the size of square-shaped network grid, and r is a transmission range. The derivation of σ in (3) comes from the calculation of a number of mini-channels, which are circles with a radius r, at the D^2 square area.

To derive h in (4), the expected Euclidean distance (ED) between two nodes(a source and a destination of a flow) randomly chosen within a square grid of size D was calculated as follows:

$$ED = 4 * D * \int_0^1 \int_0^1 \sqrt{x^2 + y^2}(1-x)(1-y)dxdy = 0.52 * D. \tag{5}$$

Using the above equations (3) and (4), in Figure 3 we show the ratio of the spatial-reuse factor σ and the average hop-count h as a function of transmission range r. It is observed that the spatial-reuse factor begins to surpass the hop-count only when the transmission range is less than 20 meters.

To estimate the minimum number of nodes which cover D^2 area with a 20 meters of transmission range, we use the relationship between the number of node n and transmission range r as:

$$n = \frac{D^2}{\pi * r^2} \tag{6}$$

According to (6), the 20 meter transmission distance can be obtained only at very high node densities (for instance of a 1000m $\times$ 1000m area, more than 3183 nodes are required).

Hence, under typical sensor environments which have less than 3000 nodes, the ratio $\frac{\sigma}{h}$ decreases as r decreases. Since the ratio $\frac{\sigma}{h}$ is proportional to the throughput of a sensor network, we can explain the phenomenon which a throughput decreases as a transmission range decreases

However, note that this translates into poor performance only for the moderate and heavy load conditions as seen in Section 3.2. For the low load scenarios, the above observation seemingly does not have any impact on the throughput. We explain this in the next section.

3.4.3 Mini-channel Utilization

Given that the ratio $\frac{\sigma}{h}$ decreases with a decrease in transmission range in typical sensor networks, we now explain the unique impact of the network load on the throughput observed by flows as the

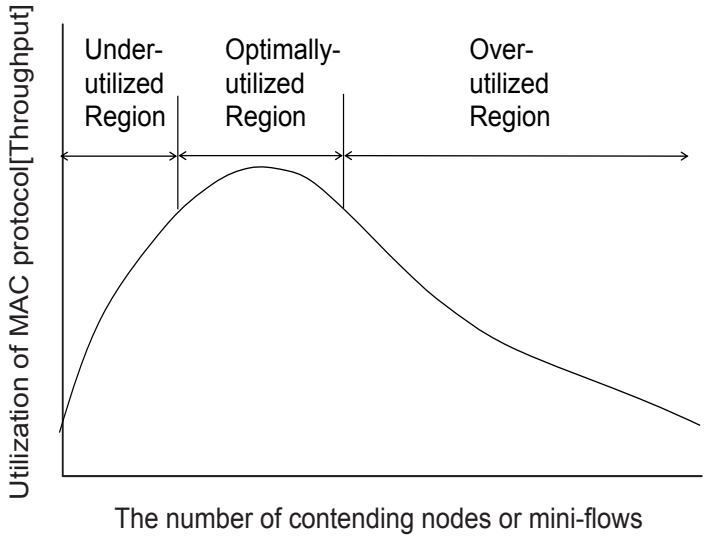


Figure 4: Utilization of a CSMA/CA MAC Layer Protocol

transmission range is changed. In other words, we focus on the parameter Γ_{mc} in (2).

In general, the throughput of a mini-channel is proportional to the utilization of the MAC layer protocol within the mini-channel. Figure 4 shows the utilization curve for a typical MAC protocol (within a mini-channel) as a function of the number of nodes which share the mini-channel [87].

In the under-utilized region, the offered load is lower than the maximum MAC capacity and hence the observed throughput is the offered load itself. In the over-utilized region, the performance of the MAC suffers drastically and the observed throughput consequently is a small fraction of the offered load. In the optimally utilized region, the MAC is fully utilized and the observed throughput is close to the offered load. Essentially,

$$\Gamma_{mc} = min(L, U(L) * C_{MAC}) \tag{7}$$

is the capacity of a mini-channel, where L is the offered load to a mini-channel, and $U(L)$ is the normalized utilization factor of the MAC protocol at the load L and the MAC capacity C_{MAC}.

In other words, since the offered load L is proportional to $\frac{f \times h}{\sigma}$, (7) can be written as

$$\Gamma_{mc} = min(\frac{f \times h}{\sigma}, U(L) * C_{MAC}) \quad (8)$$

Using the above equation as the basis, we proceed to explain the impact of transmission range decrease under the three different traffic load conditions:

- *Low load:* At low loads, $U(L)$ scales with increasing load, and hence Γ_{mc} increases as $\frac{f \times h}{\sigma}$ when the transmission range decreases. Note that this translates to a maintenance of the observed throughput for the same basic offered load.

- *Moderate load:* At moderate loads, $U(L)$ is already near the peak of the utilization curve. Hence any increase in the load (due to transmission range decrease) that will decrease $U(L)$ will cause Γ_{mc} to fall as $U(L) * C_{MAC}$. Coupled with the fact that $\frac{\sigma}{h}$ also decreases due to the decrease of transmission range in Figure 4, this phenomenon reduces the throughput observed by end-to-end flows.

- *Heavy load:* Under heavy basic loads, $U(L)$ is already down on the utilization curve. Any increase in the load will decrease $U(L)$ further, but only marginally (since U(L) is already quite low). Hence the impact on the observed performance is not much.

3.4.4 Summary of Motivation for Adaptive Topology Control

From the analysis of the impact of each parameter on the throughput, the influence of spatial-reuse factor σ and hop-count h is less than what they are expected in a typical sensor network. Instead, the throughput of a mini-channel Γ_{mc} is revealed to be the major factor on deciding the throughput Γ in typical sensor networks.

In summary, under low load conditions, it is desirable to operate using the minimally connected topology as the throughput performance does not degrade, but at the same time the energy consumption is minimized. *However, for moderate loads, it is desirable to reduce the transmission range only to the point where the induced load causes the utilization of a mini-channel to reach its peak.* At this transmission range, the throughput is still maintained, and the energy consumption is reduced to the minimum possible while not pushing the mini-channel to the over-utilization region. For the heavy load conditions, since the basic load by itself has pushed the mini-channel utilization

to the over-utilized region (thus causing the throughput to be very low), it is desirable to operate at the minimum transmission range required to keep the network connected. However, ideally the network should not be operated with the basic load exceeding the capacity of the network.

In the rest of the thesis, we propose an adaptive topology control algorithm that can adapt a transmission power to the load conditions in the network when determining the optimal transmission range.

3.5 Design Goals and Key Ideas

In this section, we present the key goals and design elements of a load-sensitive adaptive topology control strategy.

3.5.1 Goals

The following are the key goals that the design of our proposed topology control strategy is based on:

- Adaptiveness: Since the traffic load in a network can change dynamically, the topology control strategy should be adaptive to the network conditions.
- Local Information: Since using global information in a distributed environment such as a sensor network can incur high overheads, the topology control strategy should use purely local information in its approach.
- Localization: If the traffic load in one sub-area of the network changes, ideally the topology control strategy should impact only nodes within that sub-area. This goal is assuming that there is no fundamental requirement for all nodes in the network to use the same transmission power.
- Convergence: The topology control strategy should converge once the network conditions stabilize. It is also essential for a distributed scheme to converge faster than the typical rate of dynamics in the network.

- Network Partitions: In wireless sensor networks, network partition is a very critical problem to solve. However, we assume that a solution to connect the network exists orthogonal to the topology control strategy proposed in this thesis [64, 97]. Hence, the proposed schemes will limit the transmission power between the minimum power that is required to connect the network minimally (as indicated by the connectivity approach) and the maximum power that a node can transmit.

3.5.2 Key Ideas

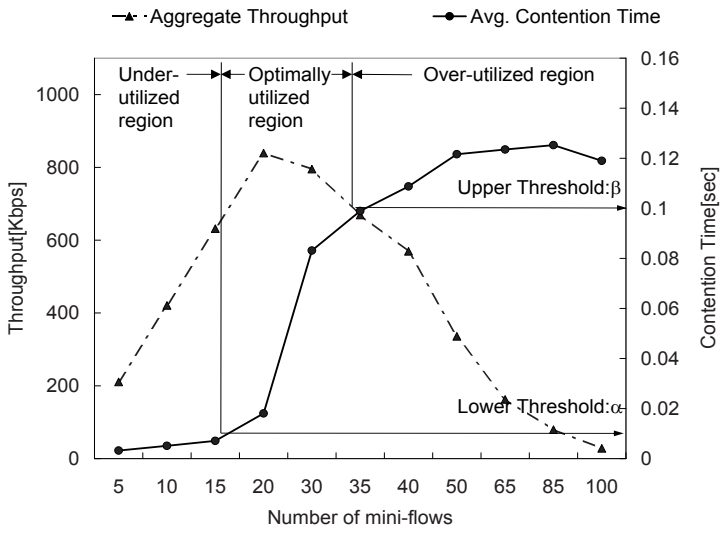
Because of the shared channel nature of wireless sensor networks, contentions and collisions occur among neighbors within each mini-channel. Assuming that there is no contention among adjacent mini-channels, the total throughput Γ of a sensor network can be approximated as (2).

The throughput Γ_{mc} within a mini-channel (shown in Figure 5) is a function of the traffic load, represented by the number of mini-flows f_{mc} within a mini-channel. The number of mini-channels σ (shown in Figure 6) is a function of the transmission distance r. Since the number of mini-flows within a mini-channel is a function of the transmission distance, the estimated throughput Γ is a function of the transmission distance. Thus, if global information about the number of mini-channels, and local information about load within a mini-channel are known, the optimal transmission distance that maximizes the throughput of a sensor network can be estimated. *However, it is unreasonable to expect the availability of such information in a sensor network due to its distributed nature.*

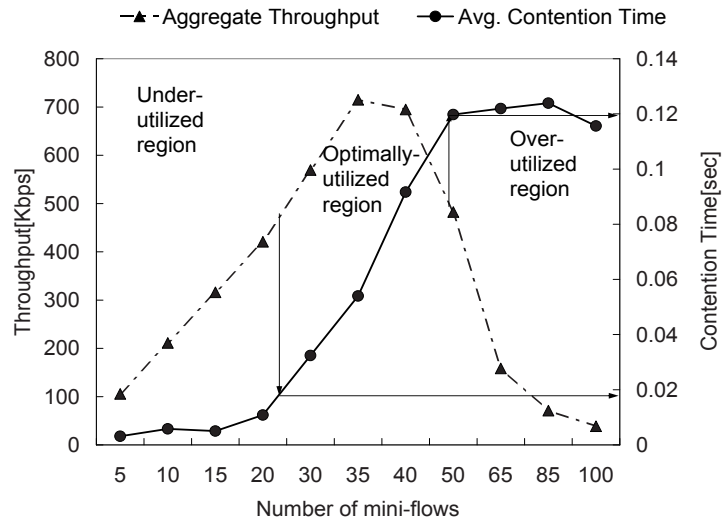
Hence, we adopt a completely distributed method to increase the throughput of each mini-channel. To ensure the maximum utilization of mini-channels, the appropriate traffic load, such as the optimal number of mini-flows, should be maintained within a mini-channel. The method consists of two phases: (1) estimating utilization (traffic load) of a mini-channel and (2) adapting transmission power to maximize utilization of a mini-channel.

3.5.2.1 Estimation of Traffic Load within a Mini-channel

To estimate exactly the current traffic load of each mini-channel, each node measures the local contention time defined in Section 3.2.1.1. Because the contention time is the delay experienced when a node tries to acquire the wireless access medium, the time is a measure of traffic load. A



(a) 15 Packets Per Second at Each Flow



(b) 5 Packets Per Second at Each Flow

Figure 5: Relationship between Traffic Load and Contention Time as a Function of the Number of Mini-flows within a Mini-channel

larger contention time indicates a higher traffic load in the mini-channel. Figure 5 (a) and (b) show the relationship between traffic load and contention time in a mini-channel for two different packet transmission rates. Similar relationships hold for other rates of packet transmissions. The dotted line is the utilization of a mini-channel (which is equal to the aggregated throughput in the mini-channel) as a function of the number of mini-flows within the mini-channel. The solid line is the average contention time, which was measured at all nodes within a mini-channel, as the number of mini-channels was increased. In Figure 5 (a), 20 is the optimal number of mini-flows that can achieve the maximum efficiency at the MAC layer. If the number of mini-flows is below 20, the mini-channel remains under-utilized. Beyond 20 mini-flows in a mini-channel, the mini-channel is over-utilized and has a lower throughput because of severe contention among mini-flows.

To identify the current traffic load region, every node uses two thresholds for the contention time: (i) α, a lower bound for an optimally utilized region and (ii) β, an upper bound for the optimally utilized region. Figure 5 (a) shows the optimal values for the thresholds in case of 15 packets per second (7.5Kbps) packet transmission rate. It can be seen that throughputs above 80% of the maximum throughput are located when the contending number of mini-flows is between 15 and 35 (labelled as the optimally utilized region). Corresponding to 15 and 35 mini-flows, the contention times are 0.1 and 0.01 seconds respectively. Therefore, to utilize the capacity of a mini-channel optimally, the contention time has to be maintained between 0.1 and 0.01 seconds. Through observations made from extensive simulation results for different packet transmission rates (see Figure 5 (b)), we empirically choose 0.01 and 0.1 seconds, as the lower threshold α and upper threshold β, respectively.

3.5.2.2 Adapting Transmission Power

After measuring the contention time, each node can identify the current status of utilization within its mini-channel. Upon inferring over-utilization, a node increases the transmission power to enlarge the area of its mini-channel, which in turn will reduce the number of mini-flows. On the other hand, in case of under-utilization, a node will decrease the transmission power, which will increase the number of mini-flows. The relationship between transmission power and the number of mini-flows is shown at Figure 6. The number of mini-flows per mini-channel is calculated by dividing the total

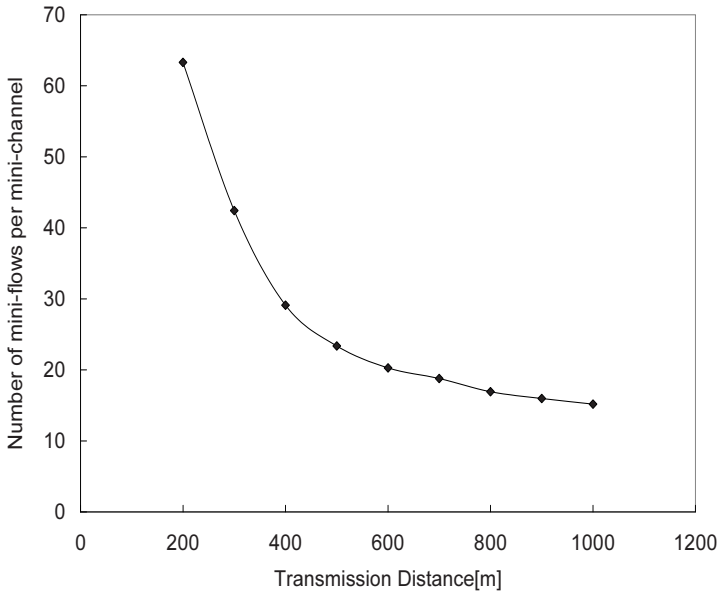


Figure 6: Relationship between the Number of Mini-flows Per Mini-channel and Transmission Distance: 15 flows(each flow has a rate of 5 packets per second) are used and 100 nodes are located in 1000m$\times$1000m area.

number of mini-flows by the spatial reuse factor.

The reasoning behind increasing transmission range to decrease the number of mini-flows is as follows: When the transmission range for a node in the network is increased, the number of mini-flows *in the network* will either remain the same or will decrease. If the number of mini-flows remains the same, the contention level will remain the same causing the concerned node to increase its transmission range further. On the other hand, a decrease in the number of mini-flows will reduce the overall contention level in the network. The reason for this phenomenon also leading to a *decrease in the contention level within the mini-channel* is the limited size of the typical sensor networks we consider. With an average diameter of 4-5 hops even when the network is minimally connected, any decrease in hop-count for the flow typically results in a decrease in the contention level for all the mini-channels that it traverses. A similar (but inverse) argument holds for decreasing transmission power in case of under-utilization.

Note that the increase decision is motivated by the desire for increasing throughput by going from the over-utilized region to the optimally utilized region. On the other hand, the decrease

decision is motivated by the desire for decreasing energy consumption by going from the under-utilized region to the optimally utilized region.

3.6 ATC Protocol

In this section, we first present a building-block algorithm for adaptively adjusting the transmission power based on load conditions of local areas.

At the first time, each node will choose the minimum transmission power, which connects the network minimally, as an initial power. Several works [64, 95, 97] have shown how to find the minimum power that guarantees connectivity.

After selecting the initial power, each node repeats a basic algorithm periodically. The basic algorithm starts and continues to measure the contention time during the period. And then it determines a transmission power. Finally, it synchronizes the transmission power among others based on different schemes, such as ATC-CP, ATC-IP, and ATC-MS, which we explain subsequently.

Based on the methodology of synchronization, the proposed scheme can be divided into three adaptive topology control schemes: (i) ATC-CP, adaptive topology control scheme using common power, (ii) ATC-IP, adaptive topology control scheme using independent power, and (iii) ATC-MS, adaptive topology control scheme using a master-slave coordination. Briefly, ATC-CP is targeted for networks where the underlying routing and MAC protocols assume that all nodes in the network use the same transmission range. On the other hand, ATC-IP allows nodes in the network to independently adjust the transmission power. Finally, in ATC-MS, nodes independently adjust transmission power, except for coordination between one-hop neighbors.

3.6.1 Basic Algorithm

All three adaptive schemes use the same basic algorithm till they determine the optimal transmission power locally. However, the schemes differ in the degree of coordination between the network nodes once they have determined the optimal local transmission power. Figure 7 shows the basic scheme which is used by all three schemes. For each period T, three phases, measurement, decision, and synchronization phase, are executed. At the beginning of the period, each node advertises the

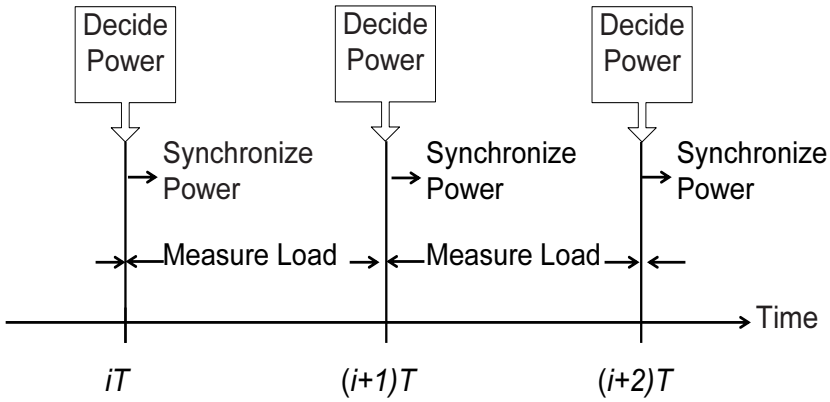


Figure 7: Procedure for the Basic Scheme at Each Period *iT*

power and status that were decided during the previous period. At the same time, each node starts to measure the traffic load using the contention time. At the end of the period, each node decides the power that has to be used during the next period.

3.6.1.1 Measurement Phase

Each node observes the contention time over a period of T seconds, referred to as an *epoch*. While a smaller T will enable quicker power adaptation to changes in the environment, the trade-off is the stability of the algorithm. We empirically set T to one second.

3.6.1.2 Decision Phase

After each node measures the contention time, it compares the measured contention time with two thresholds, α and β (described in Figure 5).

- Increasing Transmission Power: If a node detects the measured contention time to be above the upper threshold β, it increases the transmission power by the amount of Δ (Δ must be decided based on the transmission hardware specification. In this work, we use Δ amount of power to increase 100 meter transmission distance.) in order to decrease the number of contending mini-flows in the surrounding mini-channel. The corresponding improvement in utilization would in turn improve the throughput performance.

- Maintaining Transmission Power: If a node observes the measured contention time to be within the range between β and α, it maintains the transmission power in order to continue utilizing the capacity of the channel optimally.

- Decreasing Transmission Power: However, if a node observes the measured contention time to be below the lower threshold α, it should decrease the transmission power by Δ amount of power to increase the number of contending mini-flows which go through the node. Note that although the end-to-end throughput would remain the same after this phase, the energy consumption would reduce thus improving the performance of throughput per unit energy.

3.6.1.3 Synchronization Phase

In ATC-IP, each node starts using its locally computed optimal transmission power, independent of other network nodes. Hence, ATC-IP does not need to propagate and synchronize the power with the other nodes. However, in both ATC-CP and the ATC-MS, nodes synchronize with other nodes in the network (all nodes and one-hop neighbors respectively). In the rest of the section, we elaborate on the details of the three schemes.

3.6.2 ATC-CP

Most existing protocols for sensor networks assume the existence of symmetric links, with a key example being the IEEE 802.11 MAC protocol that requires bi-directional links. *We propose ATC-CP for such networks where link symmetry is critical.* In ATC-CP, all nodes in the network use the same transmission power. After each node independently uses the basic algorithm outlined earlier to determine the optimal transmission power, only the largest transmission power (among all nodes in the network) is chosen. The selection of the largest transmission power is achieved through a network wide flood. Note that although ATC-CP might seem highly inefficient, in practice (and as seen in Section 3.8) the algorithm performs quite well when compared to static topology control schemes. This can be attributed to the limited spatial-reuse in the network, which in turn leads to a high-correlation between the loads in the different mini-channels in the network.

3.6.2.1 Transmission Power Advertisement

After observing the contention time, and deciding the power with the basic algorithm, each node advertises its transmission power to the other nodes by flooding an advertisement for the transmission power. The flood forward mechanism at intermediate nodes is set up so that messages which carry a smaller advertised value than that of earlier forwarded messages within the same epoch, are suppressed to decrease the overhead of flooding. Note that while network floods performed using series of local broadcasts can induce the broadcast storm problem [54], other mechanisms can be used to alleviate some of the overheads.

3.6.2.2 Route Re-computation

Once all nodes are informed of the largest transmission power, the next step is to re-compute new routes for the flows because of the altered topology. In case of an increase in transmission power, some flows can experience path shortening. Most multi-hop routing protocols have automatic route shortening mechanisms. For example, the dynamic source routing (DSR) protocol is equipped with such a *route optimization* mechanism that will detect shorter routes and update the concerned source accordingly. However, in case of a decrease in power, some flows can suffer link failures. While mechanisms could conceivably be developed to perform *route lengthening* in an optimal fashion, we rely on the routing protocol's route re-computation process to recover from such failures. The performance benefits shown in Section 3.8 are notwithstanding the overheads due to such route failures.

Since most existing routing and MAC protocols assume bi-directional and symmetric links, the ATC-CP is preferable for adoption in networks with such protocols. However, it has two associated overheads due to: (i) advertisement of common power to all nodes and (ii) use of the worst case transmission range throughout the network.

3.6.3 ATC-IP

Because of the inherent overheads involved in global coordination, it is desirable for the ideal power control scheme to support distributed coordination among nodes. In ATC-IP, nodes use the locally computed optimal transmission power independent of the decisions made at the other network

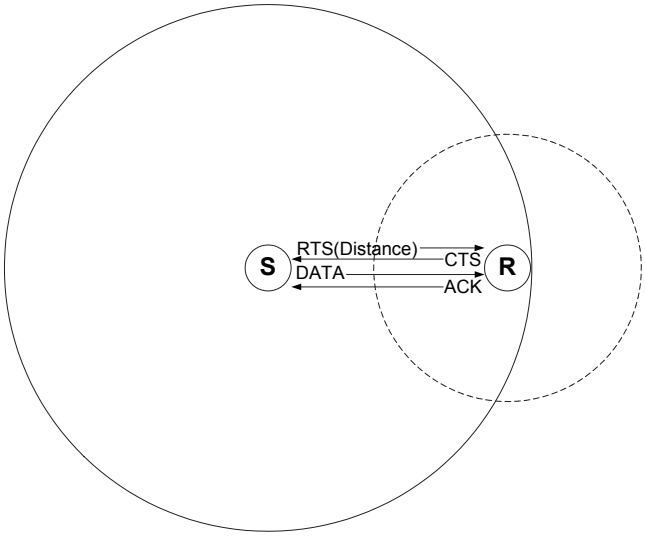


Figure 8: Illustration of Support for an Asymmetric Link at MAC Layer

nodes. However, because two neighboring nodes may use different transmission powers, it is highly probable that some links in the network become asymmetric. While several recently proposed protocols tackle the presence of asymmetric links at the routing layer [8, 45], the possibility of widespread proliferation of asymmetric links will also necessitate changes at the MAC layer (recall that the IEEE 802.11 MAC protocol requires bi-directionality). We pause to outline a simple means to adapt the IEEE 802.11 MAC protocol for asymmetric links in the context of ATC-IP.

3.6.3.1 Extending IEEE 802.11 for Asymmetric Links

In the conventional IEEE 802.11 MAC, a sender transmits RTS and DATA messages to a receiver, and the receiver responds with the CTS and ACK messages to the sender. Because the MAC layer relies on bi-directionality, asymmetric links will induce link failures. If the receiver, however, uses the power notified by the sender (say piggybacked on the RTS packet) to transmit CTS and ACK packets, asymmetric links can be supported successfully (see Figure 8). While this will increase the header overhead by about one byte, it is a negligible increase[4].

[4]The default sizes of RTS, CTS, DATA, and ACK packets in ns2 are 35 bytes, 28 bytes, 512 bytes, and 28 bytes, respectively.

3.6.3.2 Asymmetric Routes and DSR

Although sophisticated routing schemes could potentially be used to use asymmetric routes, such mechanisms are outside the scope of this thesis. For our simulations, we restrict the route selection process to choose only symmetric routes by sending back DSR route-replies along the route the route-request traversed through.

3.6.4 ATC-MS

Since mobile nodes within a mini-channel will experience the same level of contention, most nodes will eventually converge to the same transmission power after some delay. Hence, it is possible to decrease the convergence time by forcing neighbors within a one hop distance to use the same power. *Note that a one-hop coordination is sufficient since a mini-channel (as defined by the IEEE 802.11 MAC protocol) is a two-hop region.* Hence, when the node in the center of the mini-channel coordinates with all its neighbors, it synchronizes with all nodes within a mini-channel. We refer to the node at the center of a mini-channel, that has the highest locally computed transmission power, as the *master* and its neighbors as *slaves*.

To support the ATC-MS scheme, we change the MAC layer to enable the master to broadcast the current transmission power to one-hop neighbors. Specifically, we piggyback two fields: (1) the transmission power (2) the status of the node (master or slave)[5]. This scheme also requires the MAC changes discussed under ATC-IP for supporting asymmetric links.

3.6.4.1 Master/Slave Election

At the end of every epoch, a node estimates the power which is optimal for the locally observed contention and compares it with the maximum power which was last advertised by its master node. If the estimated power at the node is larger, the node becomes a master. Otherwise, the node becomes a slave.

[5]Note that while we chose to adapt the MAC layer headers to piggyback the information, the information can be propagated through other means also.

During Epoch:

1: // Whenever a node receives a RTS packet from master node
2: if $Advertised\_Power > Max\_Advertised\_Power$
3: // Update maximum power
4: $Max\_Advertised\_Power = Advertised\_Power$

5: // Whenever a node transmits a RTS packet
6: if $Status == MASTER$
7: // advertise power by piggybacking at a RTS packet
8: $flag\_master = true$

At The End of Epoch:

9: // Estimate optimal power using the measured contention time
10: $Estimated\_Power = Estimate\_Optimal\_Power(C_t)$
11: // Compare the estimated power with maximum advertised power
12: if $Estimated\_Power > Max\_Advertised\_Power$
13: // Become a master
14: $Status = MASTER$
15: else
16: // Become a slave
17: $Status = SLAVE$

Figure 9: Power Adaptation of ATC-MS during and at the End of Epoch

3.6.4.2 Transmission Power Synchronization

During each epoch, a node measures the contention time and stores the maximum power which was advertised by a master node. When the node transmits a RTS, it piggybacks its status (master or slave) and the transmission power. Figure 9 outlines ATC-MS scheme described thus far.

3.7 Convergence Analysis of ATC Schemes

Although we use adaptive schemes to change the transmission range *r* (or power) from minimum power to ensure connectivity of the network to the maximum power for each node to use practically, we can theoretically prove that the range *r* does not diverge and show the characteristic of a stationary distribution.

3.7.1 Problem Formulation

Suppose that we have an irreducible, aperiodic, and discrete-time Markov chain representing the process X_n which adjusts the transmission power from the minimum power connecting the network minimally to the infinite power connecting the network maximally. Each state of the process X_n represents the transmission power of which a node added $n \times \Delta$ to the minimal transmission power[6]. The corresponding transition probability matrix is denoted

$$\mathbf{P} = \begin{pmatrix} P_{00} & P_{01} & 0 & \cdots & \cdots & \cdots & \cdots \\ P_{10} & P_{11} & P_{12} & 0 & \cdots & \cdots & \cdots \\ 0 & P_{21} & P_{22} & P_{23} & 0 & \cdots & \cdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \cdots \\ \vdots & \vdots & 0 & P_{(n+1)n} & P_{nn} & P_{n(n+1)} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{9}$$

$P_{i(i+1)}$ and $P_{i(i-1)}$ mean the transition probability from i state to $i+1$ state and from i to $i-1$, respectively. And the transition is decided by comparing the measured contention time CT_i of state i with the threshold β and α as like:

$$P_{i(i+1)} = P[CT_i > \beta] \tag{10}$$

and

$$P_{i(i-1)} = P[CT_i < \alpha]. \tag{11}$$

3.7.2 Analysis

In order to show the stability of the proposed adaptive scheme, we use the following lemma:

Pakes's Stability Lemma[6]:

At first, to show the stability, define the drift D_i as

[6] Δ will be dependent on a hardware specification.

$$D_i = E\{X_{n+1} - X_n \mid X_n = i\} = \sum_{k=-i}^{\infty} kP_{i(i+k)}, i = 0,1,\ldots. \tag{12}$$

Suppose that $D_i < \infty$ for all i, and that for some scalar $\delta > 0$ and integer $\bar{i} \geq 0$, we have

$$D_i \leq -\delta, for \forall i > \bar{i}, \tag{13}$$

then the Markov chain X_n has a stationary distribution.

Intuitively speaking, if the sign of D_i is larger than 0, the state tends to increase. Therefore, the chain will be stable if the drift is negative for all large enough states(proof is shown at [6]).

In our scheme, drift can be defined as

$$D_i = (-1)P_{i(i-1)} + (0)P_{ii} + (1)P_{i(i+1)} = -P_{i(i-1)} + P_{i(i+1)}, i = 0,1,\ldots \tag{14}$$

Due to the inverse proportional relationship between the transmission range (which is proportional to the state index i) and the contention time CT_i [7], we can assume that

$$CT_0 > CT_1 > \cdots > CT_i > CT_{i+1} > \cdots. \tag{15}$$

If the measured contention time CT_i is a non-decreasing function of a transmission range, the transition probabilities $P_{i(i+1)}$ and $P_{i(i-1)}$ show the following characteristic

$$P[CT_0 > \beta] \geq P[CT_1 > \beta] \geq \cdots \geq P[CT_i > \beta] \geq P[CT_{i+1} > \beta] \geq \cdots \tag{16}$$

and

$$P[CT_0 < \alpha] \leq P[CT_1 < \alpha] \leq \cdots \leq P[CT_i < \alpha] \leq P[CT_{i+1} < \alpha] \leq \cdots. \tag{17}$$

If we choose threshold α such that

$$P[CT_0 < \alpha] > \frac{1}{2}, \tag{18}$$

[7] Since Figure 5 shows the proportional relationship between contention time and a number of mini-flows, and Figure 6 shows the inverse-proportional relationship between a number of mini-flows and a transmission range, the inverse-proportional relationship between a contention time and a transmission range can be derived syllogistically.

then we can also see that

$$P[CT_0 > \beta] < \frac{1}{2}, \tag{19}$$

since $P[CT_0 > \beta] = 1 - P[CT_0 < \alpha] < \frac{1}{2}$.

By replacing $P_{i(i-1)}$ and $P_{i(i+1)}$ in (14) with those of (10) and (11), the drift D_i shows

$$D_i = -P_{i(i-1)} + P_{i(i+1)} = -P[CT_i < \alpha] + P[CT_i > \beta], for \forall i > 0. \tag{20}$$

From (18) and (19), D_i cannot be larger than or equal to 0 as like

$$D_i \leq -\delta, for \forall i > 0, \delta > 0. \tag{21}$$

Therefore, based on the Stability lemma 1, the Markov chain X_n has a stationary distribution. Finally, the stationary distribution of the Markov chain X_n proves the convergence of the power although the infinity of power is assumed.

3.8 Performance Evaluation

In this work, three schemes of adaptive topology control have been proposed. To compare the performance of the adaptive schemes with those of the static schemes, extensive simulations are performed under realistic environments.

3.8.1 Evaluation Environments

The *NS2* simulator is used for all evaluations. The environment is largely similar to that used in Section 3.2 - (a) all flows are User Datagram Protocol (UDP) flows, (b) Dynamic Source Routing (DSR) is used at the routing layer, and (c) results from nine different sample runs lasting for 900 seconds are used for each data point. In order to acquire results for realistic environments, four kinds of environments (moderate node density - 100 nodes in a 1000m$\times$1000m area, high node

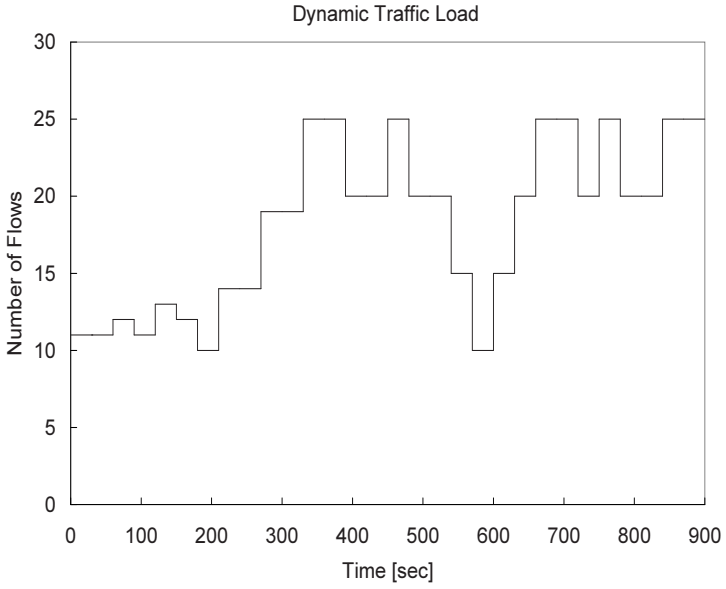


Figure 10: Illustration of Dynamic Traffic Load in Simulations

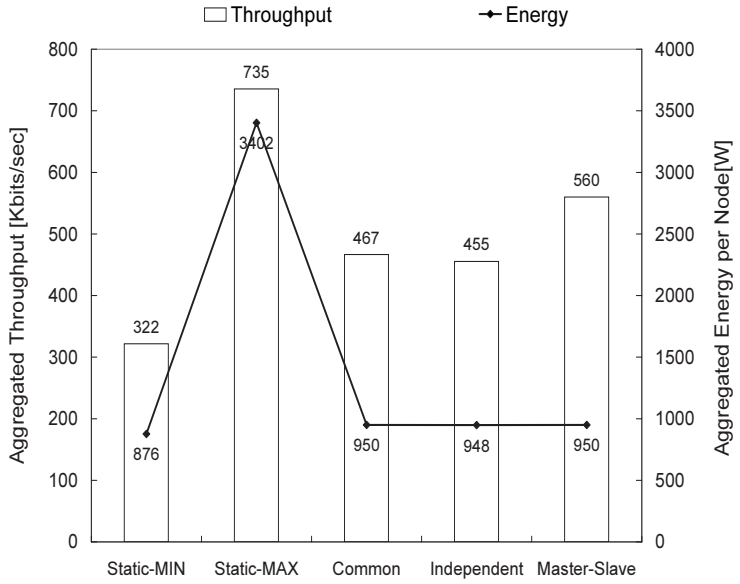
density - 400 nodes in a 1000m×1000m area, moderately-loaded mobile networks, and heavily-loaded mobile network) are used. Figure 10 illustrates the dynamic traffic load (ranging from 10 flows to 25 flows during the simulation time) used.

In each figure, Static-MIN stands for static topology control using the minimum transmission power required to keep the network connected. Static-MAX stands for static topology control using the minimum transmission power required to keep the network *fully connected*. Common, Independent, and Master-Slave stand for ATC-CP, ATC-IP, and ATC-MS, respectively.

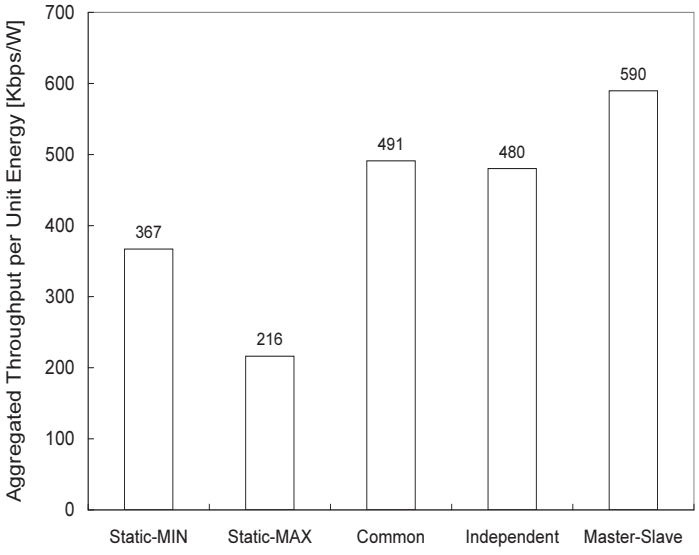
3.8.2 Evaluation Results

3.8.2.1 Moderate Node Density Network

In this scenario, 100 stationary nodes are located randomly in a 1000m×1000m area. Figures 11 (a) and (b) show the aggregate throughput and the energy consumption; and throughput per unit energy, respectively. Although static control using maximum power achieves the best throughput, it also consumes the largest amount of energy. Therefore, it has the smallest throughput per unit energy.



(a) Throughput and Energy Consumption



(b) Throughput per unit Energy

Figure 11: Simulation Results for Different Topology Control Schemes: 100 Stationary Nodes in 1000m$\times$1000m Area

Among adaptive schemes, ATC-MS outperforms ATC-CP and ATC-IP in terms of both throughput and throughput per unit energy. Since ATC-MS uses coordinated power control (unlike ATC-IP), it decreases the possibility of route failures, and also decreases the convergence time of the transmission power adaptation in the highly loaded portions of the network.

It is observed that the energy consumptions among the adaptive schemes are similar. We can attribute this to the major portion of the energy consumption spent on listening (which is a little less than that for transmitting for distances less than 300m). This is due to the fact that most of the nodes in the scenario listen. Hence, the difference in energy consumption is small, regardless of the control scheme.

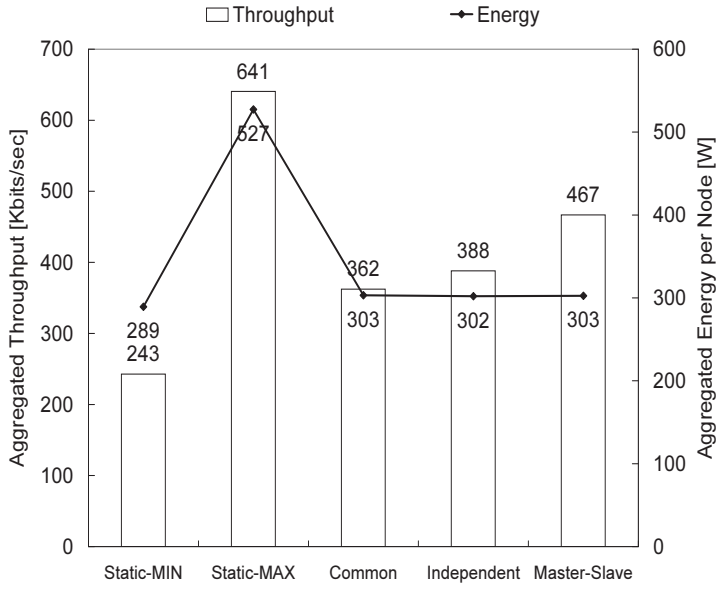
3.8.2.2 High Node Density Network

To observe the performance under high node densities, we use scenarios in which 400 stationary nodes are located randomly in a 1000m$\times$1000m area. In Figure 12, although the absolute differences among the different control schemes are small, ATC-MS outperforms the others in terms of throughput per unit energy. Comparing the results with that of the 100 stationary nodes case, it is seen that the throughput per unit energy of the 400-node case is less than that of the 100-nodes case. This is due to the fact that a larger fraction of the nodes in the 400-node scenario remain in the listen state (since the number of flows remains the same).

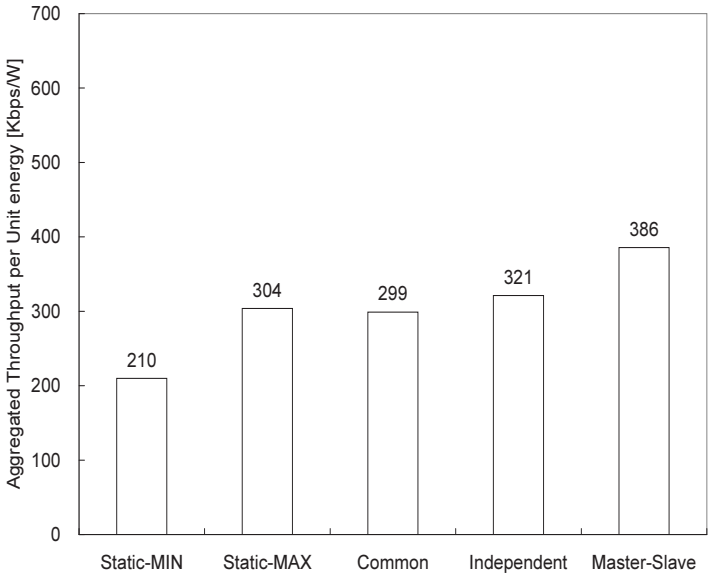
3.8.2.3 Mobile Network

Since a fundamental characteristic of sensor networks is mobility, we also observe the improvement in performance for a scenario with 100 mobile nodes with maximum speeds ranging up to 20 meter per second. The network area is 1000m$\times$1000m.

Figures 13 (a) and (b) show that, in the presence of mobility, ATC-MS again achieves the best performance over the other schemes. Moreover, when compared to the stationary nodes case in Figure 11, ATC-MS does better with mobility. Also, the static control scheme with minimum power does worse with mobility.

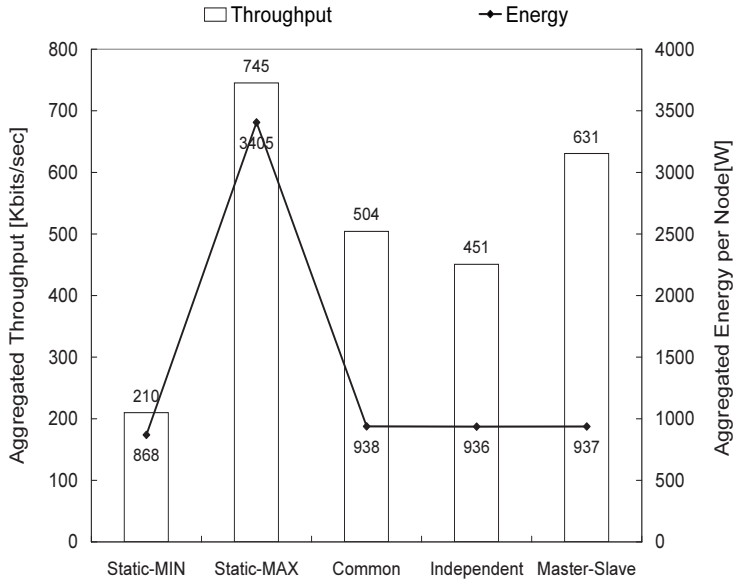


(a) Throughput and Energy consumption

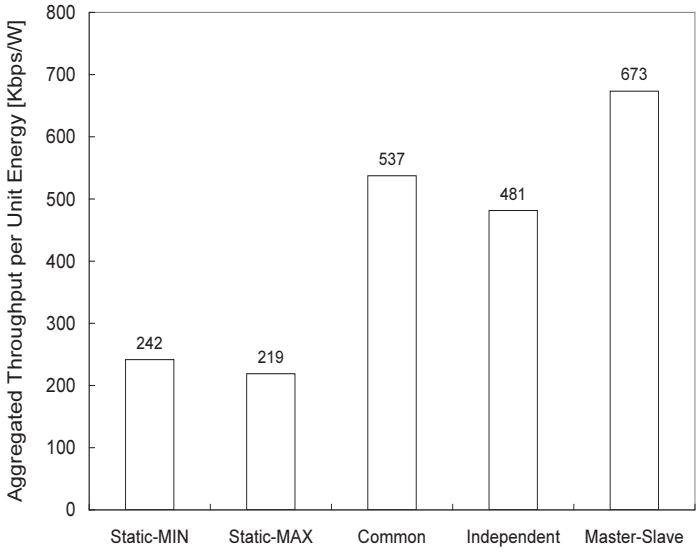


(b) Throughput per unit Energy

Figure 12: Simulation Results for Different Topology Control Schemes: 400 Stationary Nodes in 1000m$\times$1000m Area

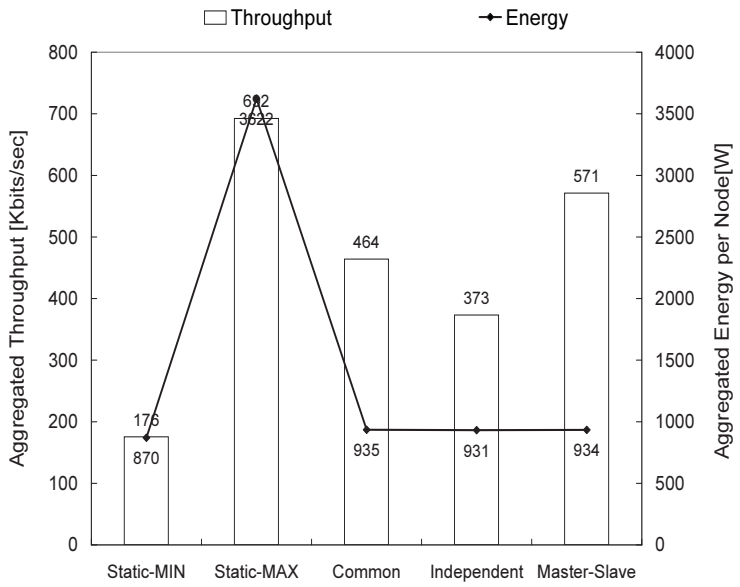


(a) Throughput and Energy consumption

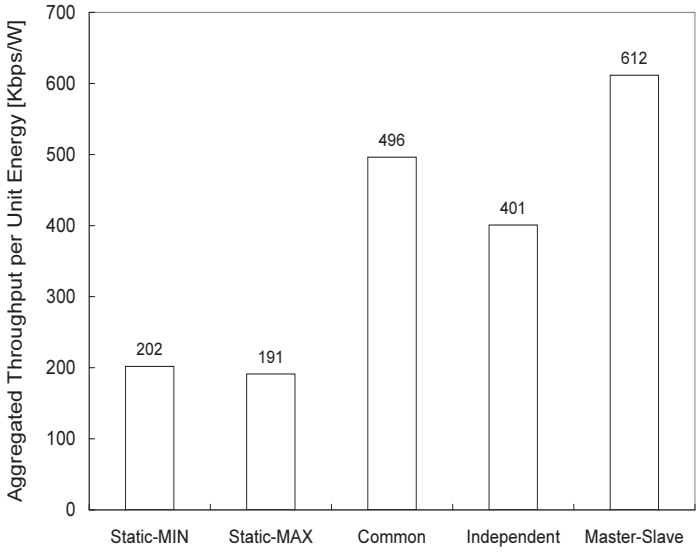


(b) Throughput per unit Energy

Figure 13: Simulation results for Different Topology Control Schemes: 100 mobile nodes located in 1000m$\times$1000m area.



(a) Throughput and Energy consumption



(b) Throughput per unit Energy

Figure 14: Simulation results for Different Topology Control Schemes: 100 mobile nodes in 1000m$\times$1000m area and 35 flows.

3.8.2.4 Heavily Loaded Mobile Network

To evaluate performance under a combination of heavily-loaded traffic and a mobile environment, we use more number of flows (ranging from 20 to 35) with mobile nodes having maximum speeds of up to 20 m/s. Figure 14 shows that ATC-MS still outperforms the others. Due to the heavy load, the throughput of each scheme is less than that of the moderately loaded mobile network in Figure 13. However, ATC-MS under mobility achieves more throughput per unit energy than under a stationary environment.

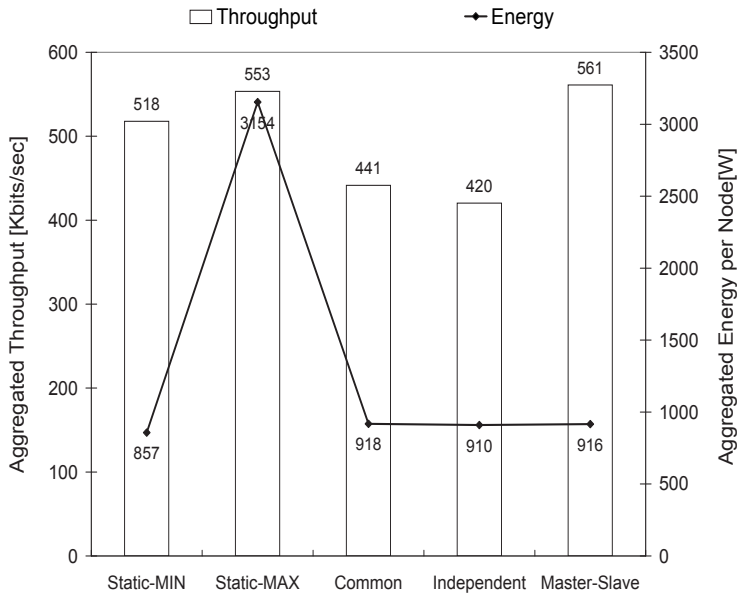
3.8.2.5 Impact of TCP

To evaluate performance of Transmission Control Protocol (TCP) with adaptive topology control schemes, we vary the numbers of TCP flows (ranging from 10 to 30) with 100 mobile nodes having maximum speeds of up to 20 m/s. Each flow uses Constant Bit Rate (CBR) traffic with data rate of 7.5Kbps. Figure 15 (a) shows that the throughput of Static-MAX case almost reaches that of ATC-MS. However, in Figure 15 (b), the throughput per unit energy of ATC-MS still outperforms those of the others.

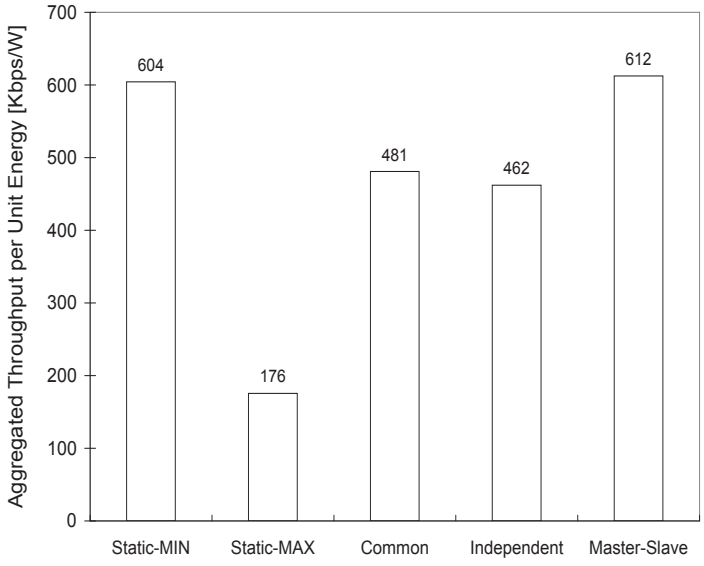
Although TCP flows with Static-MIN scheme achieves similar throughput performance, the normalized standard deviations of TCP flows using Static-MIN scheme are larger than those of ATC-MS scheme(see Figure 16). Because smaller deviation means better fairness among TCP flows, it is better to keep smaller standard deviation among TCP flows. Therefore ATP-MS scheme provides a higher degree of fairness.

3.8.2.6 Convergence of ATC-IP and ATC-MS

In this section, we show the convergence properties of the proposed distributed algorithms for scenarios with dynamic traffic loads. Figure 17 shows the instantaneous change in transmission power with changes in the number of flows. To study the convergence of the two adaptive (and distributed) topology control schemes, we show a snapshot of the transmission power changes at all nodes in the network during two time intervals: from 70 to 140 seconds, and from 200 to 270 seconds. As seen in Figure 10, the number of flows increases from 10 to 13 flows at 90 seconds, and the number of flows increases to 15 flows at 210 seconds. From Figures 17(a) and (b), we can observe the convergence



(a) Throughput and Energy consumption



(b) Throughput per unit Energy

Figure 15: Simulation results for Different Topology Control Schemes: 100 mobile nodes in 1000m$\times$1000m area and 30 flows which use CBR(7.5Kbps) over TCP protocol stacks.

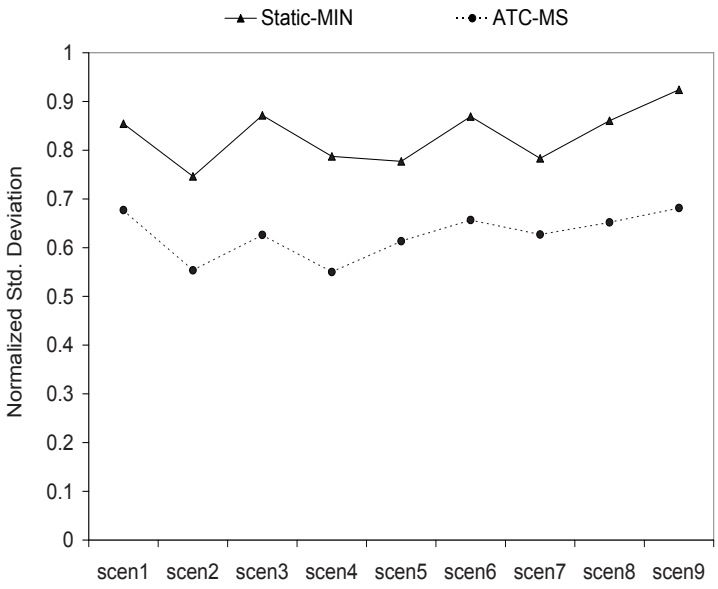


Figure 16: Normalized Standard Deviation of Each Scenario between Static-MIN and ATC-MS Schemes

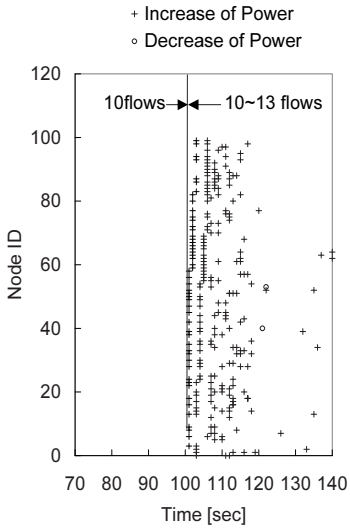
of the transmission power in response to the change in the traffic load. Furthermore, Figures 17(c) and (d) show that the convergence time of ATC-MS is shorter than that of ATC-IP. The quicker convergence time can decrease the transient time involved in reaching an optimal transmission power, contributing to the better performance of ATC-MS.

In Chapter 3.7, we already proved non-divergence of the transmission range and showed the characteristic of a stationary distribution by stability lemma in [6]. The non-divergence of the transmission range is proved by showing that the tendency to drift is negative for all large enough states.

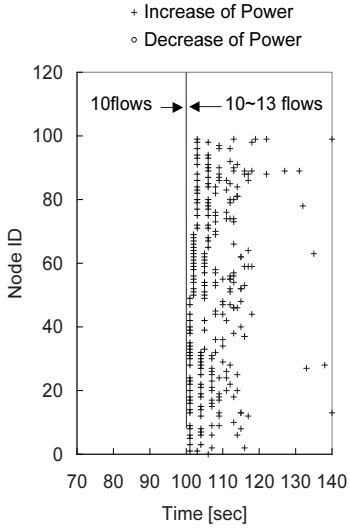
3.8.2.7 Overheads of Adaptive Topology Control

In this section we study the overheads caused due to the adaptive nature of the proposed topology control schemes. Specifically, frequent changes in the transmission power used by nodes can result in link failures that in turn lead to route failures and route re-computations.

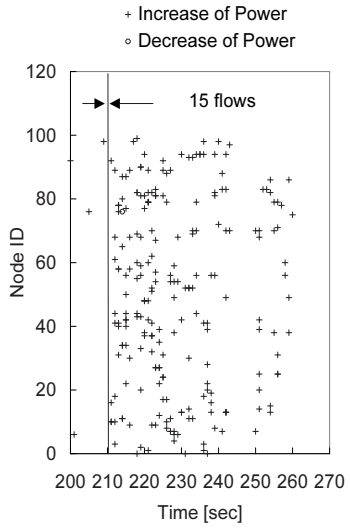
Figure 18 shows the number of route errors for static topology control schemes and adaptive topology control schemes when the maximum speed for each mobile station is 20 m/s. In Figure 18, the gray bars show the results for the mobile network scenarios, and the checkered bars show the



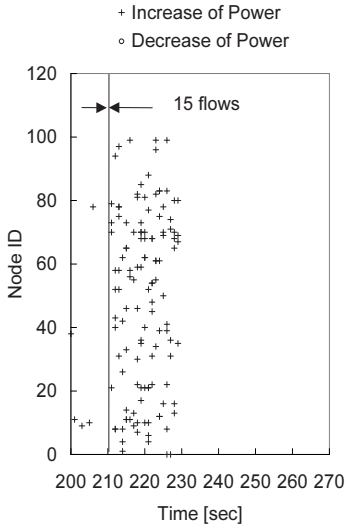
(a) ATC-IP from 70 to 140 second



(b) ATC-MS from 70 to 140 second



(c) ATC-IP from 200 to 270 second



(d) ATC-MS from 200 to 270 second

Figure 17: Illustration of Convergence for ATC-IP and ATC-MS: (a) and (b) Snapshot from 70 Seconds to 140 Seconds in Figure 10 - 100 Stationary Nodes Located in 1000m×1000m Area, (c) and (d) Snapshot from 200 Seconds to 270 Seconds 10 - 100 Stationary Nodes Located in 1000m×1000m Area.

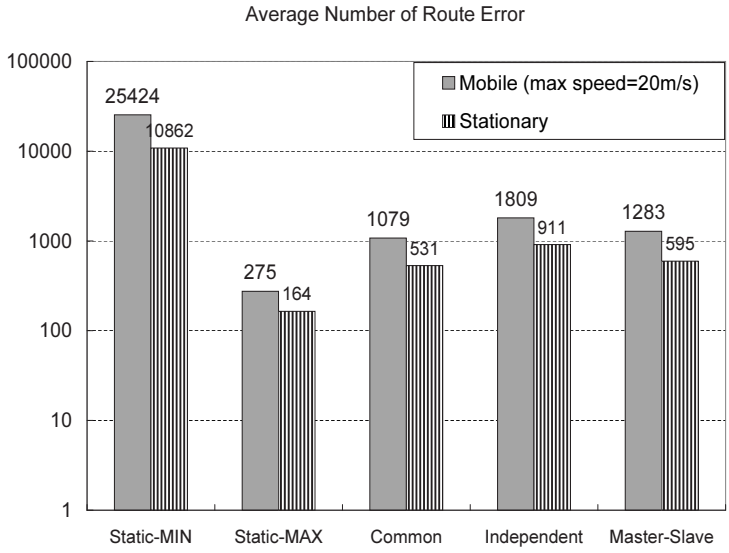


Figure 18: Number of Routing Errors for Adaptive and Static Topology Control Schemes

results for a stationary network. Both network scenarios have 100 nodes in a 1000m$\times$1000m area. Despite the adaptive topology control, the number of route errors for the adaptive schemes is fewer than five percentage of the static control scheme using MIN power. The static control using MAX power has the smallest number of route errors since all nodes are connected fully at all times.

Figure 18 shows that the number of error messages for static-MIN outnumbers that of the others. This can be explained as follows: Route errors can be triggered even due to high contention related packet losses (This is due to the fact that the MAC layer cannot distinguish a neighbor having moved out of range from the neighbor not responding due to high contention.). This problem is exacerbated in a static control scheme since the topology is not adapted when the traffic load is high. In case of adaptive schemes, since the adaptive change of topology decreases the contention level, the number of route errors is significantly fewer.

In terms of byte overheads, the fields that need to be added to MAC headers (the RTS header) include (1) transmission power that a node will use during the next period and (2) a flag that indicates the status of the node. We add only one byte each for the two fields to the packet of RTS whose size is now 40 bytes.

3.9 Summary

A topology control with a transmission power in wireless sensor networks is a relatively understudied problem. In general, the minimal transmission power is assumed to create the topology which produces the maximum performance, such as throughput or throughput per unit energy.

This work, however, shows that in contrast, the transmission power to create the optimal topology is a function of the load in the network. This argument is substantiated with simulation results in a variety of scenarios. From the observations, it can be motivated that a topology should be changed to the environments by a variable transmission power.

Finally, three adaptive topology control schemes are proposed: (i) the ATC-CP using common power among all nodes, (ii) the ATC-IP using independent power, and (iii) the ATC-MS using a master-slave coordination method. Since wireless sensor networks do not have any centralized method to synchronize the common power, the ATC-CP has an overhead to flood the common power at every period. On the other hand, the ATC-IP using independent power is a distributed scheme in which each autonomous node operates the topology control separately and independently. Since wireless sensor networks have mini-channels where nodes contend with each other and have similar contention time, the ATC-MS harmonizes transmission power of nodes within a mini-channel with a maximum power within the mini-channel.

Although the implementation of these three proposed schemes merely modifies the MAC layer, all the adaptive schemes outperform static topology control schemes. In particular, the ATC-MS shows the best performance under all environments.

From the evaluation of the proposed adaptive topology control schemes, it can be concluded that the topology should be adapted to the environment in order to produce the best performance in terms of the energy consumption and the throughput.

CHAPTER IV

GARUDA-DN: RELIABLE DOWNSTREAM DATA DELIVERY

4.1 Problem Definition

The increased awareness of the wide variety of applications for sensor networks has spurred a tremendous amount of research in the area over the past few years [4]. Because of the frugal energy budget of sensor networks, a significant amount of such work focuses on energy-aware network protocols [32, 36, 38].

In addition to the energy conservation problem, sensor networks suffer from a high rate of data loss due to wireless channel errors, congestion, and broadcast storm [54]. Under high rate of data losses, unreliable data deliveries increase the odds of data retransmission and hence waste a significant amount of valuable energy. Therefore, it is necessary to consider the robustness of protocols while taking into account energy conservation of the networks.

In this chapter, we consider the problem of *reliable downstream point-to-multipoint data delivery*, from the sink to the sensors, in wireless sensor networks (WSNs). The need (or lack thereof) for reliability in a sensor network is clearly dependent upon the specific application the sensor network is used for [92]. Consider a security application where the sensors are required to detect and identify the presence of given targets. Given the critical nature of the application, it can be argued that any message from the sink has to reach the sensors reliably. We elaborate on this further through the discussions below. We also use the discussions to identify the three classes of messages:

- If the underlying network is composed of reconfigurable sensors[1] [53], the sink may want to send a particular (say upgraded) image detection/processing software to the sensors. We refer to such messages as *control code* that a sink might want delivered to sensors. Obviously, it is highly undesirable that the control code, or parts of it, do not reach a subset of sensors.

- Next, the sink may have to send a database of target images to the sensors, to help in image

[1] Sensors that can operate in one of several modes of operation.

recognition triggered by subsequent queries. We refer to such data as the *query-data*. We separate such data from the query itself as the query-data can be expected to be much larger in size, and will be sent less often.

- Finally, the sink can send out one or more *queries* requesting information about the detection of a particular target. The sensors can then match targets detected with the pre-stored images, and respond accordingly. It is then highly desirable that the queries reach all the sensors reliably to prevent the case that the presence of a target might get unreported.

The problem of reliable data delivery in multi-hop wireless networks is by itself not new, and has been addressed by several existing works in the context of wireless ad-hoc networks [88]. However, such approaches do not directly apply to a sensor environment because of three unique challenges imposed by the following considerations: (i) *Environment considerations:* The constraints imposed by a WSN environment is substantially different from those imposed by other types of multi-hop wireless networks. A few examples include the limited lifetime of network nodes, the scarcity of the bandwidth and energy, and the size of the network itself. (ii) *Message considerations:* While most approaches for group reliable transport over multi-hop wireless networks in related work consider large sized messages (spanning several packets), most messages in a sensor network might be small sized *queries*. This raises fundamental issues on what kind of loss recovery schemes can be employed. (iii)*Reliability considerations:* The notion of reliability that is traditionally prevalent is that of a simple 100% reliable data delivery. However, WSNs might require other notions of reliability ranging from reliable delivery to only a sub-region of the network to partial probabilistic reliability for scoped-resolution based querying.

In this chapter, we address the above challenges and present an approach called GARUDA[2]-DN that provides reliable point-to-multipoint data delivery from the sink to the sensors. GARUDA-DN is scalable with respect to the network size, message characteristics, loss rate, and reliability semantics, and consists of the following elements as the cornerstones of its design: (i) an efficient pulsing based solution for reliable short-message delivery; (ii) a virtual infrastructure called the *core* that approximates a near optimal assignment of local designated servers, which itself is

[2]A mythological bird that reliably transported gods.

instantaneously constructed during the course of a *single* packet flood; (iii) a two-phase NACK based recovery process that effectively minimizes the overheads of the retransmission process, and performs out-of-sequence forwarding to leverage the significant spatial re-use possible in a WSN; and (iv) a simple candidacy based solution to effectively support the different notions of reliability that might be required in a WSN. We show through both macroscopic and microscopic results that GARUDA-DN shows great promise in terms of its performance.

4.2 Motivation

We first confine the robust data delivery problem to a simple and specific reliable delivery problem with several assumptions. We then show that the inherent redundancy in sensor networks cannot guarantee any strict reliability semantics due to a variety of reasons. We argue that robustness to losses is a necessary condition in order to conserve energy since unreliable data delivery can increase energy consumption.

4.2.1 Assumptions

In this work, we first address the issue of reliability in the following context:

- *Downstream Reliability:* Although a strong case can be made for mechanisms to ensure reliability in both the upstream and the downstream directions in mission critical applications, we restrict the scope of this work to downstream reliability.

- *Communication and Node failures:* A scheme that addresses reliability in a sensor network environment, has to deal with (i) Communication failures and (ii) Node failures. The proposed algorithm will handle *both* communication and node failures through it's design as we elaborate in Section 4.7.

- *100% reliable message delivery:* Reliability in sensor networks can have several dimensions as we mention in Section 4.4. At first, we focus on a basic framework that provides 100% reliability to all sensors. We then extend the basic framework to cover all semantics in Section 4.8.

- *Message size:* We assume that the message size to be sent by the sink consists of one or more packets. It is interesting to note that for one of the types of messages: the query, it is likely that the message size more often than not does not exceed one packet. At the same time, support for reliable delivery of one packet messages pose unique challenges as we discuss in Section 4.4.

- *Metrics:* We consider latency, retransmission overhead and energy consumption as the metrics of interest for comparison with other existing approaches. The goals in GARUDA-DN are hence to minimize these metrics.

- *Network Model:* We assume that both the sink and the sensors in the network remain *static*. We also assume that there is exactly one sink coordinating the sensors in the field. Further, since sensor networks have a large number of sensor nodes, the GARUDA-DN approach must be scalable to the number of nodes in the network.

4.2.2 Observations

Sensor networks are typically characterized by a high degree of redundancy. While the redundancy is motivated by the need to extend the lifetime without redeployment, it can be conjectured that the high degree of redundancy will also provide communication reliability. However, due to several reasons that we outline below, the redundancy by itself cannot provide any reliability guarantees, thus necessitating separate mechanisms for that purpose. While the factors compromising reliability are by no means new, we believe that this discussion serves the important purpose of highlighting the impact of the factors in the specific context of a sensor environment. The factors that contribute to the lack of reliability are:

4.2.2.1 Wireless Channel Errors

Wireless networks in general are highly influenced by random channel errors due to interference, and effects such as fading. *While the inherent redundancy of a sensor network can provide redundant paths for packet delivery to a single node, we contend that unreliability due to random wireless errors is still of concern.* Figure 19 presents the percentage of network nodes receiving a message

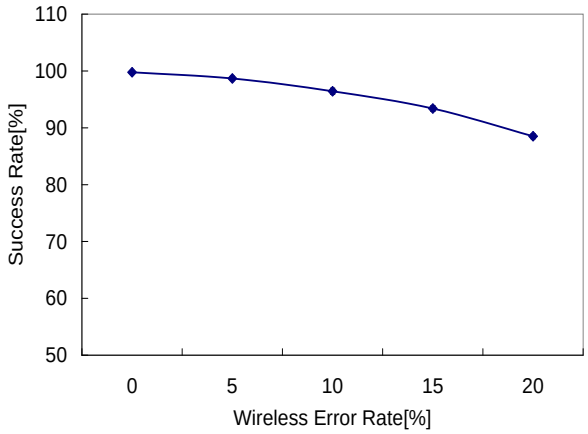


Figure 19: Delivery Ratio as a Function of Random Wireless Error Rate in Sensor Network: 95% Confidence Interval for the Mean over 20 Simulation Runs

reliably (defined as the success rate) with increasing random wireless channel error rate. The message size is set to 100 packets (packet size = 1KB), and the network is a 650x650m grid with 100 nodes. It can be observed that the success rate decreases from 100% to about 88% as the random channel error rate increases from 0 to 20%.

4.2.2.2 Congestion and Contention

The downstream and the upstream traffic will typically share the same channel, the capacity of which is limited. Hence, the downstream traffic reliability is clearly affected by the congestion caused by the upstream traffic. Figure 20 illustrates the effect of background traffic on the percentage of nodes receiving a message reliably for the same network topology and message size as mentioned in Section 4.2.2.1, but with no channel error. It can be observed that the success rate decreases from about 97% when the aggregate background traffic[3] is about 25 Kbps to about 76% when the aggregate background traffic is increased to 400 Kbps.

4.2.2.3 Broadcast Storm

Broadcast storm is the term associated with the set of problems that arise when flooding is performed in multi-hop wireless networks through a series of local broadcasts [54]. While the problem was initially identified in the context of ad-hoc networks, its impact in sensor networks is higher because

[3]Upstream flows from all 100 sensors to the sink using a CBR source.

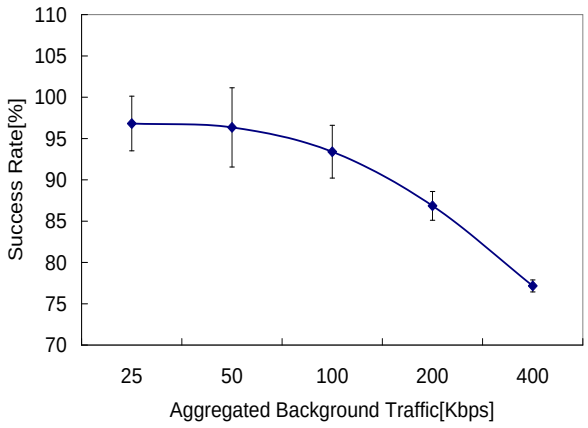


Figure 20: Delivery Ratio as a Function of Background Traffic Load in Sensor Network: 95% Confidence Interval for the Mean over 20 Simulation Runs

of the larger density of such networks. Hence, when a message from the sink is propagated as a series of local broadcasts, the problems categorized under the broadcast storm phenomenon, namely more collisions and higher degree of contention, result in several network nodes not receiving parts of the message. Figure 21 presents the percentage of nodes that receive a message reliably as the number of nodes in the network (for a fixed grid size of 650x650m) increases. The success rate drops from about 99% for the 100 node scenario to 83% for the 800 node scenario. (There is no background load or random wireless channel errors for the scenarios used to obtain the results presented in Figure 21).

Shown that the lack of reliability is a genuine problem in sensor networks, we need a reliable data delivery scheme in sensor networks.

While the above discussions substantiate our contention that the lack of reliability is a genuine problem in sensor networks, in the rest of the section we discuss some related work that pertain to providing reliability.

4.3 Related Works

To provide robust data delivery, researchers have proposed several approaches at the different protocol layers including: (i) physical/link layer approaches, such as Forward Error Correction (FEC)

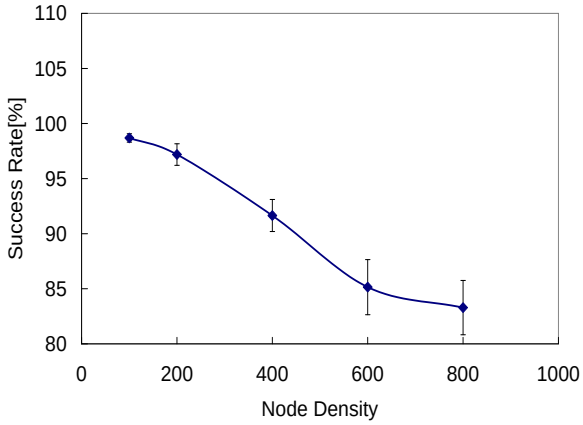


Figure 21: Delivery Ratio as a Function of Number of Nodes in Sensor Network: 95% Confidence Interval for the Mean over 20 Simulation Runs

[10, 49], (ii) MAC layer approaches, such as reliable MAC [88], and (iii) transport layer approaches, such as reliable multicast [24, 47] and reliable transport protocol [83, 92].

- [24, 46, 47, 69] are reliable multicast approaches specifically designed for wired or multi-hop wireless environments which assume an address-centric routing layer and global unique node identification. Since wireless sensor networks require a data-centric routing layer without global identification, such approaches cannot be applied directly to wireless sensor networks.

- FEC has been an appealing approach to prevent feedback implosion that can happen when performing a large scale reliable multicast [10]. However, [48] evaluates the utility of FEC for reliable multicast and compares the effectiveness of FEC with that of a subcasting[4] enabled multicast. [48] argues that FEC provides little benefit for an efficient reliable multicast protocol like [47] that uses subcasting. Since wireless sensor networks inherently support local subcasting because of the shared nature of the wireless channel, the gain of FEC in wireless sensor networks can be argued to be minimal. Currently the effect of FEC in wireless sensor networks is being evaluated.

[4]Subcasting is a functionality that involves multicasting of a retransmitted packet by a loss recovery server over the entire subtree rooted at the server. Hence, all instances of that lost packet within the subtree are recovered by the single subcast.

- Several works have been proposed to perform efficient flooding in multi-hop wireless networks [54, 58]. [98] classifies some of these approaches as probability-based, area-based, and neighbor-knowledge based schemes. While such approaches improve the successful delivery rate of messages, they still cannot guarantee any strict reliability semantics that GARUDA-DN supports. Such approaches in fact can be used in tandem with GARUDA-DN.

- In [25], the authors propose schemes to minimize the latency and the number of retransmissions involved in flooding using a scheduling scheme that constructs a broadcast tree and schedules transmissions with a greedy strategy. The approach is not targeted toward large scale networks, supports only the simplest form of reliability semantics, and does not leverage the unique characteristics of sensor environments.

- PSFQ [92] is a transport layer protocol that addresses the issue of reliability in sensor networks. The key idea in the design of PSFQ is to distribute the data from a source node by transmitting data at a relatively slow speed, but allowing nodes that experience losses to recover missing data packets from immediate neighbors aggressively. However, PSFQ does not provide any reliability for single packet messages as it uses a pure negative acknowledgement (NACK) based scheme. Also, it uses in-sequence forwarding for message delivery to accomplish the pump slowly operation. This results in the wastage of precious bandwidth as we elaborate in Section 4.4.

- In [71, 83], the authors propose reliable transport layer solutions to provide some level of reliability by controlling the reporting rate of sensors or by having multiple paths between sensors and a sink. They are concerned with upstream reliable delivery from sensors to sink.

4.4 Design Preliminaries and Challenges

The problem addressed in this work is that of reliable sink-to-sensors downstream data delivery in wireless sensor networks (WSNs). We restrict the focus of the work to WSNs with a single sink and static sensors, and assume that the lack of communication reliability can be due to various

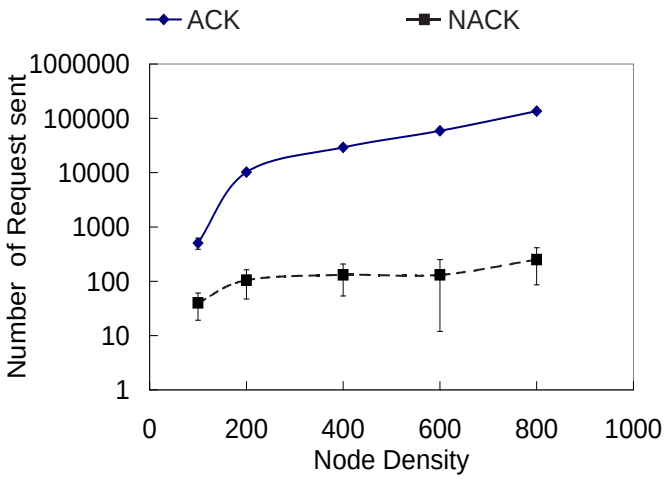


Figure 22: Comparison of ACK and NACK Schemes: under a Scenario that a Sink Broadcasts Reliably a Message to All Sensors

reasons such as random wireless errors, congestion, or other failures. The problem scope includes tackling the diverse reliability semantics that might be required in WSNs. The goal is thus to achieve reliability while minimizing bandwidth usage, energy consumption, and delay, with the solution not only addressing the unique characteristics of WSNs, but also leveraging them where appropriate.

4.4.1 Preliminary Design Choices

In this section, we first motivate the basic choices involved in the design of an approach for reliable sink-to-sensors delivery. Specifically, we organize our discussions in terms of the loss detection, loss recovery and the forwarding schemes. We then summarize the basic design choices and outline the key research challenges that need to be addressed.

4.4.1.1 Loss Detection Scheme

The request mechanism involves the approach used by sensors to initiate a retransmission request. While conventional transport protocols such as TCP use a positive acknowledgement mechanism (ACK), it is well established that in point to multi-point communication [24], using a negative acknowledgement (NACK) scheme is preferable in order to avoid the ACK implosion problem. Figure 22 illustrates the request transmission overheads (total number of request packets transmitted) to reliably deliver a message of 100 packets (of size 1KB) to a network of n nodes in a 650mx650m grid. n was varied from 200 to 800 in the figure. And the transmission range of each node was 65m.

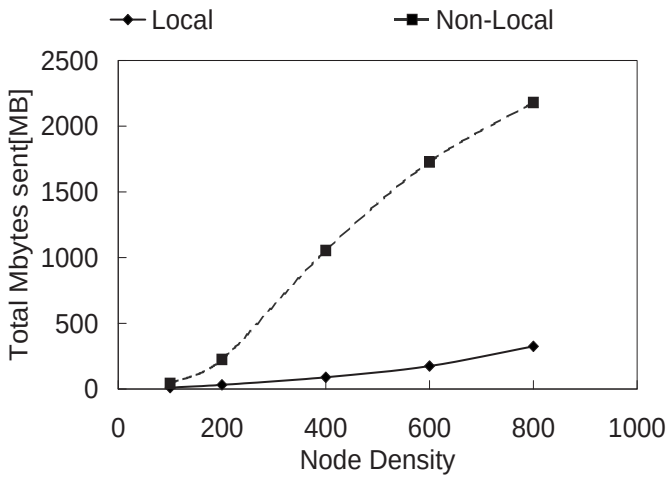


Figure 23: Comparison of Local and Non-Local Recovery Schemes: under a Scenario that a Sink Broadcasts Reliably a Message to All Sensors

The background load was fixed at an aggregate of 80Kbps, and the random wireless loss rate was set at 5%. It was observed that the total number of ACKs transmitted was an order of magnitude higher than the number of NACKs transmitted under the same conditions.

While a NACK based scheme is indeed preferable, it has one obvious disadvantage in the problem context considered in this thesis. When a message consists of only one packet, a NACK based scheme cannot suffice as receivers have no way of inferring that a lost packet was sent in the first place. This problem[5], more generically, is associated with conditions where a receiver does not receive *any* of the packets transmitted by the sender. Under such conditions, a NACK based scheme simply cannot provide the desired behavior. At the same time, in a sensor environment, it is very likely that one class of messages - the query, more often than not, consists of only a few packet transmissions. Thus, while a NACK based scheme should be used to avoid the ACK implosion problem, it needs to be supplemented with additional mechanisms to handle the case where none of the packets in a message are received at a node.

4.4.1.2 Loss Recovery Scheme

The recovery scheme determines which node can respond to a retransmission request. We refer to the nodes that are allowed to respond with a retransmission as *recovery servers*. Recovery servers

[5]We refer to this problem as the *all-packets-lost* problem later in the thesis.

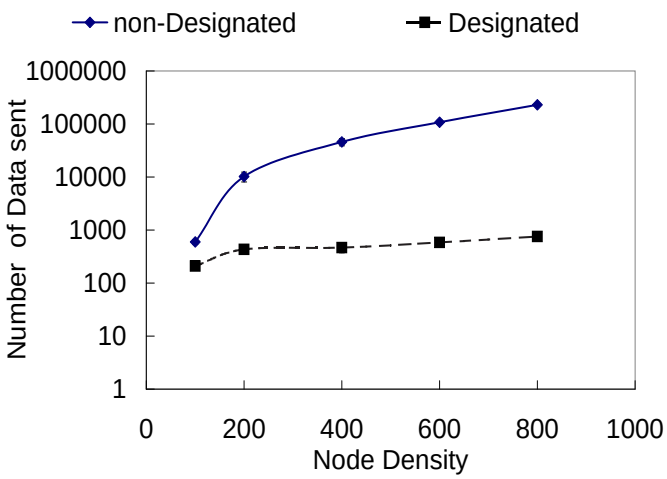


Figure 24: Comparison of Designated and Undesignated Recovery Server Schemes: under a Scenario that a Sink Broadcasts Reliably a Message to All Sensors

can either be local (local recovery), or non-local (non-local recovery). A typical example for local recovery is where any neighbor of the requesting node is allowed to respond. An extreme case for non-local recovery is where all retransmissions have to be performed only by the sink. Given the scale of a sensor network, it is evident that non-local recovery is not a scalable solution (see Figure 23).

Even given that local recovery is to be performed, another design decision that needs to be made is whether the local servers are explicitly designated (*designated server scheme*), or any sensors can act as recovery servers without designation (*undesignated server scheme*). The drawback of the undesignated scheme is that multiple neighbors can respond to the same retransmission request causing additional collisions and contention (see Figure 24), for the same network topology mentioned earlier in this section). Whereas, in the case of the designated scheme, if the designation is performed appropriately such that multiple recovery servers are not assigned within the same contention region, retransmissions from designated servers can be made to be free of collisions with retransmissions from other designated servers.

On the other hand, the problem of optimal designation both from the perspective of minimizing the number of retransmissions and in terms of updating the designations in accordance with the dynamics in the network needs to be addressed.

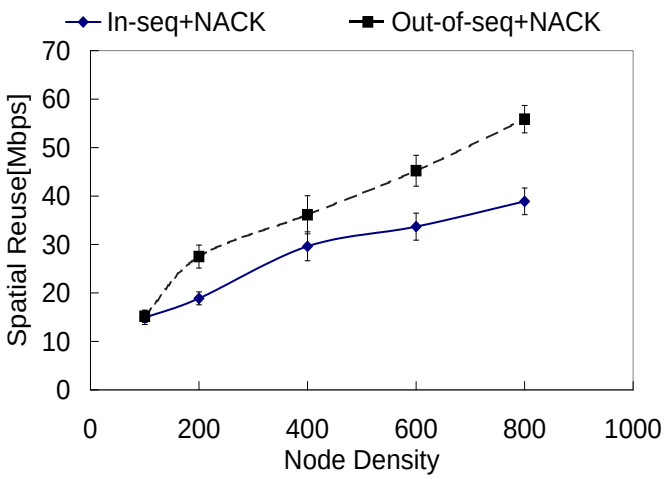


Figure 25: Comparison of In-sequence vs Out-of-sequence Forwarding Schemes: under a Scenario that a Sink Broadcasts Reliably a Message to All Sensors

4.4.1.3 Forwarding Scheme

The forwarding scheme refers to whether packets received out-of-sequence are forwarded by the underlying flooding mechanism. While the basic flooding scheme will forward packets irrespective of whether they are out-of-sequence or in-sequence, a case can be made not to forward out-of-sequence packets when a NACK based request mechanism is used [92]. Essentially, when a packet is lost and subsequent out-of-sequence packets are forwarded, all downstream nodes (from the point of loss) will detect the hole in the packet sequence and issue NACK requests despite the fact that upstream nodes do not have the missing packets. In [92], a case is thus made to forward packets only in-sequence to eliminate the transmissions of unnecessary NACKs.

However, a key drawback of the in-sequence forwarding strategy is that precious downstream network resources can be left under-utilized when forwarding is suppressed. A good measure of the utilization of a network is the *spatial re-use* achieved, where spatial re-use is defined as the total number of packets transmitted in the network in unit time. Figure 25 shows the degree of spatial re-use in the out-of-sequence and in-sequence schemes respectively for the same network topology mentioned earlier in this section. It was observed that the in-sequence scheme clearly under-utilized the network capacity.

Summary of Design Choices:

We summarize the basic design choices discussed thus far as follows:

1. A NACK based loss recovery scheme is preferable to an ACK based scheme as the latter suffers from the ACK implosion problem.

2. Local and dynamically assigned designated servers are essential to minimize the retransmission data overhead.

3. Out-of-sequence forwarding should be preferred to maximize the spatial reuse in the network.

4.4.2 Challenges

We now present the fundamental challenges that need to be addressed for providing effective downstream reliability in WSNs.

4.4.2.1 Environment Constraints:

There are two primary limitations in a WSN that need to be tackled to provide effective downstream data reliability: (i) bandwidth and energy constraints and (ii) frequent node failures.

The bandwidth and energy constraints may be tackled by minimizing the amount of *retransmission* overheads to ensure reliability. This in turn will reduce both bandwidth and energy consumption due to the reliability process. The proneness to node failures, on the other hand, should be tackled by not relying on statically constructed mechanisms (say, a broadcast tree) that do not account for the dynamics of the network. Note that "dynamic" mechanisms that periodically refresh any constructions are not desirable as the overheads due to the reliability process have to be minimized too.

Another characteristic of the target environment that needs to be accounted for is the scale of the network. WSNs can be expected to be of a large scale in terms of the number of nodes, and hence the diameter of the network. This in turn means that there is a tremendous amount of *spatial reuse* possible in the network, that should be tapped for achieving the best capacity, and hence delay. However, the specific loss recovery mechanism used may severely limit such spatial re-use as we elaborate in the next section.

4.4.2.2 ACK/NACK Paradox:

While the previous challenge was with regard to the constraints imposed by the environment, this challenge stems from the constraints imposed by typical message types that can be expected to use the downstream reliability. While the query-data and control code can be expected to be of non-trivial message size, queries pose a unique problem because of their short message sizes.

Negative acknowledgments (NACKs) are well established as an effective loss advertisement mechanism in multi-hop wireless networks in particular, and group communication in general as long as the loss probabilities are not inordinately high. However, NACKs cannot handle the unique case of all packets in a message being lost at a particular node in the network. Since the node is not aware that a message is expected, it cannot possibly advertise a NACK to request retransmissions.

If the message sizes are large, the probability of all packets in the message not arriving at a node will be negligible. But, for message types like queries, where it is very reasonable to expect messages to be merely a few packets long (if at all), the probability that a node does not receive any packet in a message is non-negligible, and hence has to be explicitly tackled.

While an ACK based recovery scheme would address the problems, its other deficiencies (in terms of ACK implosion) however clearly prohibit it from being used.

Finally, revisiting the issue of tapping spatial re-use, a NACK based loss recovery scheme will inherently require *in-sequence forwarding* of data by nodes in the network to prevent a NACK implosion [92]. This will clearly limit the spatial re-use achieved in the network.

4.4.2.3 Reliability Semantics:

Our final discussion is on constraints that are imposed by the *notion of reliability* that typical WSNs will require.

Two characteristics that are innate to a WSN environment are location dependency and redundancy in deployment. A query can be location dependent such as “Send temperature readings from rooms X, Y, and Z”. At the same time, the redundant deployment of sensors in the field means that in order to get reliable sensing *information*, it is not necessary for all sensors in the field to reliably deliver their locally sensed data to the sink. Furthermore, a sink might also choose to reliably deliver a message only to a probabilistic fraction of the entire network, say as part of a sensing strategy that

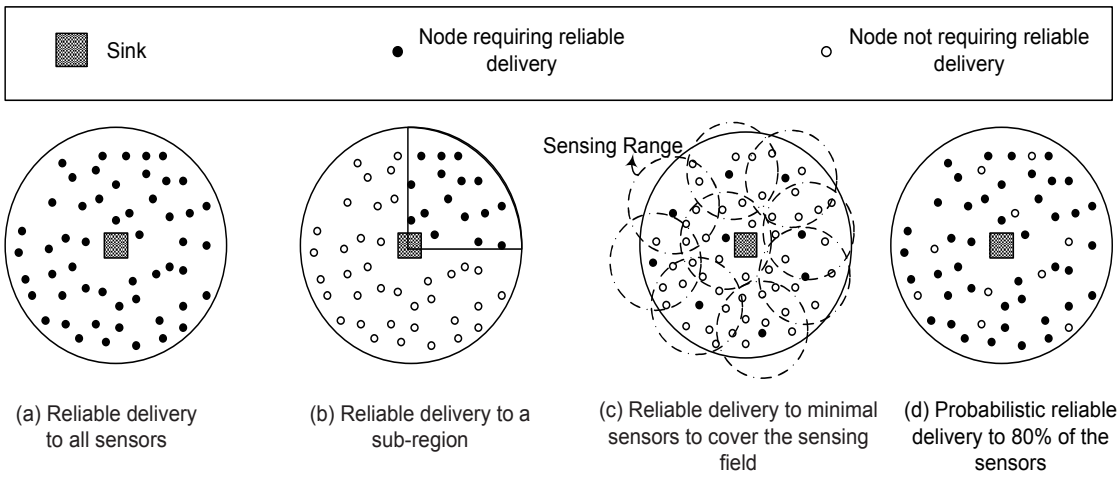


Figure 26: Types of Reliability Semantics

involves incrementally increasing resolution.

We thus define the reliability semantics that can be required in WSNs based on the above characteristics. We classify the reliability semantics into four categories: (i) *delivery to the entire field*, which is the default semantics, (ii) *delivery to sensors in a sub-region of the field*, which is representative of location based delivery, (iii) *delivery to sensors such that the entire sensing field is covered*, which is representative of redundancy aware delivery and (iv) *delivery to a probabilistic subset of sensors*, which corresponds to applications that perform resolution scoping.

Figures 26(a)-(d) illustrate categories (i) through (iv) respectively. Thus, any reliability solution should not only support the default reliability semantics, but also the other types of semantics that are unique to wireless sensor environments.

In the rest of the thesis, we describe the GARUDA-DN approach to provide reliable sink-to-sensors data delivery.

4.5 Theoretical Approaches

Before proposing a solution of the problem for reliable downstream delivery, we need to address the problem theoretically in order to solve the problem optimally. In this section, a theoretical approach

for the problem is shown.

4.5.1 Ideal Solution: Minimum Set Cover Problem

To solve the reliability problem at wireless sensor networks, it is necessary to formulate the problem into an optimization problem which has been known as a common and typical problem and investigated for optimal solutions.

4.5.1.1 Problem Description

As each packet is broadcasted through a WSN, some nodes cannot receive the packet correctly and successfully. For each packet, a WSN has different sets of nodes which do not receive the packet. Assuming that the lost packet can be retransmitted and recovered by one of neighbors which received the lost packet before, a solution tries to designate a set of nodes, called recovery servers, which retransmit the lost packet in an optimal fashion. We will call this problem as loss recovery server designation problem.

By the nature of local broadcasting of wireless communication, one recovery server can recover the lost packet of all neighbors around it. Therefore, it is optimal to minimize a size of the set of recovery servers covering all nodes which did not receive the packet. And it is necessary to find the optimal recovery sets for different loss patterns of each packet.

4.5.1.2 Problem Definition

The above loss recovery server designation problem can be defined as a set cover problem in the graph theory, the problem of covering a base set (nodes which did received a packet successfully) with as few elements of a given subset system (a set of recovery servers) as possible. Based on the above problem description, the formal definition can be derived as follows:

Minimum Set Cover

Instance: A base set $S = \{s_1, \cdots, s_k\}$, a set of nodes which did not receive a packet successfully, and a collection $F = \{S_1, \cdots, S_m\}$ of m subsets of S, where m is bounded by some polynomial in n.

Problem: Find a subset C[6] $\subseteq F$ with minimum cardinality, such that every element of S is

[6]Each subset in C is the set of nodes which are covered by a recovery server.

contained in at least one of the sets in C.

[42] showed that the decision version of the minimum set cover (MSC) is NP-complete.

A common approach of coping with NP-hard problems is approximation algorithms that run in polynomial time and deliver solutions that are close to the optimal solution. For set cover, we evaluate an approximation algorithm by considering the ratio between the number of subsets used in the cover output by the algorithm and the number of subsets used by the optimal solution. This ratio is always at least one, and the largest value that it can attain depending on an input instance set S is the approximation ratio of the algorithm defined as $\frac{APX(MSC)}{OPT(MSC)}$ such that $APX(MSC)$ is the cost of approximation algorithm and $OPT(MSC)$ is the cost of optimal solution.

In general, there is a lower bound for performance ratio of the approximation of MSC to the optimal solution of MSC. And [21] showed that there is no efficient approximation algorithm with a quality guarantee better than that of a simple greedy algorithm, namely $\ln(k)$.

4.5.2 Reduction to Minimum Dominating Set Problem

If we define the loss recovery server designation problem as the MSC problem, the given set S will be different depending on loss patterns. From the definition of MSC problem, the set S for different packets will be different because the given set S is a set of nodes which did not receive a packet. Therefore, to find the minimum number of servers to cover the loss pattern S, we need to individually solve the MSC problems whenever loss patterns are different to previous ones.

In real environment, it is hard to solve the NP-complete problem with decentralized fashion. What is worse, it is impractical to separately solve different instances of the MSC problem depending on different loss patterns in WSNs.

Therefore, we address the loss recovery server designation problem with an alternative which has similar complexity and advantages to solve the problem in decentralized fashion.

4.5.2.1 Definition of Minimum Dominating Set (MDS)

In a graph, a dominating set is a subset of nodes such that for every node v in a graph, either a) v is in the dominating set or b) a direct neighbor of v is in the dominating set. The minimum dominating

set problem asks for a dominating set of minimum size. It's formal definition is as follows:

Instance: A graph $G = (V, E)$.

Problem: Find a subset D with minimum cardinality for G, i.e., a subset $D \subseteq V$ such that for all $u \in V - D$, there is a $d \in D$ for which $(u, d) \in E$.

4.5.2.2 Reason to Choose MDS

The MSC is equivalent to the MDS problem under L-reduction closely related to each other and have been shown to be NP-hard [26, 42]. In common, two problems try to find a subset covering neighbors with minimum cardinality. Therefore, one of two problems can be transformed into the other without loss of information. Instead of solving MSC problems, we can solve the other problems of MDS that are transformed from the MSC problems.

In network research area, MDS problem has been considered as one of popular approaches to solve various networking problems[81]. Since WSNs require decentralized algorithms, it is easy to address the MSC problem using the MDS problems of which solutions have characteristics of decentralization.

4.5.2.3 Decoupling a Solution of MDS from Loss Patterns

Although the MDS problem has different instances reduced from different instances of MSC problem, an instance for MDS problem can include a whole network by covering a set of nodes and edges which are not adjacent to a given set S. Therefore, we can handle the MDS problem without concerning the loss pattern S although there are trade-offs: the advantage of MDS is that we can solve MDS problem without considering different instances for different loss patterns; and the disadvantage of MDS is that the cost of optimal solution for an instance of MDS is larger than that of optimal solution for an instance of MSC for given loss pattern S.

4.5.3 Performance Ratio between MDS and MSC

To make a solution practical, we need to decouple the given set S from assumptions of the problem. Therefore, the practical solution of MDS problem will cover all nodes in a WSN irrespective of

specific loss patterns.

To find the ratio between the practically approximated solution and the optimal solution, we define two costs(The cost means the number of dominating nodes in a given graph or the size of cover set in a set system): $PAPX(MDS)$ is the practically approximated solution for the MDS problem that does cover not loss patterns S, but all nodes V; and $OPT(MSC)$ is the cost of optimal solution for the MSC problem that is related to loss patterns. Then the cost ratio is $\frac{PAPX(MDS)}{OPT(MSC)}$.

Given a graph $G=(V,E)$ and a set system (X,S), we assume that a given graph $G=(V,E)$ has the maximum degree of network, G_d which limits the maximum number of neighbors at any node in the graph. To compare the costs with easy explanation, we divide a problem into three cases: (i) a given set S is subset of dominating set D which is approximated by practical solution, (ii) a given set S is subset of the complement set $\overline{D}$, and (iii) the other case when a part of given set S is subset of set D and remaining part of set S is subset of set $\overline{D}$.

CASE 1:

If $S \subseteq D$, each element s_i in a set $S=\{s_1, \mathrm{csc} \cdots, s_k\}$ is located at one of nodes in set D which guarantees minimum number of nodes covering all nodes in V.

In this case, the cost of $PAPX(MDS)$ to cover set S is equal to k that is the size of set S although the size of dominating set D is larger than that of S. And in best case, the cost of optimal solution of MSC is as follows:

$$OPT(MSC) \geq \frac{|S|}{G_d} \tag{22}$$

since the maximum number of nodes in S to be covered is limited by the maximum degree G_d of a graph. Therefore, we can find the upper bound of the ratio as follows:

$$\frac{PAPX(MDS)}{OPT(MSC)} \leq \frac{|S|}{\frac{|S|}{G_d}} = G_d \tag{23}$$

CASE 2:

If $S \subseteq \overline{D}$, each element s_i in a set $S=\{s_1, \cdots, s_k\}$ is a neighbor of nodes in set D which guarantees minimum number of nodes covering all nodes in V.

In worst case, the cost of $PAPX(MDS)$ to cover set S is less than or equal to k since each element

s_k can be dominated by different nodes in set D. And in best case, the cost of optimal solution of MSC is same as in (22) since the maximum number of nodes in S to be covered is limited by the maximum degree G_d of a graph. Therefore, we can find the upper bound of the ratio same as in (23).

CASE 3:

If $S_D \subseteq \overline{D}$ and $S_{\overline{D}} \subseteq \overline{D}$, such that $S = S_D \cup S_{\overline{D}}$, in worst cast, each element in set S_D and set $S_{\overline{D}}$ still can be dominated by different nodes in set D. Therefore, $|S|$ number of nodes from MDS set D are required to cover set S at most. And in best case, the cost of optimal solution of MSC is same as in (22) since the maximum number of nodes in S to be covered is limited by the maximum degree G_d of a graph. Therefore, we can find the upper bound of the ratio same as in (23).

Therefore, for all cases, the ratio $\frac{PAPX(MDS)}{OPT(MSC)}$ is bounded by G_d, which is not much different to the approximation ratio of $\frac{APX(MSC)}{OPT(MSC)}$, $ln(k)$, because the average value of G_d is $log(n)$ [31].

From the above proof, we can use the approximated solution of MDS to solve the MSC which is the optimal solution of the loss recovery server designation problem.

4.6 GARUDA-DN Design

In this section, we present an overview of GARUDA-DN's design that explicitly tackles the challenges identified in Section 4.4. The centerpiece of GARUDA-DN's design is an instantaneously constructible loss recovery infrastructure called the *core*. The *core* is an approximation of the minimum dominating set (MDS) of the network sub-graph to which reliable message delivery is desired. While using the notion of a MDS to solve networking problems is not new [81], the contributions of this work lie in establishing the following for the specific target environment: *the relative optimality of the core for the loss recovery process, how the core is constructed, how the core is used for the loss recovery*, and *how the core is made to scalably support multiple reliable semantics.*

We present a *core construction* approach that constructs the *core* during the course of a single packet flood, and propose a *two-phase loss recovery* strategy that uses *out-of-sequence forwarding* and is tailored to satisfy our basic goals of minimizing retransmission overheads and minimizing delay. Finally, we show how a simple *candidacy* based approach for *core* construction can make the

core scalably support multiple reliability semantics.

The second cornerstone of the GARUDA-DN design is a pulsing based approach to deliver a single packet reliably to all the network nodes. Recall the trade-offs identified in Section 4.4 for reliable delivery of short-messages. Since GARUDA-DN can ensure the reliable delivery of the first packet of messages of any size, it is no longer vulnerable to the *all packets lost problem* that straightforward NACK based schemes are susceptible to. This enables GARUDA-DN to tap the advantages of NACK based schemes, but at the same time avoid any pitfalls that consequently arise.

In the rest of the section, we provide high level overviews of each of the above components. For the sake of clarity, we start with discussing the details about the *core* infrastructure in GARUDA-DN. We assume that the first packet is reliably delivered for the initial discussions. Then, in Section 4.6.4, we present the details of how GARUDA-DN achieves reliable first packet delivery.

4.6.1 Loss Recovery Servers: Core

GARUDA-DN uses *local and designated loss recovery servers* in its loss recovery process. The motivations for localized recovery - reducing bottlenecks at the (otherwise) non-local servers, and reducing recovery time; and designated servers - preventing unnecessary redundant retransmissions by neighbors upon a retransmission request, have been well established in related work ([24]), and we do not delve further into the motivation in the interest of space.

The *core* in GARUDA-DN thus forms the set of local designated loss recovery servers that help in the loss recovery process. The challenges that hence arise are (i) how should the core nodes be chosen in order to minimize retransmission overheads? and (ii) how can the *core* be constructed in a manner that is appropriate for the limiting characteristics (dynamic topology change due to node failures) of the target environment?

Ideally, the core designation should be done on a per-packet basis based on the loss pattern experienced during the packet delivery. Once the loss pattern is known, performing optimal[7] server designation reduces to the well known *minimum set-cover problem* (MSC) [37] as discussed in Section 4.5.

While the solution to the set-cover problem is ideal, it is obviously not a feasible one from the

[7] In terms of the number of retransmissions required.

standpoint of performing such a core designation on a per-packet basis.

GARUDA-DN, instead, performs core designation on a per-message basis[8], and independent of the loss patterns of the packets. It designates loss recovery servers by dynamically electing a subset of the nodes in the network as *core* nodes for each message delivery. While the *core* is thus not optimal for each packet loss pattern (does not approximate the minimum set cover for the loss pattern), it approximates the minimum dominating set (MDS) [22]

4.6.1.1 Instantaneous Core Construction

In GARUDA-DN, the *core* is constructed using the first packet delivery. The reliable delivery of the first packet determines the *hop_count* of the node in the network, which is the distance of the node from the sink. A node, which has a *hop_count* that is a multiple of three, elects itself as a core if it has not heard from any other core node. In this fashion, the core selection procedure approximates the MDS structure in a distributed fashion. The uniqueness of the *core* in GARUDA-DN lies in the following characteristics: (i) the *core* is constructed using a single packet flood, more specifically during the flood of the first packet; and (ii) the structure of the sensor network topology (with sensors placed at fixed distances from the sink) is leveraged for more efficient, and fair *core* construction. Note that such an instantaneous construction of the core nodes during the first packet delivery of every new message addresses any vulnerability in the network in terms of node failures occurring at the granularity of a message. We defer the discussion of how our approach handles node failures occurring during a message transmission to Section 4.7.

4.6.2 Loss Recovery Process

4.6.2.1 Out-of-Sequence Forwarding with A-map *Propagation*

In GARUDA-DN, an out-of-order forwarding strategy is used while forwarding packets as opposed to an in-sequence forwarding scheme. A key drawback of the in-sequence forwarding strategy is that precious downstream network resources can be left under-utilized when forwarding of higher sequence number packets is suppressed in the event of a loss. An out-of-sequence forwarding on the other hand can overcome this problem as nodes that have lost a packet can continue to forward

[8]Performing designation at any larger time granularity will compromise the goal of addressing network dynamics and supporting different reliability semantics.

any higher (or lower) sequence number packets that they might have received. However, such an approach can potentially lead to unnecessary NACK implosion, where downstream nodes will issue a chain of NACK requests for holes detected in the sequence of packets received, even when the concerned packets are not available.

To inhibit such unnecessary retransmission requests, GARUDA-DN uses a scalable *A-map* (Availability Map) exchange between core nodes that conveys meta-level information representing availability of packets with bits set. Any downstream core node initiates a request for a missing packet only if it receives an *A-map* from an upstream core node with the corresponding bit set. The *core* recovery phase in GARUDA-DN is highly efficient as the core nodes initiate requests only when they are sure of an upstream core node having a particular packet. While the overhead associated with the A-map is an obvious concern, the performance results for GARUDA-DN in Section 4.9 take into account the A-map overhead, and hence any improvements shown are after accounting for the A-map overhead.

4.6.2.2 Two-Phase Loss Recovery

Once the *core* is constructed, the framework employs a *two-phase recovery process* that first involves the core nodes recovering from all lost packets, and then the recovery of lost packets at the non-core nodes. The reasons for using two-phase recovery are threefold: (i) the number of non core nodes will be a substantial portion of the total number of nodes in the network, and hence precluding any contention from them is desirable; (ii) when the core nodes perform retransmissions for other core nodes, holes corresponding to a single packet among a core node's neighbors would also be filled with a single retransmission; and (iii) when only the core nodes are performing retransmissions during the second phase, due to the nature of the *core* (ideally, no two core nodes are within two hops of each other), the chances for collisions between retransmissions from different core nodes are minimized.

- *Loss Recovery for Core Nodes*: The recovery process for the core nodes is performed in parallel with the underlying default message-forwarding. This is done in order to ensure that the core nodes receive all the packets in a message as quickly as possible. This parallel recovery

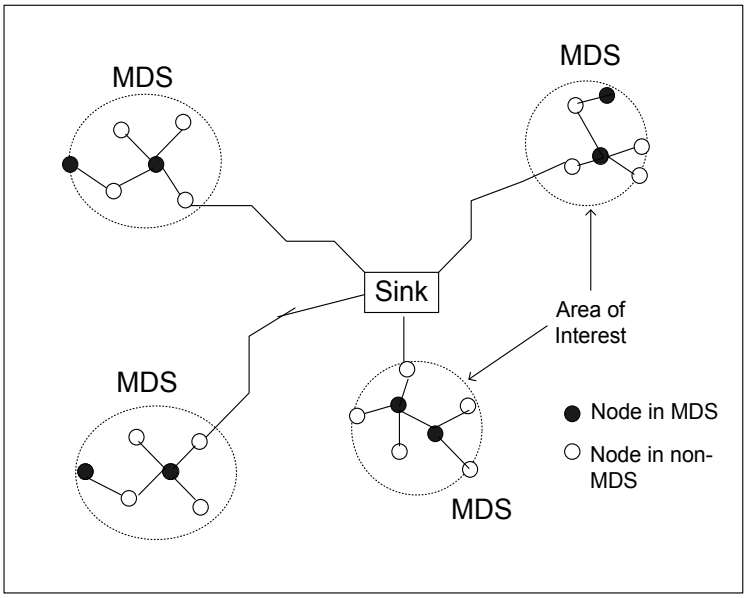


Figure 27: Core Structure When Target Subgraph $G_S \subset G$

process for the core nodes does not increase the contention in the network significantly because the fraction of core nodes is very small compared to the total number of nodes in the network, and all requests and retransmissions are performed as unicast transmissions to the nearest upstream core that has a copy of the lost packet.

- *Loss recovery for Non-core Nodes*: The second phase of the loss recovery starts only when a non-core node overhears an *A-map* from the core node indicating that the core node has received all the packets in a message. Hence, the second phase of the loss recovery does not overlap with that of the first phase in each local area, preventing any contention with the basic flooding mechanism, and with the first phase recovery.

While the two phase loss recovery can potentially increase latency, we show in Section 4.9 that the proposed framework incurs a latency which is in fact significantly smaller than competing approaches.

4.6.3 Multiple Reliability Semantics

In this section, we outline briefly how the *core* construction can be simply modified to account for the multiple reliability semantics identified in Section 4.4. We first assume, without loss of generality, that a given instance of reliability semantics will require reliable delivery to a subset

G_S of the nodes in the underlying graph G. Consider the subset G_S to consist of K components, where each component is connected, but the components themselves are not connected with each other. The desired infrastructure for such a setting will entail the computation of the MDS for each component, and connecting the components back to the sink using a *traveling salesman path (TSP)*, if bandwidth costs were the optimization criterion [91].

GARUDA-DN uses a simpler, but reasonably effective, technique of computing the individual MDSs and connecting them back to the sink using an approximation of the *shortest path tree* (SPT). While this may incur additional bandwidth costs, note that it will have the benefit of better delay properties, in addition to being implicitly constructible as we describe in Section 4.8. Figure 27 shows GARUDA-DN's solution that finds the minimum dominating set within each partition and approximates the SPT connecting all minimum dominating sets (MDS) to the sink.

The MDS within each component is constructed with minor changes to the *core* construction algorithm that merely involves nodes employing a *candidacy* check before participating in the *core* construction algorithm. The candidacy check is where nodes, upon receiving the first packet, determine whether or not they belong in the subset G_S. Nodes outside G_S but required for the construction of the SPT are inducted into the core structure through a *forced candidacy* mechanism.

4.6.4 Reliable Single/First Packet Delivery

Thus far, we have discussed the details of the *core* infrastructure in GARUDA-DN, *assuming that the first packet is delivered reliably to all nodes in the network.* In the rest of the section, we outline how such first packet reliability is achieved.

Since NACK based request schemes do not suffice for a single packet delivery (or when all packets in a message are lost) without any support, we consider an ACK based scheme as an alternative just for the first packet[9]. However, such an approach will still incur the undesirable ACK implosion problem identified in Section 4.4.

GARUDA-DN addresses the reliable delivery of the first packet using a *Wait-for-First-Packet (WFP)* pulse, which is a small finite series of short duration pulses, where the series is repeated

[9]Note that as long as one of the packets is delivered to every node with sufficient information about the message, e.g., length, a NACK based scheme can be successfully used to provide guaranteed reliability.

periodically. The pulse has an amplitude that is much larger than that of a regular data transmission, and a period that is significantly smaller than that of a regular data transmission. The unique property of the pulse is that any receiving node, irrespective of whether it is currently idle or receiving a regular data packet, can sense the pulse due to the pulse's specific amplitude/period characteristics.

When a sink wants to send the first packet, the sink transmits the finite series of WFP pulse on a periodic basis. The sensor nodes within the transmission range of the sink, upon reception of the pulses, also start pulsing with the same periodicity between two series of pulses and this process is repeated until all the nodes start pulsing in anticipation of the reception of first data packet. The sink after pulsing for a finite duration (so as to ensure that the pulses have propagated across multiple hops in the network), transmits the first packet as a regular data packet transmission and stops sending any further WFP pulses. Every sensor upon reception of the first packet also performs the same set of two actions.

Essentially, the WFP signal serves two purposes: (i) it allows the sink to inform the sensors about an impending message that has reliability requirements, and (ii) it enables sensors to request for retransmissions when they do not receive the first packet successfully. It might appear that resource constrained sensors can be overloaded, in terms of energy consumption, and cost, with the addition of the pulsing mechanism. However, we argue that the addition of the WFP signal alleviates several problems associated with reliable message delivery, and can in fact provide benefits that far outweigh the costs.

Briefly, (i) since the WFP pulse is just used to indicate the arrival of an impending new transmission, it requires a simpler modulation scheme than the default data transmissions and, is more robust to fading effects; (ii) the message advertisement scheme using WFP pulses is inherently robust to collisions, as the collisions of WFP pulse with other such pulses or data transmissions does not prevent sensors hearing the WFP pulses from inferring the impending message transmission (they still will sense that the WFP pulses are being sent) [34]; (iv) unlike in the ACK based scheme, where the ACK implosion can adversely impact the data transmissions as they do not scale well to increasing number of nodes in the network, the WFP pulse serves as an *implicit NACK* and (because of their small width) interferes to a very minimal extent with the regular data transmissions; and (v) the energy consumption of the WFP pulse is significantly smaller than that of a regular data

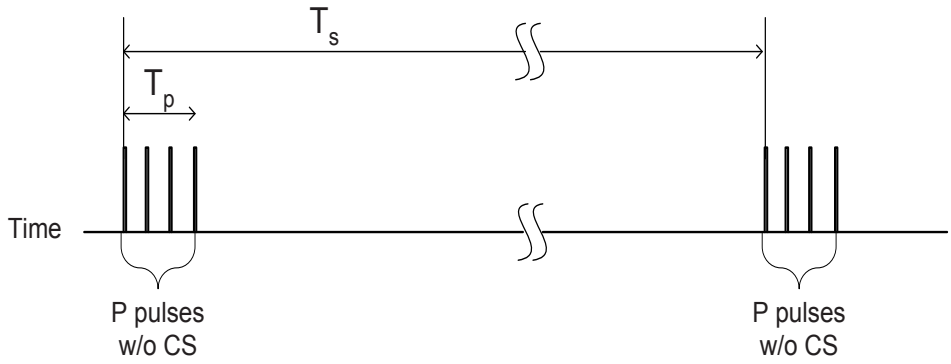


Figure 28: Transmission Time of Wait-for-First-Packet Pulse

transmission, thus rendering any additional energy consumption to be far less than the actual energy savings because of the other benefits[10].

4.7 GARUDA-DN Framework

The details of the GARUDA-DN framework is presented in this section assuming a simple underlying flooding mechanism. However, GARUDA-DN can as well be integrated with the flooding scheme itself. We assume that every incoming flooded packet is passed to GARUDA-DN if it is part of a message that requires reliability.

The different components of GARUDA-DN are explained in the chronological order that they occur when a reliable message is flooded. Hence, we first describe the details of GARUDA-DN's pulse based single packet delivery mechanism, and then go on to describe the *core* construction and loss recovery procedures. Note that the reliable single packet delivery is leveraged for the instantaneous *core* construction.

4.7.1 Single/First Packet Delivery

4.7.1.1 WFP Pulse Transmission

Since a WFP Pulse can be regarded as a short period signal which does not include any information, the transmission period of the WFP pulse is significantly smaller when compared to the transmission time T_D required for a regular data packet. Also, twice the regular transmission power is used

[10] We profile the energy savings through the use of WFP pulses in Section 4.9.

to transmit the pulses to achieve a relative amplitude of 3dB at the receiver (with respect to a default reception). The detection of a WFP pulse at a receiver is done based on a simple energy detection strategy that monitors changes in the amplitude of the energy of the incoming signal, and the duration of any such changes [34]. Note that the changes in energy can be detected even at receivers whose local channel is busy with an ongoing data transmission. The only nodes that cannot hear the WFP pulses are those that are not listening (either in transmit mode, or in a power-down mode).

To increase the robustness of the pulse detection, every set of pulse transmission includes p pulses transmitted consecutively within a period T_P ($T_P << T_D$). Figure 28 shows the transmission scheme for the WFP pulse. Hence, receivers infer an incoming WFP signal only after detecting p pulses.

The basic (and the only required) mechanism for WFP pulsing in GARUDA-DN does not use any carrier sensing, and hence is referred to as *forced WFP pulsing*. This ensures that nodes that need to transmit the WFP (either as an advertisement or a NACK for the first packet) can do so without having to suffer from any MAC layer starvation problems. However, such transmissions clearly increase the chances for collisions with regular data packet transmissions, and hence are performed with a period T_s, where $T_s >> T_D$.

However, the forced pulsing in GARUDA-DN is complimented with a carrier-sensing based WFP, and a data packet piggybacking based advertisement scheme that reduce the impact of the forced WFP[11].

4.7.1.2 First Packet Delivery in GARUDA-DN

The delivery procedure for the single/first packet consists of three modes: (1) the advertisement which notifies the ensuing single/first packet to all nodes with the forced WFP pulses; (2) the delivery which sends the single/first packet through simple forwarding; and (3) the recovery which sends NACKs using WFP pulses to request for retransmission of the single/first packet.

Figure 29 shows the basic procedure of the single or the first packet delivery with a simple topology. When a sink wants to initiate a reliable single/first packet delivery, it sends a set of forced WFP pulses without sensing the wireless channel. When neighboring sensors hear WFP pulses,

[11] But note that the *guarantee* of reliable first packet delivery is provided only by the forced WFP.

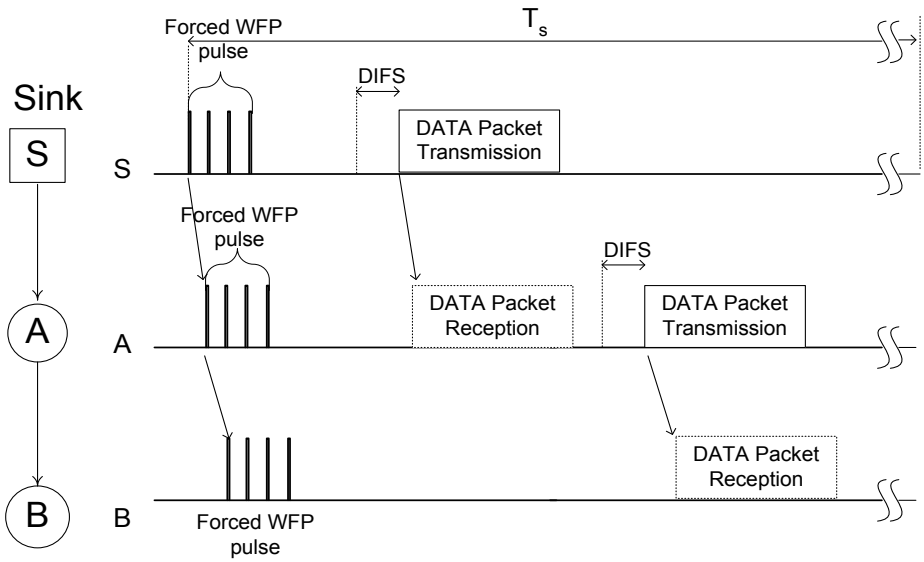


Figure 29: Example for Single or First Packet Delivery

they send a set of forced WFP pulses immediately. After a deterministic period that is set based on the diameter of the network, the sink transmits the single/first data packet subject to the medium access scheme, e.g., CSMA (Carrier Sensing Multiple Access).

If the node A receives the single/first packet, it changes its operation from the advertisement mode to the delivery mode by halting the WFP pulses, and by sending the single/first data packet after carrier-sensing. However, if the single/first packet is lost, nodes will continue to transmit the WFP pulses, which in turn trigger retransmissions. Figure 30 shows the case of retransmission.

Since the forced WFP pulses sent every T_s period play the role of a NACK signal, node B will wait for a duration of at least T_s to send next set of forced WFP pulses. Therefore, the latency for the single/first packet delivery is directly dependent upon T_s.

To reduce the latency, GARUDA-DN uses another kind of WFP pulse which a node sends after a regular carrier sensing operation. Node B sends p number of WFP pulses after carrier-sensing (WFP_{cs}) opportunistically (unless it has received the single/first packet) with a period T_c which is smaller than T_s. The period T_c should be proportional to the hop distance of the node B from the sink because a node should wait until the upstream nodes between the node and the sink receives the single/first packet. T_c is heuristically set to the following value in GARUDA-DN:

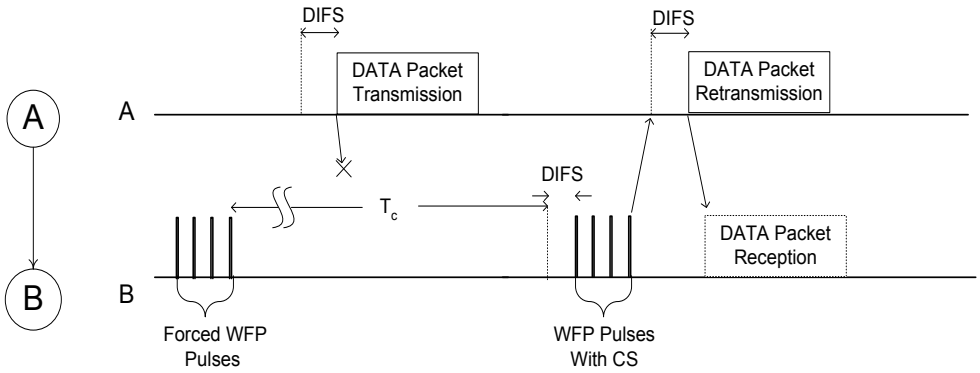


Figure 30: Example for Loss Detection and Recovery

$$T_c = i \times \Delta \times T_D, \tag{24}$$

where i is the hop distance from a sink to a node, and Δ is the maximum node degree.

Since a node senses the state of channel before transmitting WFP_{cs} pulses, the WFP_{cs} pulses have a lesser probability of colliding with data packets than WFP pulses. When a node gets to transmit WFP_{cs} pulses, it resets the timer corresponding to the T_s time period for forced WFP pulses.

A further opportunistic optimization that GARUDA-DN uses is the piggybacking of the NACK information on the regular data packet transmissions. The NACK is merely the sequence number of the last message ID the node has received thus far. Any neighbor that is aware of a greater message ID and has the corresponding first packet then retransmits. We refer to this as an implicit NACK mechanism.

4.7.2 Instantaneous Core Construction

4.7.2.1 Core

In this section we present the details of the instantaneous *core* construction assuming a simple 100% network-wide reliable flood. We revisit the case of other reliability semantics in Section 4.8.

Assuming a network organization with the sink at the center of a sensor field, the first packet delivery establishes *band-ids* for nodes based on the hop distance that they perceive from the sink[12].

[12]Note that this view of the network is purely to ensure description clarity, and has no bearing on the correctness of the approach.

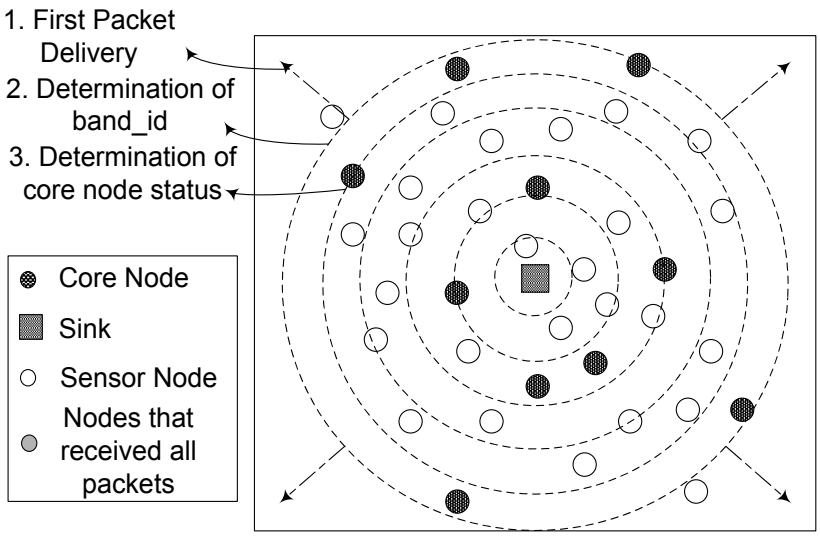


Figure 31: Instantaneous Core Construction in GARUDA-DN

This is shown in Figure 31. We consider all nodes with the same band-id as forming a "band" with a certain id. The bands can thus be viewed as concentric circles around the sink.

4.7.2.2 Procedure

The *core* construction algorithm works as follows:

Sink:

- When the sink sends the first packet, it stamps the packet with a "band-id" (bId) of 0[13]. When a sensor receives the first packet successfully, it increments its bId by one, and sets the resulting value as its own band-id. The band-id is representative of the approximate number of hops from the sink to the sensor[14].

Nodes in $3i$ **bands:**

- Only sensors with band-ids of the form $3i$, where i is a positive integer, are allowed to elect themselves as core nodes.

[13]To balance the load of core and non-core nodes, the sink can choose the band-id among 0, 1, and 2. Therefore $3i$ bands (core bands) can be changed dynamically

[14]Note that due to the availability of multiple paths from a sink to sensors, it is possible that the computed band-id is either greater than the minimum number of hops from the sink to the sensors.

- When a sensor S_0 with a band-id of the form $3i$ forwards the packet (after a random waiting delay from the time it received the packet), it chooses itself as a core node if it had not heard (or snooped) from any other core node in the same band. Once a node chooses itself as a core node, all packet transmissions (including the first) carry information indicating the same.

- If any node in the core band that has not selected itself to be a core receives a core solicitation message explicitly, it chooses itself as a core node at that phase.

- Every core node S_3 in the $3(i+1)$ band should also know of at least one core in the $3i$ band. If it receives the first packet through a core in the $3i$ band, it can determine this information implicitly as every packet carries the previously visited core node's identifier, *bId*, and *A-map*. However, to tackle a condition where this does not happen, S_3 maintains information about the node (S_2) it received the first packet from, and the S_2 node maintains information from the node (S_1) it received the first packet from. After a duration equal to the core election timer, S_3 sends an explicit *upstream core solicitation* message to S_2, which in turn forwards the message to S_1. Note that by this time, S_1 will already have chosen a core node, and hence it responds with the relevant information.

Nodes in $3i+1$ bands:

- When a sensor S_1 with a band-id of the form $3i+1$ receives the first packet, it checks to see if the packet arrived from a core node or from a non-core node. If the source S_0 was a core node, S_1 sets its core node as S_0. Otherwise, it sets S_0 as a candidate core node, and starts a core election timer[15]. If S_1 hears from a core node S_0' before the core election timer expires, it sets its core node to S_0'. However, if the core election timer expires before hearing from any other core node, it sets S_0 as its core node, and sends a unicast message to S_0 informing it of the decision.

Nodes in $3i+2$ bands:

- When a sensor S_2 with a band-id of the form $3i+2$ receives the first packet, it cannot (at that point) know of any $3(i+1)$ sensor. Hence, it forwards the packet without choosing its core

[15]The timer is set to a value larger than the retransmission timer for the first packet delivery.

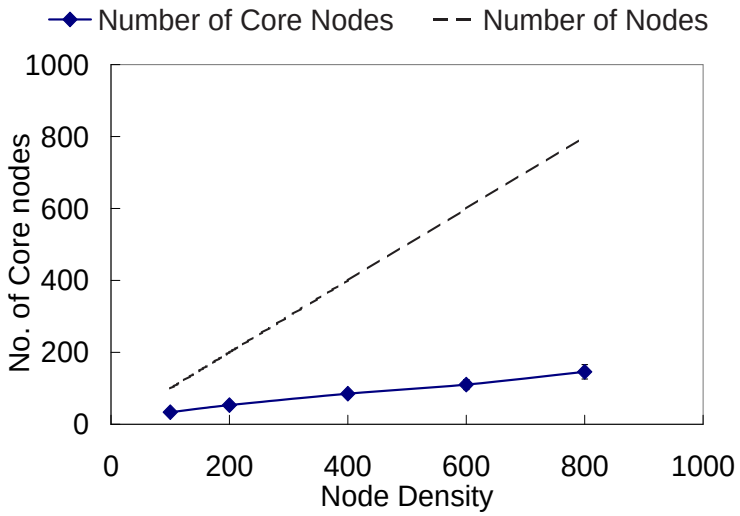


Figure 32: Number of Core Codes vs. Total Number of Nodes

node, but starts its core election timer. If it hears from a core node in the $3(i+1)$ band before the timer expires, it chooses the node as its core node. Otherwise, it arbitrarily picks any of the sensors that it heard from in the $3(i+1)$ band as its core node and informs the node of its decision through a unicast message. If it so happens that S_2 does not hear from any of the nodes in the $3(i+1)$ band (possible, but unlikely), it sends an anycast *core solicitation* message with only the target band-id set to $3(i+1)$. Any node in the $3(i+1)$ band that receives the anycast message is allowed to respond after a random waiting delay. The delay is set to a smaller value for core nodes to facilitate re-use of an already elected core node.

- A boundary condition that arises when a sensor with a band-id of $3i+2$ is right at the edge of the network, is handled by making the band act just as a candidate core band ($3i$). Such a condition can be detected when nodes in that band do not receive any response for the anycast core solicitation message.

Thus, at the end of the first packet delivery phase, each node knows its bId, whether it is a core node or not, and in the latter case its core node information. In addition, every core node in the $3(i+1)$ band knows of at least one core node in the $3i$ band.

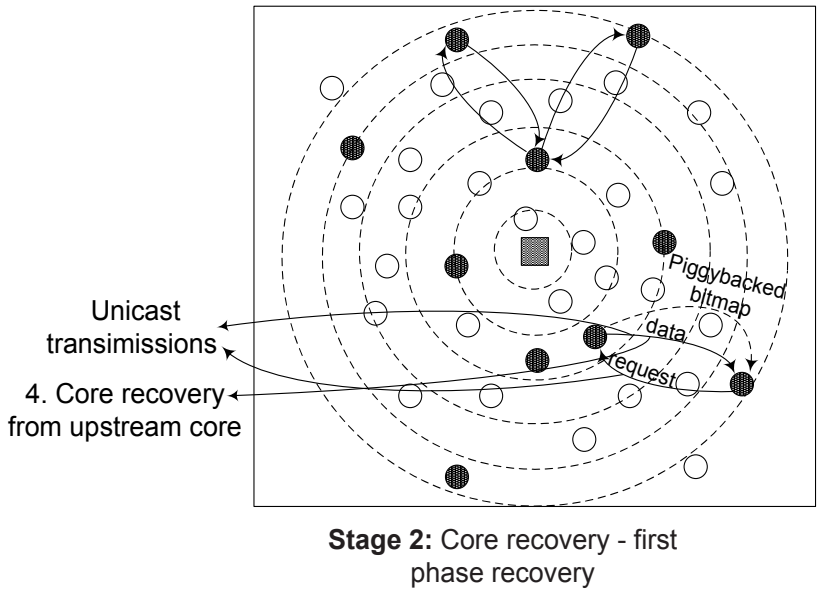


Figure 33: Loss Recovery for Core Nodes in GARUDA-DN

4.7.2.3 Optimality of the Core

Since the core nodes approximate a minimum dominating set, an obvious question is how is the *core* construction set up in a way to minimize the number of core nodes. Ideally, for any given core node, there should not be any other core node in its 2-hop neighborhood. The proposed framework attempts to achieve this condition using a two-pronged approach: (i) only nodes in 3i bands (core bands) are allowed to contend to become a core node; and (ii) of the nodes that belong to the core bands, only nodes that have not heard from any other core node from its band are allowed to choose themselves as core nodes. Figure 32 shows the number of core nodes as the node density is increased from 100 to 800. As we can see, the number of core nodes decreases from 30% when the node density is 100 to about 13% when the node density is 800. Every non-core node preferentially chooses its core by choosing a node which has already advertised itself as core in the 3*i* band, thus minimizing the number of core nodes in the network.

4.7.3 Two-Phase Loss Recovery

4.7.3.1 Loss Recovery for Core Nodes

Loss Detection When a core node receives an out-of-sequence packet, the core node infers a loss. A core node sends a request to an upstream core node only if it is notified through an *A-map* that the missing packet is available at the upstream core node.

Loss Recovery When a core node receives a unicast request from a downstream core node, it performs a unicast retransmission for the request. Figure 33 shows the loss detection and the loss recovery between core nodes at $3i$ band and core nodes at $3(i+1)$ band. If any of the non-core nodes on the path of the unicast request has the requested packet, it intercepts the request and retransmits the requested packet.

The use of the *A-map* is central to the core recovery process. For the sake of brevity, we assume that the *A-map* is capable of representing all packets of a message irrespective of the message size. The core recovery process works as follows:

Upstream Core Nodes:

- A core node, when it forwards a packet, stamps on the packet the following meta information: (C_{id},*A-map*, bId, $vFlag$), which consists of the core node's identifier, bit map, band-id, and valid flag respectively. The valid flag is used by a recipient core node to determine whether the path in the meta information is valid or not.

- When a core node receives a retransmission request, it responds with unicast retransmissions of the available packets.

Intermediate Non-core Nodes:

- Any non-core node NC_{id} that forwards a packet leaves the *A-map* information untouched, but adds its own identifier as follows: ($C_{id} + NC_{id}$,*A-map*, bId). If the number of the identifiers in the incoming information is equal to three, the non-core node does not add its identifier and sets the $vFlag$ to NULL.

Downstream Core Nodes:

- Thus, when a core node receives the meta information, it not only knows of what packets the source core node has, but also the path it can use to request for a retransmission. If the $vFlag$ is NULL, the core node still uses the *A-map* information, but falls back on any earlier cached path to the relevant core node for issuing the request.

- Each core node maintains two *A-map*s locally: *myBM* which represents the successfully received packets, and $totBM$ which represents both the received and the requested packets.

- When a core node receives an incoming *A-map* ($inBM$), it checks to see if the *A-map* is from a valid source. If the source is valid, it then checks to see if the A-map conveys availability of a packet that has neither been received nor been requested. If at least one such packet is available, the node creates a request *A-map*, updates its $totBM$, and sends the request. It also starts an expiry timer for the request.

- For a successful packet reception, the core node updates its $totBM$ and *myBM*. Also, when a timer expiry occurs for a request, $totBM$ is updated accordingly.

- When a core node does not hear an *A-map* from any of its upstream core nodes for a specified duration (*core presence timer*[16]), it explicitly issues a request to the default upstream core node to which the upstream core node responds with its current *A-map*.

4.7.3.2 Loss Recovery for Non-Core Nodes

A non-core node snoops all (re)transmissions from its core node. Once it observes an *A-map* from its core node with all the bits set, it enters the non-core recovery phase by initiating retransmission requests to the core node. Alternatively, if it does not hear from its core node for the period *core presence timer*, it sends an explicit request to the core node to which the core node responds with its current *A-map*. Figure 34 presents the loss detection and recovery between non-core nodes and a core node. Since all retransmissions from the core nodes are snooped by the non-core nodes, redundant retransmissions for the same loss are removed.

[16] *The timer is set to a value larger than three-hop round trip time.*

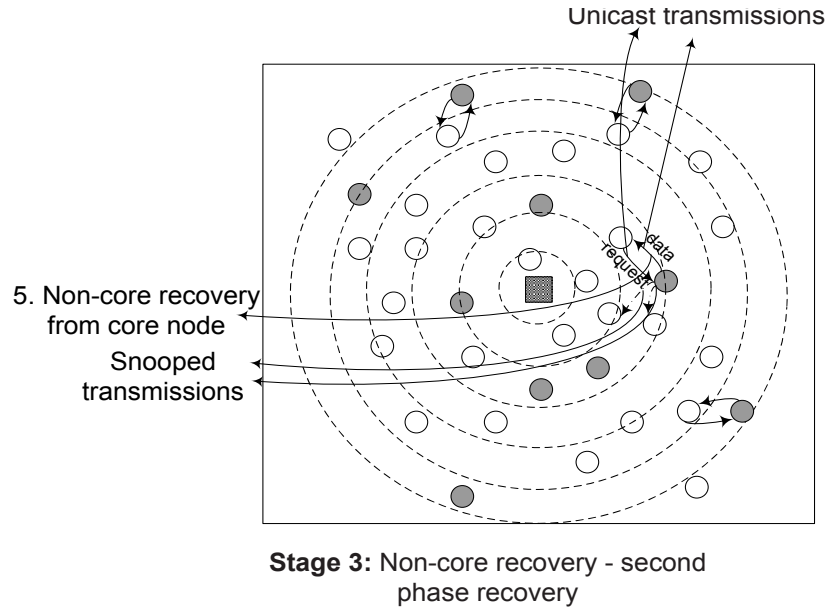


Figure 34: Loss Recovery for Non-core Nodes in GARUDA-DN

4.7.3.3 Loss Recovery in Case of Node Failures

To address the issue of node failures at core nodes and non-core nodes, we adopt simple back-up mechanisms which can reasonably handle node failures. Also note that any reliability guarantees provided in the presence of node failures are conditional on the underlying network still remaining connected. Since there is no node that relies on a failed non-core node, the failures of non-core nodes do not compromise on the reliability mechanisms of the proposed framework. Therefore, we discuss the case of core node failures which impact core nodes and non-core nodes as follows:

Downstream Core Node Depending on the Failed Core Node If a core node at $3i$ band fails after core construction, other core nodes that designate the failed core node as an upstream core suffer packet losses. After packet losses make a timer for node failures expire, other core nodes at $3(i+1)$ band fall back on an *anycast* core solicitation message to either "join" another core node at $3i$ band, or to induce another node in the core $3(i+1)$ band to join the core.

Non-Core Node Depending on the Failed Core Node If a non-core node at $3i+1$ band or $3i-1$ band realizes that its core node at $3i$ band has failed, the non-core nodes will also fall back on an anycast core solicitation message sent to 1-hop neighbors at the $3i$ band.

If non-core node at $3i+2$ band realizes that its core node at $3(i+1)$ band has failed and it is at right at the edge of the network, it declares itself as a core node and sends a core solicitation message to a core at the $3i$ band.

4.8 Supporting Other Reliability Semantics

In Section 4.7, GARUDA-DN was described in the context of single and multiple packet delivery, while assuming the simplest form of reliability semantics along the other dimensions (all nodes, 100% reliability). In this section, we revisit the GARUDA-DN design and show how it can accommodate the other reliability semantics. Specifically, we discuss three variants in terms of the reliability semantics: (i) reliable delivery to all nodes within a sub-region of the network; (ii) reliable delivery to minimal number of sensors required to cover entire sensing area; and (iii) reliable delivery to p% of the nodes in the network.

The fundamental difference between the context in Section 4.7 and in the following variants is that only a subset of the nodes in the network require reliable delivery. The variants differ in *which subset of nodes* receive the message delivery. We refer to the problem of determining the subset as the *candidacy* problem. Also, in all of the solutions discussed, the first packet is always delivered to all nodes in the network. All subsequent packets are delivered based on the candidacy.

Generically, the solutions to the three variants use three common elements to tackle the other reliability semantics:

- The first packet carries information to identify the eligibility for candidate nodes that should receive the entire message reliably. For example, in the case of reliability within a sub-region, the first packet may carry a coordinate based description of the sub-region.

- Participation in the *core* construction is limited to only those nodes that have chosen themselves as candidates. Note that the other aspects of the *core*-construction still remain the same (nodes only in the $3i$ bands can select themselves as core nodes, etc.). At the end of the *core* construction, each independent component of the candidate sub-graph G_S thus has its own core.

- The last element in GARUDA-DN is that of *forced candidacy* to enable the *core* of the different components to be connected back to the sink. Thus non-candidate nodes in the $3i$ bands on the path from each component to the sink are forced to participate as candidate core nodes to ensure connectivity. The forced candidacy is actually achievable in GARUDA-DN with very minimal changes to its original design (as described in Section 4.7). Essentially, non-candidate nodes in core bands, if they would have otherwise chosen themselves as core nodes identify themselves as non-candidate core nodes when the first packet is forwarded. A downstream candidate core node that has not heard from any other candidate upstream core node explicitly requests the upstream non-candidate core node to become a candidate. Through this process, a structure that is an approximation of independent MDSs (within each component of G_S) connected through an SPT is achieved.

In the rest of the section, we elaborate on how the candidacy for the three variants are established in GARUDA-DN.

4.8.1 Reliable Delivery within a Sub-Region

As we motivate in Section 4.4, it is quite likely that the sink requires reliable delivery of a query or a message only to sensors within a specific sub-region of the network area. We assume that the specifications of the sub-region are available in the form of coordinates. Without loss of generality, we also assume that the sub-region is rectangular in shape (although the GARUDA-DN design by itself does not have any such limitations). The sub-region can either be contiguous or non-contiguous with the region occupied by the sink.

The desired sub-region coordinates is piggybacked on the first packet sent by the sink. Each sensor in the network that receives the first packet can thus determine locally whether it is a candidate or not, based on its own location and the desired sub-region. Once the candidacy is determined, the behavior of sensors is exactly the same as described in Section 4.7, except if the sensor were to be on a core band. Whereas in the default operation, a sensor does not choose itself as a core node only if it hears from another core node before it transmits, under this variant, a sensor does not choose itself as a core node if it is not a candidate irrespective of the other conditions. Note that this does not mean that such a sensor can later be forced to become a core node, as we elaborate next.

4.8.2 Reliable Delivery to Cover Sensing Field

This variant requires reliable delivery while remaining aware of the inherent redundancy in the sensor network deployment. Specifically, under this variant, reliable delivery needs to be performed only to a minimal subset of the sensors in the network such that the entire sensing field is covered. For purposes of this discussion, we assume that the sensing range S is always less than or equal to the transmission range R.

Unlike in the previous variant where the candidacy of each node is determined locally, in this variant coordination between nodes is required in order to eliminate sensors, which are covering a region already covered by other sensors, from the candidacy. In GARUDA-DN, the core nodes are best equipped to perform such coordination as they are immediately adjacent to all non-core nodes that depend on them, and under ideal conditions are at least a distance of $2R$ away from the nearest core node (which gives a core node a virtual "ownership" of at least the sensing region defined by its transmission range). Thus, non-core nodes under this variant seek permission from their respective core nodes to become candidates. Each core node keeps track of the coverage of the region defined by the square[17] of side $2(S+T)$ (with itself at the center). It provides permission to a seeking non-core node only when the node can cover an area not already covered inside the square. Note that given our assumptions about S and T, no non-core node within a core node's transmission range can have a sensing coverage area that even lies partially outside the above defined square.

All core nodes implicitly become candidates. This is reasonable even without any coordination with other nearby core nodes as under ideal conditions, the distance between a core node and its nearby core nodes will be $2R$, which in turn means that a core node can choose itself as a candidate without concern of overlapping with a nearby core node's sensing region.

4.8.3 Reliable Delivery to Probabilistic Subset

This variant involves support for reliable message delivery to say p% of the network sensors. Such semantics might be useful when the sink intends to perform *scoped* sensing. In other words, the sink can at the outset decide to sense only 25% of the field, with the intent of increasing the sensed region only upon some triggers detected during the preliminary sensing.

[17] As an approximation of a circle for simplicity.

Just as in the case of delivery within a sub-region, determining candidacy in this variant is purely a local process. When a sensor receives the first packet, it chooses itself as a candidate with a probability of p. If the sensor is on a core band, and decides not to be a candidate, it does not choose itself as a core node irrespective of the other conditions.

4.9 Performance Evaluation

This section evaluates the performance of the GARUDA-DN framework for 100% reliability to all sensors. For single packet reliable delivery, we compare GARUDA-DN's performance with that of an ACK-based scheme that uses ACK feedback for packet delivery along with a retransmission time-out. For multiple packet delivery, we compare the GARUDA-DN framework with both in-sequence delivery and out-of-sequence delivery mechanisms that use NACKs. We also provide microscopic results to highlight the efficiency of specific components of the GARUDA-DN framework.

4.9.1 Simulation Environment

The *NS2* simulator is used for all evaluations. For all experiments: (a) the first 100 nodes are placed in a grid fashion within a 650m x 650m square area to ensure connectivity, while the remaining nodes are randomly deployed within that area, and the sink node is located at the center of one of the edges of the square; (b) transmission range of each node is 65m [74]; (c) channel capacity is 1 Mbps; and (d) each message consists of 100 packets (except for the single packet delivery part); and the size of packet is 1 KB. Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) is used as the Media Access Control (MAC) protocol. We use basic flooding as the routing protocol. All the simulation results are shown after averaging the metrics over 20 randomly generated topologies and calculating 95% confidence intervals.

As described in Section 4.2, losses can occur due to wireless channel errors, or collisions during transmissions. To emulate the two types of losses, we choose a fixed packet loss rate of 5% for wireless channel error, and vary the number of nodes in the network (and hence the network density) which in turn increases the degree of contention in the network.

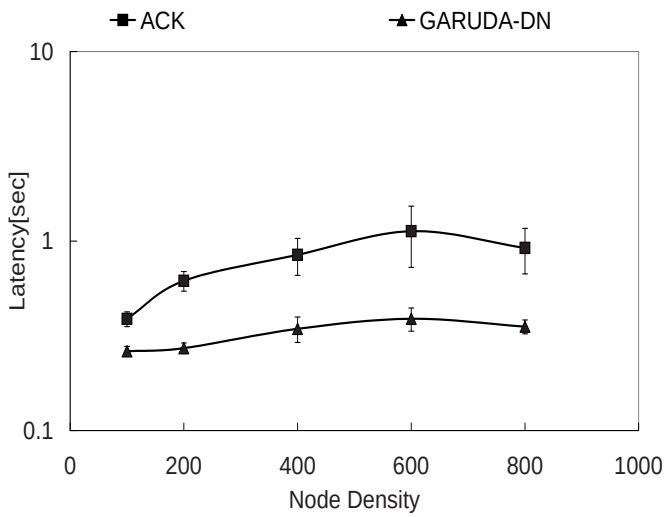


Figure 35: Latency Comparison between GARUDA-DN and Basic ACK Scheme for First/Single Packet Delivery

4.9.2 Metrics

To evaluate the performance of the GARUDA-DN framework, we use the following metrics:

- For single packet delivery, we consider (1) latency which is the duration for all nodes to receive all packets in a message, (2) the total number of data packets sent by the sink or forwarded by other sensor nodes (including recovery packets) to capture the amount of overhead involved in the reliable delivery of a message, and (3) the overall energy consumption per node.

- For multiple packet delivery, in addition to the metrics mentioned for single packet delivery, we present (4) the total number of requests sent or forwarded by nodes,

- For microscopic analysis, we consider (5) the aggregate *A-map* overhead which is measured as the total bytes spent in exchanging *A-maps*, (6) the number of recovery events that occurs during the first recovery phase between cores, and the second recovery phase between core and non-core nodes with increasing node density, and (7) the latency as the wireless error rate is varied from 0 to 50%.

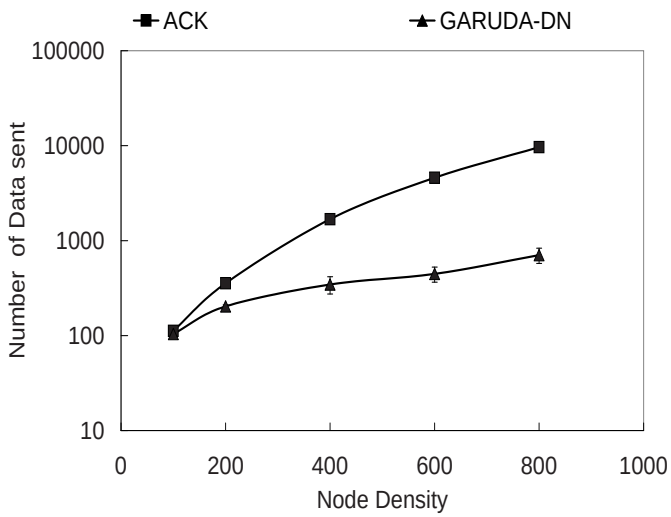


Figure 36: Number of Data Packet between GARUDA-DN and Basic ACK Scheme for First/Single Packet Delivery

4.9.3 Evaluation of Single Packet Delivery

As an alternative mechanism to provide reliability for single packet delivery, we simulate a basic ACK scheme, in which a sender retransmits a data packet until it receives an explicit ACK feedback from each of its receiving neighbors. We describe each of the results in detail below:

4.9.3.1 Latency

The latency involved in receiving a single packet reliably with increasing number of sensors is presented in Figure 35 for both the GARUDA-DN framework and the ACK based scheme. The latency of the proposed scheme was significantly smaller because of the WFP pulses approach, which used an implicit NACK scheme. This means that there was no explicit NACK sent to the sender of a packet if a packet was not received, thus not increasing the load in the network. We also observed that the latency scaled well with the increase in the number of nodes because of the same reason. However, in the ACK based scheme, the latency was appreciably higher because every ACK was addressed to the sender node and the sender retransmitted if it had not received the ACK from a particular node, thus increasing the load in the network.

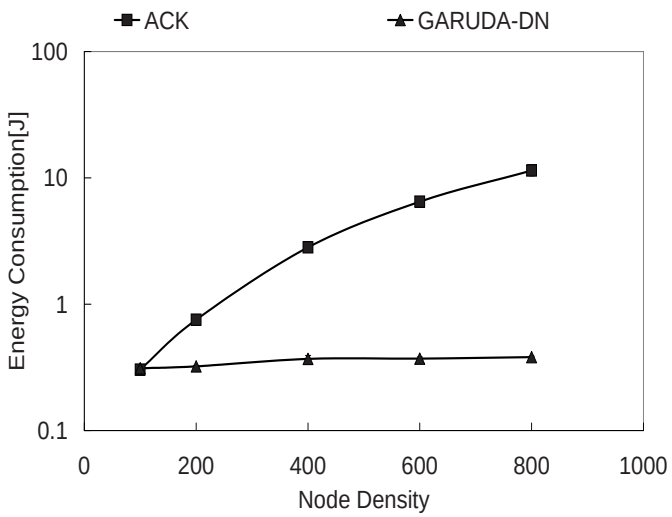


Figure 37: Energy Consumption between GARUDA-DN and Basic ACK Scheme for First/Single Packet Delivery

4.9.3.2 Number of Data Packets Sent

Figure 36 shows the number of data sent by the GARUDA-DN framework and the ACK based scheme. It is interesting to note that in the GARUDA-DN framework, the number of data sent increased more or less linearly (with a slope of 1 approximately) as the number of nodes increased. The implicit NACK scheme coupled with the inherent redundancy involved in the flooding process itself is the main reason for this trend. The implicit NACK scheme alleviates congestion related losses, while the inherent redundancy and the broadcast nature of the flooding process ensures that the packet is received successfully without any need for retransmission even in the presence of losses. For the ACK based scheme, the number of data packets sent was appreciably higher and showed a non-linear increasing trend with increasing number of nodes in the network. This is again because of the increased load in the network due to the presence of ACK transmissions thus increasing the losses in the network.

4.9.3.3 Energy Consumption

The energy consumed per node in joules for both the schemes are shown in Figure 37. The energy consumed per node was significantly smaller for the GARUDA-DN scheme than the ACK based

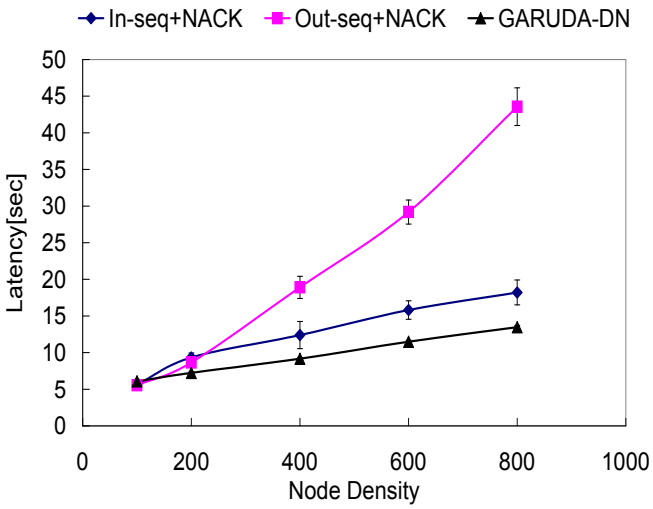


Figure 38: Latency among GARUDA-DN and Alternatives for Multiple Packets Delivery

scheme. This is because of two reasons. Firstly, the total number of transmissions on the GARUDA-DN scheme was significantly smaller than the basic ACK scheme. In fact, it showed a linear increase with increasing number of nodes. Secondly, the WFP pulses were just series of short duration pulses so that they did not consume more amount of energy than normal data packets.

4.9.4 Evaluation of Multiple Packet Delivery

To compare the performance of the GARUDA-DN framework for multiple packet delivery, we have implemented two simple reliable transport protocols that allow in-sequence and out-of-sequence forwarding respectively, coupled with NACK based error detection and non-designated local recovery servers.

4.9.4.1 Latency

Figure 38 shows the latency for 100% delivery as a function of increasing number of nodes in the network. The GARUDA-DN framework had significantly lower latencies compared to the other two schemes when the node density was increased. The reasons for reduced latencies are two-fold: (1) the advantage gained by having a designated server as opposed to a non-designated which reduced the amount of data sent (Figure 24 in Section 4.4); and (b) the advantage gained by using out-of sequence forwarding but without the NACK implosion problem. The latency of the out-of-sequence

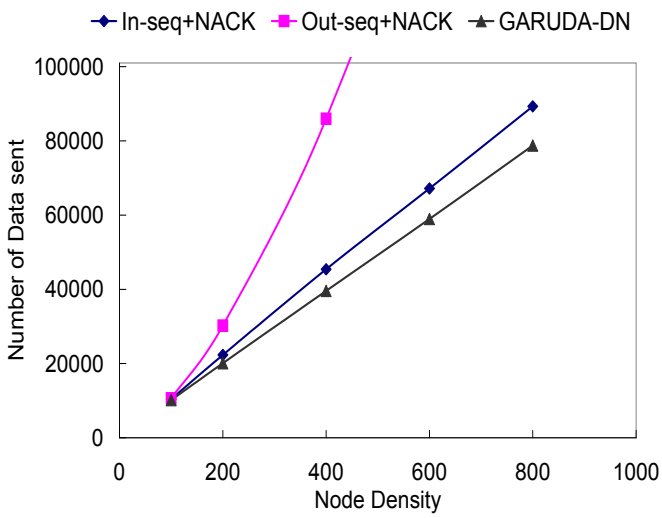


Figure 39: Number of Data Packets Sent among GARUDA-DN and Alternatives for Multiple Packets Delivery

with NACK scheme was significantly higher at higher node densities and increased at a much faster rate than the other two schemes because of the NACK implosion problem. Although, our core construction scheme used out-of-sequence delivery, we piggybacked the *A-map* of the core node along with the transmission of each packet which allows the non-core nodes to wait for the core to recover from all loses prior to any retransmission requests thus eliminating the NACK implosion problem.

4.9.4.2 Number of Data Packets Sent

The number of data sent for all three schemes are presented in Figure 39. Among the three schemes, GARUDA-DN performed the best followed by the in-sequence with NACK and the out-of-sequence with NACK schemes. The number of packets sent in GARUDA-DN was about 10% lower than that of in-sequence with NACK scheme for node density of 400, 600, and 800 and 55% to 80% lower when compared with out-of-sequence with NACK scheme. The reasons for GARUDA-DN's significantly better performance is again mainly due to the improvement gained by having a designated server as opposed to a non-designated server; and for the out-of-sequence with NACK scheme, poor result is due to the NACK implosion problem.

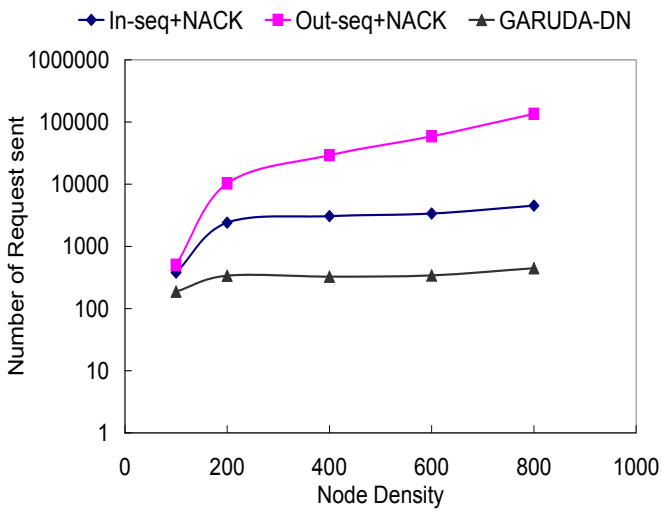


Figure 40: Number of Loss Recovery Request Packets Sent among GARUDA-DN and Alternatives for Multiple Packets Delivery

4.9.4.3 Number of Request Packets Sent

Figure 40 shows the number of requests sent with increasing node density for all three schemes. GARUDA-DN had the least number of requests, followed by the in-sequence and the out-of-sequence schemes. The values for GARUDA-DN were about 20-30% of the in-sequence scheme and about 1-5% of the out-of-sequence scheme depending on the node density. The reason for the increasing trend shown by the out-of-sequence scheme with increasing node densities is again due to the NACK implosion problem. While in-sequence uses a per packet NACK scheme, GARUDA-DN uses *A-map* exchange when doing a NACK request and hence recovers from all losses that can be serviced (depending on whether the requesting node is core or not, this is different)in one request.

4.9.4.4 Energy Consumption per Node

The average energy consumed per node is significantly smaller for the GARUDA-DN case when compared to the other two cases (Figure 41). The average energy consumed for all three cases was directly proportional to the number of transmissions, which was the sum of the number of requests sent and the number of data sent per node. Since, the sum of the number of request and data sent was the least for the GARUDA-DN scheme, the energy consumed per node was also significantly lesser. In fact, results indicate that the energy consumed per node was about 30% lesser compared

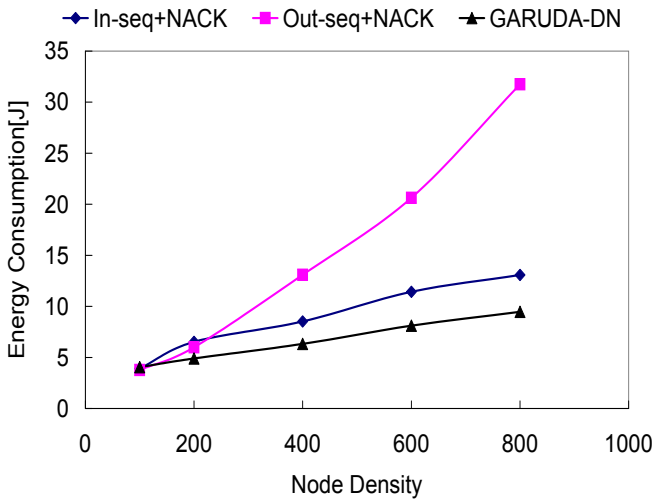


Figure 41: Energy Consumption among GARUDA-DN and Alternatives for Multiple Packets Delivery

to the in-sequence case and about 80% lesser compared to the out-of-sequence scheme for 800 node scenario.

4.9.5 Microscopic Analysis

4.9.5.1 A-map *Overhead*

The second important aspect in the GARUDA-DN framework is the overhead incurred by *A-map* transmission by the core nodes while sending both data and requests and the non-core nodes while sending the requests only. While we do not expect the *A-map* overhead to be a problem for non-core nodes as their recovery happens only after their corresponding core nodes have recovered from all loses, it is an issue for the core nodes. However, Figure 42 indicates otherwise when the *A-map* overhead is compared with the total data sent by the GARUDA-DN scheme. There are two main reasons for this: firstly, the number of core nodes was only a small fraction of the total number of nodes (10-30%); and secondly, the number of requests was substantially lower (less than 1%) than the amount of data. In fact, from the Figure 42, we see that the *A-map* overhead was only 0-3% of the total amount of data sent.

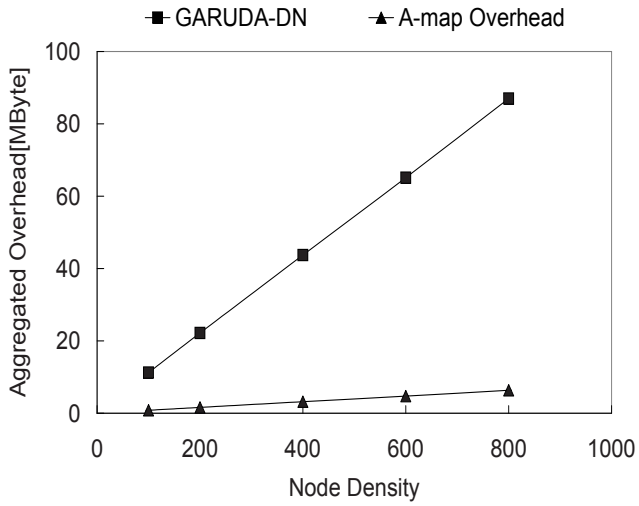


Figure 42: Microscopic Analysis: the *A-map* Overhead

4.9.5.2 Number of Recovery Events

We investigate the number of recovery events for core and non-core nodes and compare it with the total number of recovery events (Figure 43). This result helps us understand the two phase recovery process better. It shows that the core recovery process (first phase recovery) was two times more likely than the non-core recovery process (second phase) since non-core nodes were allowed to snoop recovery packets during the first recovery phase.

4.9.5.3 Effect of Random Wireless Errors

We have compared GARUDA-DN with the in-sequence with NACK scheme for packet error rates ranging from 0% to 50%. For fair comparison between results of GARUDA-DN and those presented in [92], a linear topology consisting of 21 sensors was used in the simulation. Figure 44 shows that the latency of the GARUDA-DN was much shorter than that of the in-sequence with NACK, and the difference between them increased with the increase of error rate. Although we assume a severe environment with error rate up to 50%, the latency of the GARUDA-DN shows almost constant. GARUDA-DN, therefore, is more adequate to wireless sensor networks since wireless sensor networks experience higher error rate than other wireless networks.

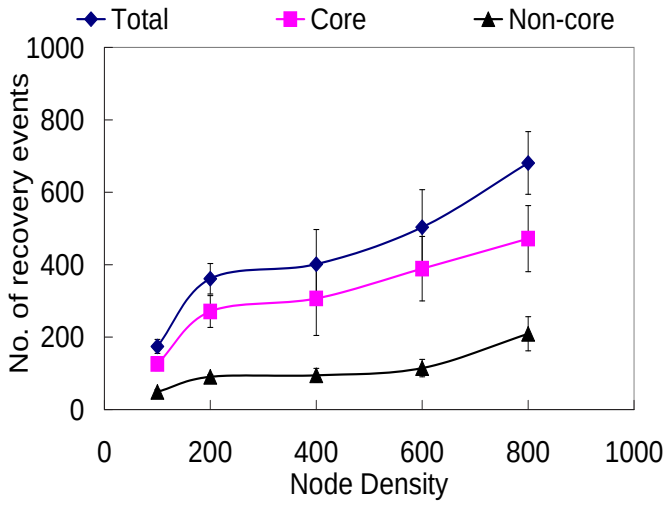


Figure 43: Microscopic Analysis: the Number of Recovery Events

4.9.6 Evaluation of GARUDA-DN Variants

4.9.6.1 Reliable Delivery within a Sub-Region

Figures 45(a)-(c) present performance results for the first variant for a 200 node, 650mx650m network with a transmission range of 67m per node. Figure 45(a) shows the partitioning of the network grid into sub-regions. Figure 45(b) shows the latency incurred with increasing number of regions for both contiguous and non-contiguous regions respectively. While it is obvious that the latency increases with increasing number of regions, an interesting observation is that they latency for the non-contiguous regions scenario is always more. Recall that this is due to the latency involved in non-candidates being forced into candidacy. Figure 45(c) shows the number of data packets transmitted for the same scenarios. For the contiguous regions scenario, the achieved number of candidates is typically very close to the ideal number of candidates. However, for the non-contiguous regions, the achieved numbers are typically higher due to the forced candidacy of nodes to achieve connectivity.

4.9.6.2 Reliable Delivery to Minimal Set of Sensors

Figure 46 shows the number of nodes selected as candidates for the second variant. It can be observed that the number of nodes chosen decreases with increasing ratio $\frac{S}{R}$. The decrease is not much for the smaller values of $\frac{S}{R}$ because for the scenario considered (400 nodes in a 650mx650m

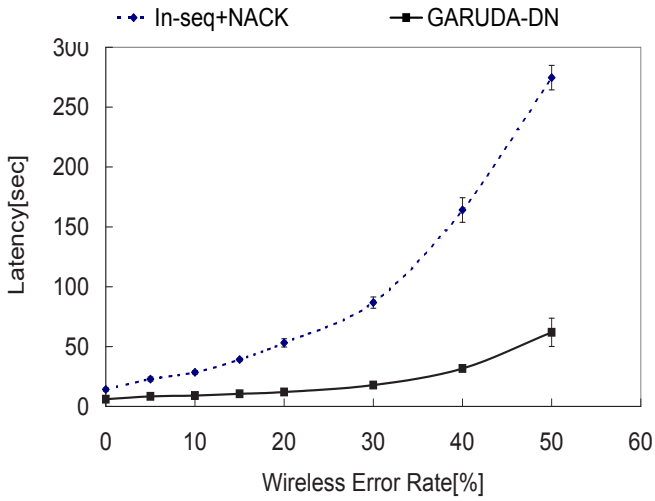


Figure 44: Latency of GARUDA-DN for Different Loss Rates

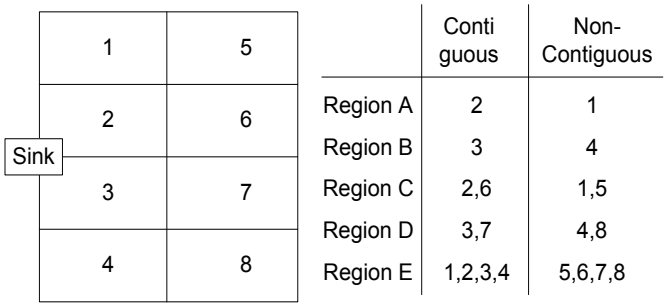
grid with a transmission range of 67m), the minimum value for $\frac{S}{R}$ required to cover the entire area is approximately 0.5. As the ratio of $\frac{S}{R}$ increases beyond 0.6, we see a more pronounced decrease in the number of candidate nodes. This is because the overlap area among nodes become more pronounced as the sensing range approaches the transmission range.

4.9.6.3 Reliable Delivery to Probabilistic Subset

Figure 47 presents simulation results for the third variant. The scenario considered is 200 nodes in a 650mx650m grid, with nodes having a transmission range of 67m. The number of candidate nodes chosen with increasing probability is shown. It can be seen that at lower probabilities, the achieved number of candidates is larger than that of the expected number due to the forced candidacy of nodes to achieve connectivity. However, for larger probabilities ($\geq$ 50%), the achieved number of candidate nodes closely approximates the ideal values.

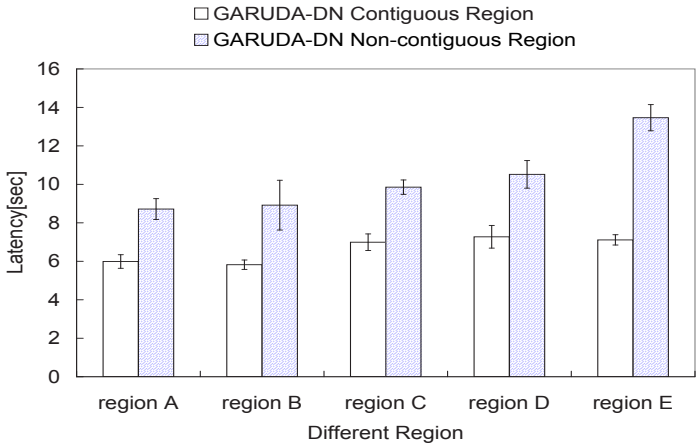
4.9.7 Summary of Evaluation

We have compared GARUDA-DN with an ACK based scheme (for single packet messages) and with both in-sequence and out-of-sequence NACK schemes (for multiple packet messages) and results indicate that GARUDA-DN performs substantially better both in terms of latency and the number of retransmissions. The cardinal reason for GARUDA-DN outperforming the ACK based

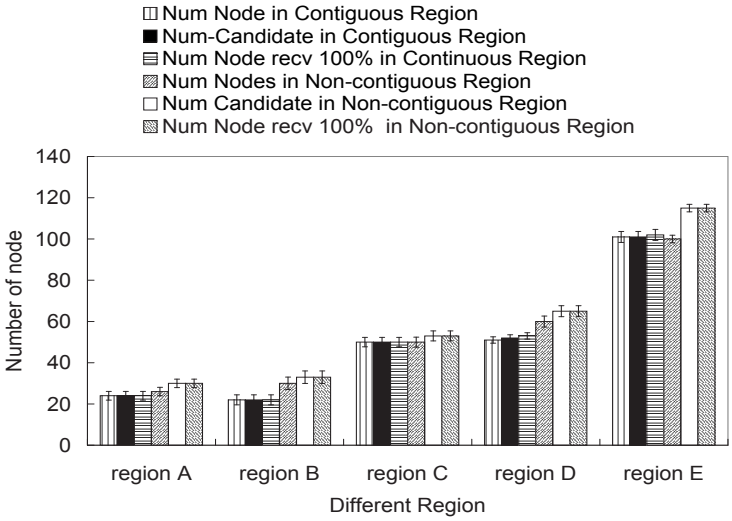


	Conti guous	Non-Contiguous
Region A	2	1
Region B	3	4
Region C	2,6	1,5
Region D	3,7	4,8
Region E	1,2,3,4	5,6,7,8

(a) Layout of Sub-Regions



(b) Latency for Different Sub-regions



(c) No. of Nodes Requiring Reliable Delivery for Different Sub-regions

Figure 45: Reliable Delivery to All Sensors in a Sub-region

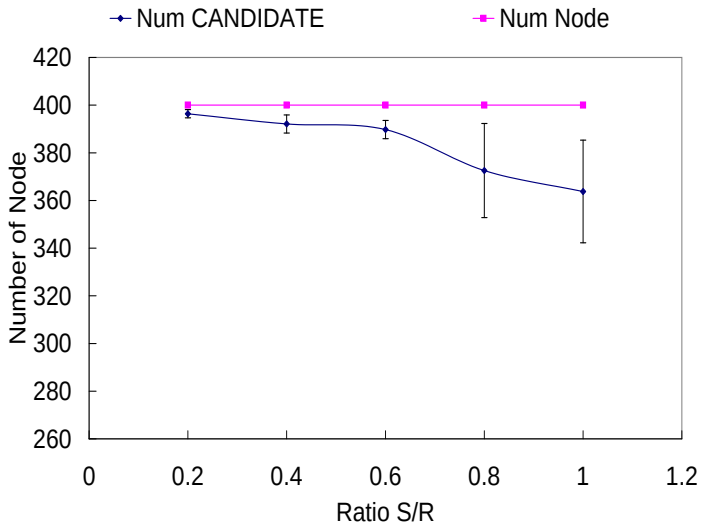


Figure 46: Reliable Delivery to Minimal Number of Sensors in a Region

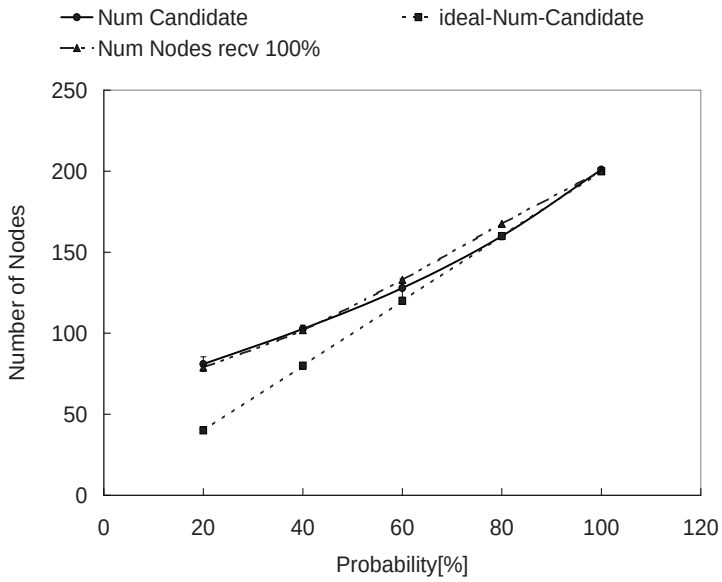


Figure 47: Probabilistic Reliable Delivery of GARUDA-DN Variant: the Number of Candidates

scheme is because of the WFP pulses approach used by GARUDA-DN where the WFP pulses act as an implicit NACK, thus not increasing the load in the network. GARUDA-DN performs better in the multiple packet messages scenario largely due to the way it addresses the NACK implosion problem by *A-map* propagation while exploiting the high spatial reuse due to out-of-sequence forwarding.

4.10 Summary

In this thesis, we have proposed a new framework for providing sink-to-sensors reliability in wireless sensor networks. We have identified several challenges to provide sink-to-sensors reliability and addressed the challenges by proposing key elements: (1)Wait-for-First-Packet (WFP) pulse, (2) core structure approximating the minimum dominating set, (3) instantaneously constructible optimal core structure, (4) availability bitmap, and (5) two-phase recovery process. Note that, although we have proposed an effective way to realize the WFP pulse in-band, it is equally possible to use out-of-band signaling in scenarios where a pilot radio is available. We have also identified three new types of reliability semantics unique to downstream sensor environment and elaborated how our proposed framework can provide reliability to such variants. We have shown through ns2-based simulations that the proposed framework performs significantly better than the basic schemes proposed thus far in terms of latency and energy consumption. We have also profiled the A-map overhead in GARUDA-DN and observed it to be minimal. We have also studied how the mechanisms in GARUDA-DN can handle node failures.

CHAPTER V

GARUDA-UP: ENERGY-EFFICIENT UPSTREAM DATA AGGREGATION

5.1 Problem Definition

In Chapter 4, we addressed the reliable downstream data delivery for queries and application codes with the GARUDA-DN approach which approximates the optimal solution by utilizing the minimum dominating set (MDS). Corresponding to a query from a sink, some or all sensor nodes of interest will respond to the query and send data to a sink through a upstream data delivery structure. This task of collecting sensor data from the sensors in the field is referred to as *data aggregation*.

In this Chapter, we consider the problem of data aggregation in environments where the data from the different sensors are *correlated* to each other. In wireless sensor networks, there are three types of data correlation as follows:

- *Spatial Correlation:*

  This refers to the correlation of data containing information about an event or phenomenon, which overlaps in the spatial domain. For example, consider the query: *What is the temperature in the region defined by the rectangle (x1,y1,x2,y2)?* Given the typical dense deployments of sensors in WSNs, it is very likely that data reported by different sensors within the rectangular region overlap in terms of the sub-area the sensors are reporting temperature for. An extreme case in the above example is a scenario where two sensors that are right next to each other report the temperature, as the data are then (almost) perfectly correlated. We refer to this special case where the data are fully correlated as having a degree of correlation, ρ, of 1.

- *Semantic Correlation:*

  This refers to the correlation of data that are reporting information, which is semantically correlated. For example, consider the query: *What is the number of cars within the field*

defined by the coordinates (x1, y1, x2, y2)? In this example, even if sensors are reporting data about *different* cars, the information is still correlated and can be aggregated[1], as the required information is merely the count of the number of cars.

- *Temporal Correlation:*

  There is also another type of correlation when there is some relationship in the information transmitted by the same sensor over time. We refer to this type of correlation as *Temporal Correlation*.

Depending on the type or the degree of correlation among data, an energy-efficient data aggregation structure can be different. In this thesis, we focus on the construction of an energy-efficient correlation-aware structure to optimize aggregation costs for scenarios when there is spatial and/or semantic correlation. We do not focus on temporal correlation, which can be addressed by coding techniques to reduce the amount of redundant data transmitted [14].

Such correlation of the data to be collected can be leveraged by appropriately fusing the data inside the network to the best extent possible, thereby reducing the delivering cost for the gathering process. In WSNs, this processing at intermediate nodes is called "in-network processing" which makes sensor networks different to the other networks, e.g., cellular or ad-hoc networks.

Hence, the specific problem we address in this thesis can be stated as: *Given that there is correlation among sensor data, how can the data gathering structure be built so as to minimize the cost of delivering data ?*

Intuitively, it is energy-efficient to construct a tree for gathering data since a tree does not have redundant links among nodes. Assuming a tree structure as a solution, it is better to aggregate data near sources so as to prevent redundant data from going through other paths. Therefore, one could design a data aggregation tree which makes source nodes favor path selection to increase early sharing of paths among sources and reduce energy consumption.

Assuming perfect aggregation (i.e., the size of aggregated data is equal to that of data before aggregation because in case of spatial correlation, data have same redundant information, or in case of semantic correlation, the level of interest for data can be kept by lossy filtering), the total

[1] We use the terms aggregation, merging, and fusion interchangeably in the rest of the thesis.

cost of delivery is the number of links on a tree. Therefore, the optimal aggregation tree is the Steiner minimal tree (SMT) which is known to be a NP-hard problem[33]. And the decision version of the Steiner minimum tree problem is a NP-complete[42]. Consequently, no polynomial time algorithm for the SMT problem is likely to exist. Although there have been many research works on approximation of the SMT problem, they require high computational complexity and centralized algorithm that generic WSNs can hardly support. As a candidate for the approximated solution of the SMT, the minimum spanning tree (MST) of which cost is at most two times of the cost of the SMT has been investigated.

Assuming no correlation among data, the optimal solution is the shortest path tree (SPT) where each source has the shortest path to the destination, sink. Since aggregation cannot reduce the size of data, it is better for each source to favor an individual shortest path.

In this thesis, we first limit the scope of problem to the perfect correlation among data since no optimal solution is known yet in case of partial correlation among data. In this context, we present a simple, scalable, and distributed approach called GARUDA-UP for approximating the *Steiner minimum tree*, and thereby achieve the potential cost benefits introduced earlier. Moreover, we can solve the upstream data delivery problem without any overhead because GARUDA-UP uses the same minimum dominating set structure, the core, which already has been constructed through GARUDA-DN's query delivery.

To aggregate perfectly correlated data in an energy-efficient way, the GARUDA-UP basically uses two structures that GARUDA-DN has constructed during downstream data delivery: (i) the minimum dominating set (MDS) which is same to the core structure proposed in Section 4.6 and (ii) the shortest path tree which is constructed through a basic flooding. The purpose of the MDS structure is to aggregate correlated data from neighboring sources; that of SPT is to gather aggregated data among core nodes in the MDS. Through theoretical analysis and simulations, we derive the delivery cost incurred by GARUDA-UP for varying conditions, and compare them with those of other structures, e.g., SPT and MST.

5.2 Motivation and Idealized Models

5.2.1 Correlation of Data

In this thesis, we consider a multi-hop WSN with one sink at the center and n sensors distributed randomly in a sensor field. The sink sends a query and k of the n sensors respond to that query. We consider the problem of efficiently aggregating the information sent by the k sensors to the sink.

Specifically, the goal is to optimize the message complexity, or the transmission cost, for the sensor data generated by the k sensors to reach the sink. It is assumed that there is correlation, defined as ρ, among the sensor data generated by the k sensors. We consider the case when there is a reasonable degree of correlation between the information collected from different sensors.

For example, let m_1 and m_2 be the amount of data generated by two sensors in response to a query. Without loss of generality, if the size of the data generated by any sensor in response to a particular query is the same, m, we have $m_1 = m_2 = m$. Now, the message size after aggregation of data from these two sources is:

$$A(m_1, m_2) = m_1 + (1-\rho) \times m_2 \tag{25}$$

where $A(m_1, m_2)$ is the amount of data after aggregating m_1 and m_2. For this case, when the information from the two sensors are perfectly correlated ($\rho = 1$), we see that the message size after aggregation is the same as the amount of data generated by the sources (m). On the other hand, if there is no correlation ($\rho = 0$) between the two sensor data, the message size after aggregation is of size $2m$.

Considering a more general case where the correlated information from $(i-1)$ sensors merges with the sensor data of the ith sensor, the message size after the merge is

$$A(m_1, \cdots, m_{i-1}, m_i) = A(m_1, \cdots, m_{i-1}) + (1-\rho) \times m_i \tag{26}$$

where $A(m_1, \cdots, m_i)$ is the amount of data after aggregating a set of nodes from m_1 to m_i. Without loss of generality, if we assume that the message size m_i of all the sensors is m, then the message size after the merge is

$$A(m_1, \cdots, m_{i-1}, m_i) \quad = \quad m + (1-\rho) \times (i-1) \times m \tag{27}$$

Thus, we take a conservative standpoint in only considering the correlation between any pair of sensor data.

The following types of correlation are within the scope of this work:

- *Spatial Correlation*: As mentioned in Section 5.1, it refers to the redundancy in the sensor data generated by the different sensors sensing the same event. In this case the degree of correlation, ρ can vary between 0 and 1 depending on the proximity of the sources among themselves and with respect to the event.

- *Semantic Correlation*: This refers to the correlation in the data generated because of the semantics of the query. As we have mentioned in Section 5.1, even though the content of the data generated by each source may not by spatially correlated, the size can still be reduced if they are semantically correlated.

  Examples of such correlation in the data generated include queries that request for average, minimum, maximum, median, first, and last.

5.2.2 Problem Statement

For our discussions, we determine the message complexity of correlation aware and unaware schemes for a given k number of sources and distribution of k sources among n sensors in a network. We assume that perfect aggregation is possible whenever data from two or more sensors merge. Given these assumptions, the objective is to minimize the *message complexity*. We define the objective function as follows:

$$MC = \sum_{i=1}^{T} m_i \tag{28}$$

where MC represents the objective function, i.e., the message complexity, T represents the total number of transmissions required for the query response from all k sensors and m_i represents the size of the message for the ith transmission. Note that each transmission may not have the same

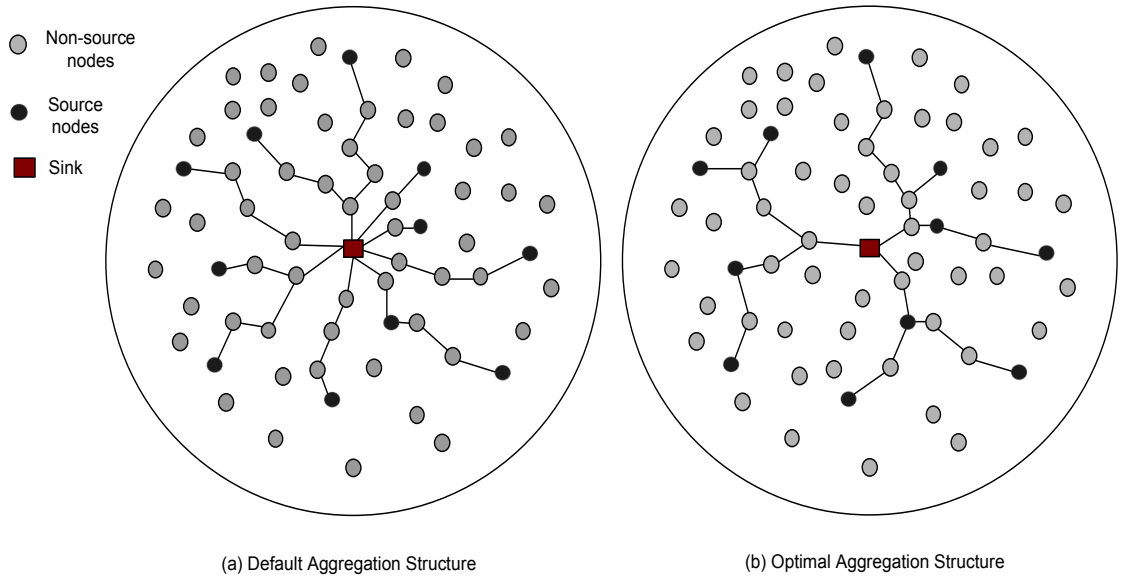


Figure 48: Example: A Typical WSN Environment

message size even if the sensor data generated in response to a query has the same size, because when two or more sensor data are aggregated at an aggregation point the amount of information generated may not be the same as the size of the original sensor data. The problem can now be formulated as "*Construct the optimal aggregation structure for receiving the query response from k of the n sensors in a sensor field, where the optimal aggregation structure is the structure that can minimize the objective function by enabling the best aggregation possible*".

To understand the problem statement, let a typical sensor network environment as shown in Figure 48. Figure 48 (a) shows a default aggregation structure that is unaware of the correlation in the sensor data generated by the sources. It is representative of most of the current sensor network routing protocols [36, 40, 71] in the upstream direction. While there might be some opportunistic aggregation possible because of the overlap of paths from different sources, these structures do not optimize for the total number of transmissions and typically have a large message complexity. If we assume perfect correlation ($\rho = 1$), the sizes of messages forwarded will be the same even if sensor data from several sources fuse. If we refer to this size of the message as m bytes, the message complexity (in bytes) for the default case is:

$$MC = m \times T = m \times 35 = 35m \tag{29}$$

where T refers to the total number of transmissions which for the figure 48(a) is 35. In contrast, if we consider the optimal aggregation structure for the same scenario as shown in Figure 48 (b), which minimizes the number of transmissions, the message complexity (in bytes) is:

$$MC = m \times T = m \times 26 = 26m \tag{30}$$

The above example clearly illustrates the potential benefit of having an optimal aggregation structure in terms of reducing the message complexity.

5.2.3 Optimal Solutions for Different Correlation Factors

In this section, we determine the optimal structures for two extreme cases: (i) zero correlation among data and (ii) perfect correlation among data. In case of zero correlation, since all data from sources have no redundant information inside them, the size of aggregated data is the sum of those of original data. However, in case of perfect correlation, since all data from sources have same kinds of information, the size of aggregated data should be equal to that of each original data.

5.2.3.1 Zero Correlation among Data

Here, we will identify the optimal structure for the zero correlation aggregation approach and determine the message complexity of that structure. The best representative structure for the zero correlation is a *Shortest Path Tree* (SPT). A typical zero correlation approach optimizes delay from every source to the sink. Due to this reason, the paths that are chosen from every source to the sink are typically the shortest path. This leads to the formation of the shortest path tree rooted at the sink. This observation that correlation unaware approaches can be represented by SPT has also been corroborated in other related works [16, 17]. From now on, we will choose SPT as representative of any correlation unaware approach.

Let us determine the message complexity of SPT in network graph. In a network graph, there are finite number of points enclosed within the disk. As before, we consider k sources in the disk with n nodes.

Proposition 1: *The message complexity for SPT in a network graph is* $O(min(k\sqrt{n}, n))$.

Proof: If we consider the n nodes to be uniformly distributed within the disk, the average

number of nodes along the radius of the network is $c \times \sqrt{n}$ [30], where c is a constant. The number of sources (num_s) along the path from a source to the sink is given by:

$$num_s = max(\frac{k}{n} \times c \times \sqrt{n}, 1) \tag{31}$$

where $\frac{k}{n}$ represents the probability of a node being a source and $c \times \sqrt{n}$ represents the number of nodes along the path. From (31), the number of paths, l, required by the k sources to reach the sink is given by:

$$\begin{aligned} l &= \frac{k}{max(\frac{k}{n} \times c \times \sqrt{n}, 1)} && (32) \\ &= min(\frac{\sqrt{n}}{c}, k) && (33) \end{aligned}$$

Since a network graph has a finite number of nodes in the network, the message complexity is represented in terms of number of transmissions. From (33), the message complexity, *MC*, in terms of the total number of transmissions is given by:

$$\begin{aligned} MC &= l \times (c \times \sqrt{n}) \\ &= min(\frac{\sqrt{n}}{c}, k) \times c \times \sqrt{n} && (34) \\ &= O(min(k\sqrt{n}, n)) && (35) \end{aligned}$$

This represents the message complexity for SPT in a network graph. ■

5.2.3.2 Perfect Correlation among Data

Now, we consider correlation aware approaches assuming perfect correlation among data from all sources.

Proposition 2: *The optimal aggregation structure for the perfect correlation with n nodes and k sources, when the sensor data from k sources are perfectly correlated ($\rho = 1$) is a network Steiner tree.*

The proof follows from definition of a network Steiner tree [105]. Let $G = (V, E, d)$ be the network graph with a vertex set V, an edge set E and distance function d. The distance function in

our environment is the edge cost, which is a function of both the message size and the distance as described in (28). When $\rho = 1$, the message size is the same even after fusion of data. So, in this case, the edge cost is a function of the distance only. As defined in [105], the network Steiner tree is the shortest tree spanning a given vertex subset within a network G. From this definition and our problem environment, the Steiner tree is the optimal aggregation structure when $\rho = 1$. ■

It has been observed in [16] that for $\rho < 1$, there is no existing optimal aggregation structure. The reason is that the message complexity now is a function of both the message size and the number of transmissions.

Although a network Steiner tree has the optimal cost for our target problem, there are no polynomial time algorithms for finding the Steiner tree in a graph [41]. Even approximation algorithms, such as the ones proposed in [39, 86], are computationally very expensive. For this reason, we consider the *Minimum Spanning Tree* (MST) as the approximated optimal solution. It has been proved in [27, 86, 94] that the cost of Euclidean Steiner tree and Euclidean MST are of the same order. It has also been shown that the cost of network Steiner tree and the network MST is also of the same order [86, 105]. From now on, we will consider the *network MST* as the optimal solution of the target problem.

We now consider the message complexity of the MST in a network graph. As before, we consider n nodes and k sources.

Proposition 3: *For $\rho = 1$, the message complexity for MST in a network graph is $O(\sqrt{k}\sqrt{n})$.*

Proof: The average number of nodes along the path from a source to the sink is $c \times \sqrt{n}$ [30], where c is a constant. From [86, 94], the maximum number of paths, l, required by the k sources for a MST is given by:

$$l = O(\sqrt{k}) \tag{36}$$

From (36), the message complexity, *MC*, given by the total number of transmissions is given by:

$$
\begin{aligned}
MC &= l \times c \times \sqrt{n} \\
&= O(\sqrt{k}) \times c \times \sqrt{n} \qquad (37) \\
&= O(\sqrt{k}\sqrt{n}) \qquad (38)
\end{aligned}
$$

The above derivation concludes the proof. ■

From the above description, we can see that the optimal aggregation structure should have a message complexity of $O(\sqrt{k}\sqrt{n})$. In general, the solution needs centralized computation and the exact location of all the k sources at the sink[27, 86]. Even the approximation algorithms for computing MST require the knowledge of the location of sources. However, it is not practical to assume that a sink knows which sensor is going to send a message *a priori*. For this reason, we conclude the need for a decentralized approach that approximates the optimal aggregation structure without the knowledge of the exact location of sources.

5.3 Related Works

In this section, we analyze some related works that have been proposed to perform aggregation for problems similar to the optimal information collection problem that has been considered in this thesis. We can categorize these works into three types: (i) default aggregation in WSNs, (ii) intelligent aggregation in WSNs, and (iii) graph theory based approaches.

5.3.1 Default Aggregation in WSNs

[36] is a data-centric routing framework for gathering information from the sensors to the sink in a WSN. While, it is possible that aggregation can happen opportunistically due to any overlapping paths from the sources to sink, it may not be efficient because of following two reasons: (i) the structure from the sensors to the sink does not approximate a Steiner tree or a MST; and (ii) the nodes that are incidentally chosen as aggregation points do not have any notion of the amount of time to wait before it can aggregate the data from all sensors downstream of it efficiently. While, [104] addresses the second problem to a certain extent, it is still of concern that the proposed structure

may not be optimal. [50, 65] are other aggregation structures proposed in the context of sensors-to-sink communication in WSNs. However, they are not proposed in the context of aggregation of information for correlated sensor data and are not efficient for the problem considered in this work.

5.3.2 Intelligent Aggregation in WSNs

There have been a couple of works that have been proposed to do explicit aggregation in the context of sensor networks. [16, 17] propose a simplified information model and try to solve the problem of aggregating correlated data with two simple heuristics. They consider a correlation model similar to the one considered in this paper and propose two simple heuristics for a given correlation factor ($0 < \rho < 1$): (1) Leaves deletion heuristic: Starting from a SPT, each source node does a local search to find a neighboring source node such that the cost of sending information first to the neighbor and then to the sink is smaller than the cost of both nodes to send their information to a sink separately. After a certain number of iterations, the resulting spanning tree is considered as a good approximation of the optimum minimum cost spanning tree; and (2)Balanced SPT/multiple traveling salesman problem (TSP) tree: This solution is based on the assumption that the optimum solution should be a combination of partial SPT, and partial TSP. They find the SPT for nodes within a certain distance to the sink and successively add nodes to the tree such that the message cost is minimized. This is a simple suboptimal nearest neighbor approximation of the multiple TSP. However, for both solutions, *a priori* knowledge of location of sources is assumed. Furthermore, due to the computation cost of tree construction, both approaches are more suitable for continues information collection rather than a one-shot collection process.

[2] assumes a more generic cost function $f(x)$ given x sources. Assuming that $f(x)$ is a concave non-decreasing function and $f(0) = 0$, it argues that there exists an information collection structure that is good for all canonical concave cost functions f. For this sub-optimal solution, it shows that the cost is at most $1 + log(k)$ times the optimal cost, where k is the number of sources. However, this work does not identify or construct the optimal tree for a specific cost function. This algorithm gives a good approximation for a large class of cost functions in the context of WSNs. However, the algorithm is centralized in nature, and assumes the knowledge of the exact location of sources. For this reason, this solution can only be useful for offline computation.

[35] proposes a simple modification to the directed diffusion to come up with a structure that encourages aggregation. Briefly, it uses a greedy incremental tree to greedily aggregate the information from the different sources where the sources try to reach the aggregation tree that is already constructed in the least number of hops. This is in stark contrast to directed diffusion, where the gradients are set in the direction of the sink and aggregation happens only in a opportunistic fashion. However, this approach is only a heuristic and does not guarantee any cost bounds in relation to the optimal solution for the problem. Also, in pathological scenarios where sources are uniformly distributed, the solution may not be able to aggregate efficiently.

5.3.3 Graph Theory Techniques

In [39], an information flow model is considered, where a single server sends a data item to a set of clients requesting this data. The distribution and number of clients is not known *a priori*. However, each client in a network will choose to contact the server independently with some probability p_i. Using properties of concave cost functions, the authors prove that the optimal solution is a special case of the Steiner minimum tree, which they call as a maybecast tree. A hub and spoke model is proposed as an approximation of the optimal maybecast tree.The paper uses existing heuristics for the Steiner minimum tree construction can be used to solve the problem of connecting the hubs to the root with minimum cost. This approach is also centralized as the clustering and the determination of hubs are all done in a centralized fashion. [94, 105] present a set of heuristics to approximate the optimality of the Steiner minimum tree. However, these approaches are still centralized and cannot be realized in the context of WSNs.

5.4 Design Goals and Key Idea

In this section, we present the design goals and key elements of a proposed upstream correlated data aggregation scheme.

5.4.1 Problem Scopes and Goals

In real life, the correlation factor ρ can be between 0 (no correlation among data) and 1 (perfect correlation) depending on the location of data or semantics of query. To the best of our knowledge,

a theoretically optimal solution for general data aggregation problem assuming correlation factor between 0 and 1 has not been identified yet.

In the thesis, we limit the correlation factor to 1 when all data are perfectly correlated with each other. As discussed before, the optimal solution is the Steiner minimum tree (SMT) known to be a NP-hard problem[42].

Although there have been many previous works in [33] on the approximation of the SMT, those schemes still require computational and communication overheads that WSNs cannot support. In this thesis, we design an aggregation structure that approximate the optimal solution in a distributed fashion with less amount of overhead than distributed approximation of the SMT.

The following are the key goals that the design of our proposed data aggregation strategy is based on:

- **Perfect Correlation:**

  As discussed, this thesis focuses on the aggregation problem assuming all data from sensors are perfectly correlated. Therefore, the amount of aggregated data is equal to the amount of original data before aggregation.

- **Efficiency:**

  Since the energy conservation is the critical issue in WSNs, the goal of design is to minimize the energy consumption at data aggregation. To minimize the energy consumption, it is better to reduce redundancy among data while data are delivered. Therefore, the proposed scheme will aggregate correlated data as soon as and as much as possible to reduce redundancy.

- **Scalability:**

  In general, WSNs might have more than tens of thousands sensors. The proposed scheme should be operated efficiently with reasonable amount of overhead linearly increasing to the scale of WSNs.

- **Decentralization:**

  Since using global information in a distributed environment such as a sensor network can incur high overheads, the proposed scheme should use purely local information in its approach. Then it will be operated in a decentralized fashion over large scale of WSNs.

- **Loose Synchronization:**

  To minimize the cost of aggregation, most of theoretical solutions use tree structures, e.g., the shortest path tree, the minimum spanning tree and the Steiner minimum tree. Although these tree structures reduce the redundancy among data, they also requires synchronization among nodes that transmit, aggregate or forward data. However, since the synchronization is also one of hard problems in WSNs, the proposed scheme will relax the degree of synchronization so that it can be operated without assumption of other synchronization algorithms.

- **Mobility and Node Failures:**

  The dynamic change of network topology due to mobility and node failures makes aggregation schemes in WSNs inefficient and even more out of service. Therefore, the proposed scheme will address this problem by constructing a aggregation structure dynamically and instantaneously.

5.4.2 Key Ideas

5.4.2.1 General Heuristics for Steiner minimum tree

Although there have been many research works on the approximation algorithms for the Steiner minimum tree (SMT) which is the optimal solution in case of the perfect correlation case, most of works assume centralized coordination which WSNs can hardly support. To approximate the SMT solution in distributed environments, several heuristics have been investigated[33]:

- Shortest Path Heuristics

  They start from any source and expand the tree until it spans all sources. This expansion is typically based on the addition of the shortest paths between a source in the tree and another source not yet in the tree[86].

- Tree Heuristics

  Another class of heuristics is based on the idea of constructing a tree spanning all sources. Usually a variant of the minimum spanning tree algorithm is used to obtain this initial tree. Then various strategies to improve the initial tree can be applied[86].

- Vertex Heuristics

The major difficulty when solving the Steiner minimum tree problem is to identify non-sources that belong to the SMT in order to connect sources in the SMT. Once given, the SMT can be found easily; it is a minimum spanning tree for the subnetwork induced by the sources and selected non-sources. The general idea behind vertex heuristics is to identify "good" non-sources.

5.4.2.2 Heuristics in GARUDA-UP

From the definition of the Steiner minimum tree (SMT), we need to find an additional set of nodes that are not sources and inserted into the SMT in order to achieve the shortest connectivity. In graph theory, this set is called "Steiner points". Therefore, one of the above heuristics also tries to find these Steiner points. However, since these Steiner points depend on the locations of sources, we need to find the optimal set of Steiner points after we know the exact locations of sources. Again, we are confronted with the same situation as that when we solve the minimum set cover problem for the loss recovery server designation in Section 4.5.

Instead of solving the SMT problem of which optimal solutions are different to each other based on given set of sources, we address it with the minimum dominating set (MDS) problem of which optimal solution is not changed irrespective of given set of sources.

Assuming perfect correlation among all data, it is well known that the early aggregation around sources is to reduce redundant data in tree structures. And we can utilize the above heuristic using the MDS approach. Each node in MDS can work as a Steiner point if it has any neighboring sources around it.

Furthermore, we already have the simple and decentralized solution, the core, for the MDS problem in reliable downstream data delivery discussed in Chapter 4. One of major applications in reliable downstream data delivery is a query dissemination that is tightly coupled with data gathering problem. Therefore, after a query flooding constructs the core structure, data aggregation can use the core to find the set of Steiner points which aggregate data from neighboring sources.

Then the data at some core nodes can be forwarded to its upstream core locating at inside core band since the core structure has the shortest path information toward a sink. Eventually, all data from core nodes will reach a sink through the shortest path that was constructed while a query was

flooded.

Although there is a gap between the optimal solution of the Steiner minimum tree and the approximated solution using the minimum dominating set, the proposed MDS approach can obtain a promising result compared to other approximations that assume centralized coordination and high computational complexity. In Section 5.5, we will discuss the optimality of the proposed scheme, GARUDA-UP.

5.5 GARUDA-UP Design

In this section, we present a high level overview of the GARUDA-UP approach and the design elements in detail. We also derive the message complexity of the proposed scheme and show that it has a comparable order with the optimal solution. Only for the presentation, we assume a circular network field with *n* sensor nodes randomly distributed in the field. We assume that each sensor knows only the identification information of neighboring sensors without any global coordination. By default, the location information of source nodes is not known and the information from all sensor nodes is perfectly correlated to each other($\rho = 1$).

5.5.1 Overview

There are three key elements in the design of the GARUDA-UP approach: (i) Designation of aggregation nodes: approximate the minimum dominating set through the core construction procedure discussed in Section 4.7, and guarantee that every node should know its aggregation node around itself; (ii) Data gathering at each core node: first allows sources not in the core set to transmit data based on the contention based scheduling; (iii) Aggregation among core nodes: next allows nodes in the core set to transmit aggregated data from sources to any node in the inside core band so that aggregated data can reach a sink eventually through the shortest path to a sink; and (iv) Synchronization: bounds nodes in the core set to the time constraint based on band-id (the number of hops from a sink) so that data between core nodes can be aggregated efficiently. We will describe the details of each of the elements and the rationale behind the realization of these elements in the sections below.

5.5.2 Construction of the Core Set

The purpose of constructing the core set C is to find the MDS of which size is minimum enough to cover all nodes in set N in a network graph $G = \{N, E\}$. Since nodes in the MDS act as the Steiner points to aggregate data, it is better to minimize the number $|C| = n_c$ of nodes in the MDS.

As mentioned before, GARUDA-UP uses the same procedure to construct the core set in order to approximate the minimum dominating set in a network(see Section 4.7) without any overhead of constructing the core structure for upstream correlated data delivery.

5.5.3 Aggregation at a Core Node

Once the core set C is determined by the instantaneous construction procedure, each non-core node nc_i not in set C should know its core node at core bands or non-core at core bands. If a node does not have any neighboring node at core bands, it declares itself as a leaf node which exceptionally will send data up to a core node in a inner core band through the shortest path tree.

If a source node not in set C wants to transmit data, it sends data to a core node in set C or a neighboring non-core node in core bands so that data can be aggregated at a core node in core bands. Since a core node acts as a Steiner point, the MDS aggregates data with minimum number of Steiner points so that early aggregation can reduce redundant data as early as possible.

5.5.4 Aggregation among Core Nodes

While constructing the core structure, GARUDA-UP also has the shortest path tree rooted at a sink. Basically, every node in a network has its precedent node in the shortest path tree so that all data can be forwarded toward a sink. Therefore, GARUDA-UP does not require any explicit routing scheme that requires overhead to construct because it uses the shortest path tree, the by-product of a query flooding.

After core nodes aggregate data from neighboring non-core nodes, they send aggregated data to a core node in an inner core band through the shortest path tree. Instead of reaching a sink directly from each core node, data will be forwarded to a core node at inner core band to prevent aggregated data from selecting individual paths to a sink. Therefore, each core node transmits aggregated data to another core node in inner core band through at most three hops.

5.5.5 Synchronization

In general, data aggregation trees, e.g., the Steiner minimum tree, the shortest path tree and the minimum spanning tree, require synchronization for data transmission between two nodes sharing a link in a tree. Each node in a tree should not transmit data until it receives all data from all child nodes. In practical systems, it is necessary to ensure that these nodes wait for an optimum delay value to ensure perfect aggregation of the source data at the Steiner points. If a node waits for a value less than this optimum delay, it will not be able to aggregate the data from all the sources downstream of it. This will lead to inefficient aggregation and consequently increase the message complexity of aggregation. Therefore, most of aggregation schemes require synchronization for data transmission of all nodes in an aggregation tree.

GARUDA-UP, however, does not require the above tight synchronization that all nodes in a tree should follow. Instead, GARUDA-UP needs a loose synchronization that only core nodes in a tree should follow. The other non-core are not required to follow the loose synchronization since they are located at leaf nodes of a tree. Practically, we can implement the loose synchronization between two core nodes at different core bands by an easy way. For example, we can synchronize those two core nodes with band-id (hop distance from a sink to a node). Core nodes with lower band-id wait for longer time than other nodes with higher band-id.

5.5.6 Message Complexity

In this section, we derive the message complexity of the proposed scheme, GARUDA-UP, for the case when there is perfect correlation ($\rho = 1$) among all data generated by sources.

Given a network graph $G = \{N, E\}$, we assume that there is a minimum dominating set C of which size is n_c. In worst case, all k sources are located at non-core nodes so that each source should transmit its data at least one time. And all core nodes should transmit the aggregated data up to another core node at an inner band by at most three transmissions because any pair of two core nodes has a path consisting of at most two non-core nodes. Therefore, total message complexity is as follows:

$$MC = k + 3n_c, \tag{39}$$

where k is the number of sources and n_c is the cardinality of MDS C.

There are several upper bounds for the cost of the MDS as a function of the number of nodes n. [5] proves that $n_c \leq n\frac{1+\ln(D_G+1)}{D_G+1}$, where D_G is the minimum degree of a network. However, to the best of our knowledge, there is not known yet the upper bound for the cost of the MDS, which is comparable to a logarithmic or square root function. Therefore, we will compare the message complexity of GARUDA-UP with those of other two theoretical approaches, SPT and MST, using extensive simulations in Section 5.7.

5.6 GARUDA-UP Framework

In this section, we will present the GARUDA-UP approach in detail and describe the functionalities performed by the sink, sources and the core node acting as the Steiner point (We will use those two terminologies, Steiner point and core node interchangeably in the rest of this Chapter.). Basically a source can be one of non-core, core and leaf nodes based on constructed core structure.

For the data aggregation, GARUDA-UP uses two stages: (i) at stage 1, sources only at non-core nodes including leaf nodes transmit data; and (ii) at stage 2, core nodes transmit data to another core nodes. Those two stage will occur separately.

5.6.1 Core Construction

As proposed, GARUDA-UP uses the same core construction procedure of GARUDA except one minor rule about core the solicitation message from non-core node in Section 4.7.2. Figure 49 shows the instant result for core construction by disseminating a query through a network. Basically, all nodes can access a precedent in the shortest path tree rooted at a sink. Based on this core structure, a node in a network should be one of core nodes, non-core nodes, or leaf nodes as follows:

- A core node is a node at a core band of which band-id[2]is $3i$. Two core nodes in the same core band should have at least two-hop distance between each other to reduce the total number of core nodes. A core node also keeps the information of a precedent in the shortest path tree

[2]In the thesis, the band-id means the shortest hop distance from a sink to a node in a network.

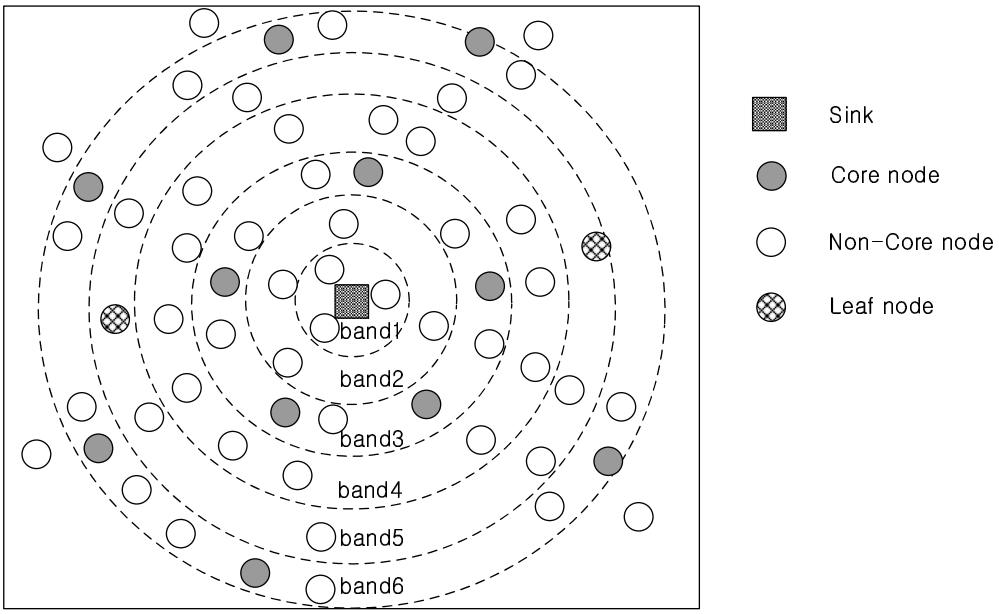


Figure 49: Instantaneous Core Construction in GARUDA-UP

root at a sink so that the core at $3i$ band can transmit the data to another core node at inner core $3(i-1)$ band eventually.

- All nodes at non-core bands $3i+1$ or $3i-1$ should be a non-core node. And some nodes at core band $3i$ might become a non-core node based on the core construction procedure. All non-core nodes should access two nodes: its core node at $3i$ band and its precedent in the SPT, of which band-id is less than its band-id. Some non-core nodes at $3i+1$ or $3i-1$ band cannot have a neighboring core node at $3i$ band. In this case, they can still access a core node at $3i$ band through its neighboring non-core node at $3i$ band indirectly.

- For exceptional cases, some non-core nodes of which band-id is $3i+2$ cannot have any neighboring nodes located at core band $3i+3$. These non-core nodes declare themselves as a leaf node. Then they always transmit data to a precedent that is a non-core node at inner band $3i+1$.

5.6.2 Stage 1: Original Data Transmission

We assume that all nodes know the start time of data transmission for each query.

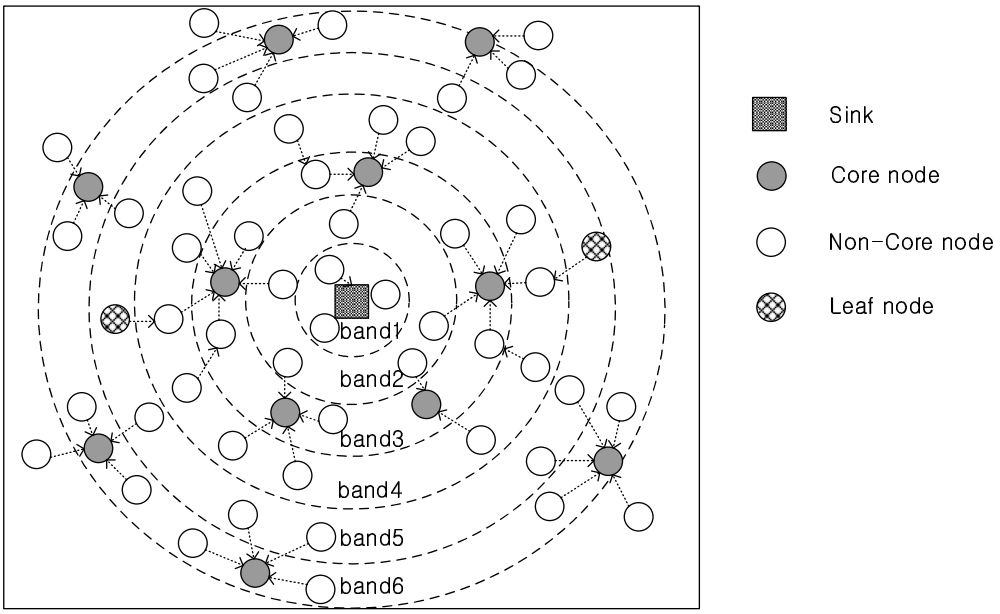


Figure 50: Stage 1: Original Data Transmission in GARUDA-UP

5.6.2.1 Non-core Nodes

If a non-core node at $3i-1$ or $3i+1$ band is a source node, it will transmit data to its core at core bands after a delay δ_{nc}[3]. If the receiving node at core band $3i$ does not declare itself as a core node, it will forward the data to its core node at the same core band $3i$. We use a contention-free medium access control scheme to coordinate all non-core sources around a core node based on the number of non-core nodes around the core node. In Figure 50, all non-core nodes, white circles, send data to core nodes, gray circles. Between different groups around each core node, we don't need to consider scheduling because they are separated with each other at least two-hop distance.

5.6.2.2 Leaf Nodes

If a leaf node at $3i+2$ band is a source node, it will transmit data immediately to its neighboring non-core node at $3i+1$ bands so that the neighboring non-core node can receive the data successfully before it sends its own data. In Figure 50, leaf nodes at band 5, checked circles, send data to

[3]The delay δ_{nc} is set based on the maximum number of leaf nodes around a non-core node so that leaf nodes can transmit data successfully to a non-core node within δ_{nc}.

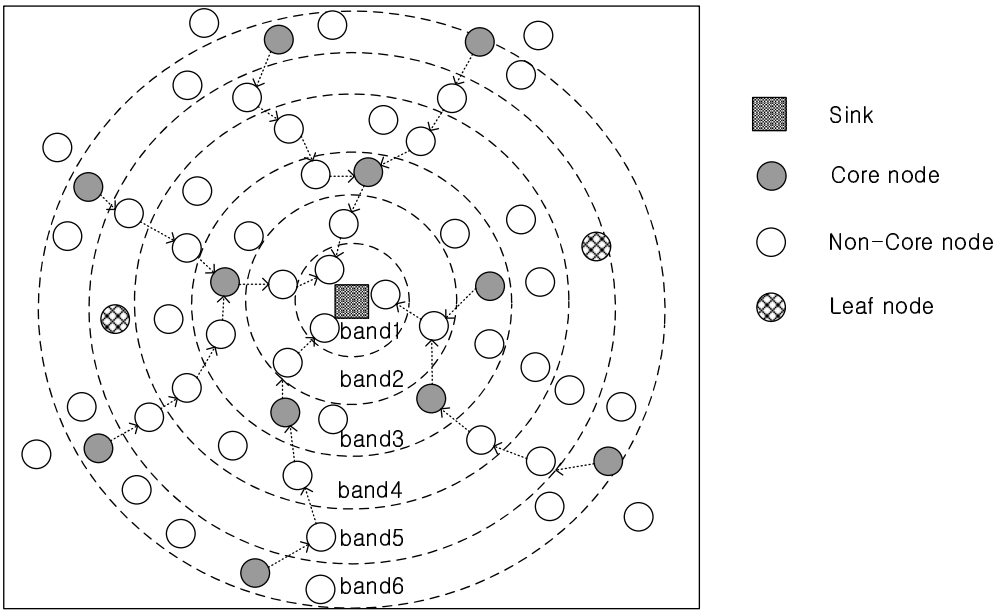


Figure 51: Stage 2: Aggregated Data Transmission in GARUDA-UP

non-core nodes at band 4.

5.6.2.3 Core Nodes

If a core node at core bands is not a source node, it does not need to transmit data unless it receives any data from its non-core nodes or core nodes at outer core band. Although the core node has data to send, it will wait for a time δ_c[4] so that it can wait and aggregate its own data with incoming data from other core nodes that are located at outer bands.

5.6.3 Stage 2: Aggregated Data Transmission

After stage 1, we assume that all data from non-core nodes are received by core nodes and aggregated with other data. The remaining procedure is to deliver the aggregated data to a sink. To deliver these aggregated data, GARUDA-UP uses the shortest path tree that was constructed during the corresponding query flooding. Figure 51 shows delivery paths between core nodes at different core bands. Compared to the original shortest path tree, the paths have some differences. Instead of reaching a sink directly using the SPT, it is better to reach another core node at inner band since

[4]The delay δ_c is set inverse proportionally to the band-id.

it can reduce redundancy among other aggregated data. Whenever a non-core node at core bands receives aggregated data from other core nodes at outer bands, it will forward them to its core node at the same core band.

5.6.4 Other Considerations

Since wireless sensor networks have constraints: scarce energy and frequent node failures, etc, the proposed scheme needs to address these constraints. Especially, GARUDA-UP tackles the unbalanced energy consumption by proposing a load balancing method and a node-failures problem by proposing a backup mechanism.

5.6.4.1 Load Balancing

A load balancing scheme is important to ensure that the resources of all non-source nodes are utilized to roughly the same extent over a period of time. This is done by making sure that the same set of Steiner nodes are not selected for multiple queries sent by the sink. In GARUDA-UP, nodes at core band $3i$ can consume more amount of energy than other nodes at non-core bands $3i+1$ or $3i+2$ because some of them should be selected as core nodes which should act as an aggregating point as well as a source. To solve this load balancing problem, GARUDA-UP uses a method to shift the bands during the construction of core structure. Instead of selecting core nodes at $3i$ bands, GARUDA-UP can select them at $3i+1$ or $3i+2$ bands by shifting the core band. The other non-core nodes will be selected at one-hop inner nodes and one-hop outer band around the shifted core bands. The remaining procedure will be same. Therefore, GARUDA-UP can balance the load through other bands with this shifting method.

5.6.4.2 Node Failures

We discuss the impact of node failures on the correctness and optimality of the proposed scheme. Basically, since each query constructs a new core structure through flooding, node-failures before query flooding are discarded in this consideration. We consider node failures after core construction at all possible sensor nodes:

- If a non-core node at $3i-1$ or $3i+1$ fails, and it is one of nodes in a path between two core nodes, then a core node at outer core band cannot reach another core node at inner core band.

Since all data are transmitted through a unicast communication between nodes, a sender node can detect the non-core node's failure. In this failure, the sender will broadcast a solicitation message only to one-hop neighbors at inner band. If any non-core replies to this message, the sender finds an alternative path to inner core bands.

- If a non-core node at $3i$ band fails, and it is one of nodes in a path between two core nodes, then the above problem can happen. If a sender at $3i+1$ in a path detects the non-core's failure, it also broadcast a solicitation message to one-hop neighbors at inner band $3i$. And the remaining procedure is the same to the above procedure.

- If a core node at $3i$ band fails, and any sender tries to transmit data to the failed core, it can detect the core's failure. In this case, the sender will broadcast a solicitation message for core to one-hop neighbors in the same core band $3i$. If any node at $3i$ band replies to this message and declares itself as a new core. Then the other non-core nodes around the new core node will update their core information. Especially, the node that was dominated to the failed core node will have high priority for reply with shorter delay to respond.

5.7 Performance Evaluation

This section evaluates the performance of the GARUDA-UP approach that uses a decentralized and simple scheme under different network configurations. To do the evaluation systematically, we compare it with two schemes: the shortest path tree (SPT) constructed in a decentralized fashion and the minimum spanning tree (MST) constructed in a centralized fashion with high computational complexity. We vary the node density, source density, source distribution; and compare the performance of the three schemes.

5.7.1 Simulation Environments

- We assume a typical one-shot query-response model in sensor networks. In this model, a sink broadcasts a query to the entire network and sensors that have corresponding information will reply with one message. Notice that GARUDA-UP is also applicable for continuous

query-response model, but one-shot model is where we get the best improvement over other existing approaches, therefore we will concentrate on this model in our simulation. In terms of message size, we assume that every source sends one message of the same size, but the specific length of the message does not matter.

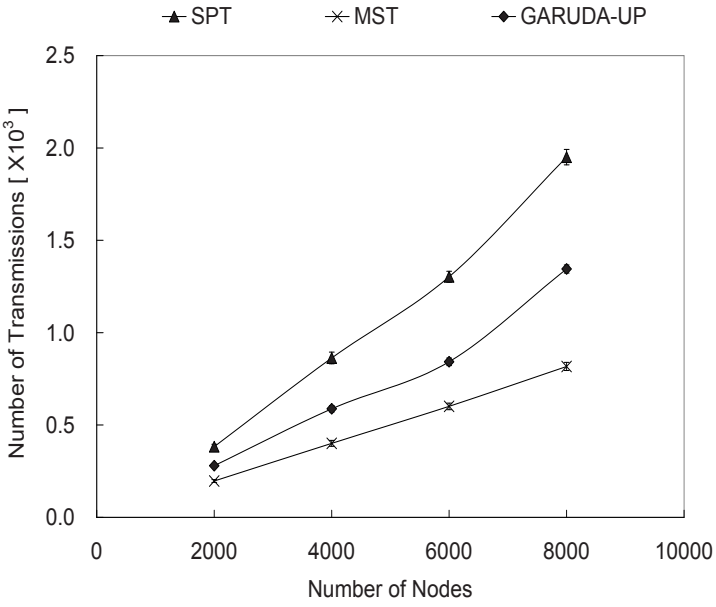
- We use a discrete event simulator for all evaluations. And the simulation topologies are largely similar to that used in general sensor networks: 2000 to 8000 nodes uniformly distributed within a circular field of radius 400m. The number of sources that generate messages for one specific query varies from $\frac{1}{10}$, $\frac{1}{6}$, $\frac{1}{4}$ to $\frac{1}{2}$ of the total number of nodes in the network.

- We compare GARUDA-UP with SPT since most of the current routing protocols in the context of WSNs such as Directed Diffusion and GPSR try to approximate the message complexity of SPT. And we are interested in how GARUDA-UP performs better compared with the centralized algorithm. We also compare it with MST, which represents the optimal solution in the target environment. Ideally, we should have compared it with the Steiner minimum tree. But as we mentioned before, the computation overhead is very high, especially we are considering thousands of nodes, the time it takes to generate even one sample is prohibitive. For this reason, we use MST to approximate Steiner Tree performance which has the same message complexity order ($O(\sqrt{k})$) and a competitive cost ratio of less than $\frac{1}{2}$ as that of Steiner minimum tree, but a much less computation cost. We generate SPT with Dijkstra's algorithm, and MST with Prim's algorithm.

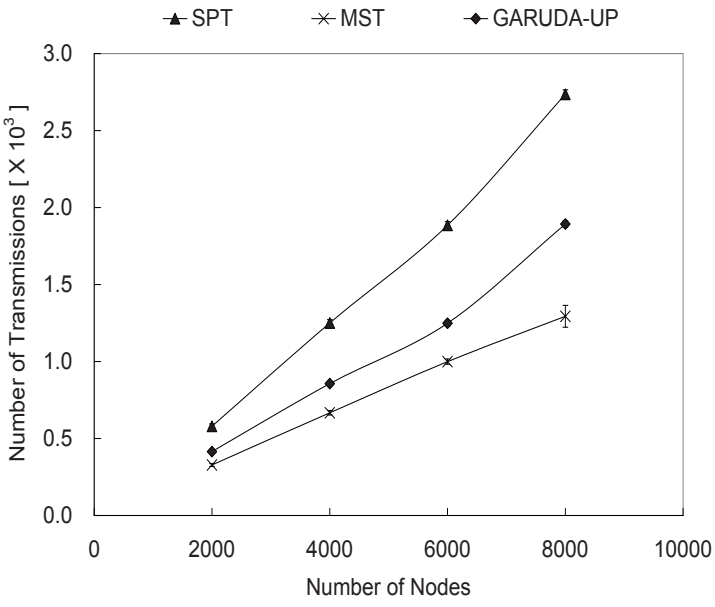
- We evaluate the GARUDA-UP approach using message complexity that is equal to the total cost of data aggregation. For message complexity, we measure the total number of transmissions required for all responses to reach the sink.

- To focus on the comparison of aggregation efficiency of different structures, we assume a perfect Media Access Control (MAC) layer that avoids collisions for all approaches.

- All the simulation results are derived after averaging results over 10 random seeds and are presented within 95% confidence intervals.

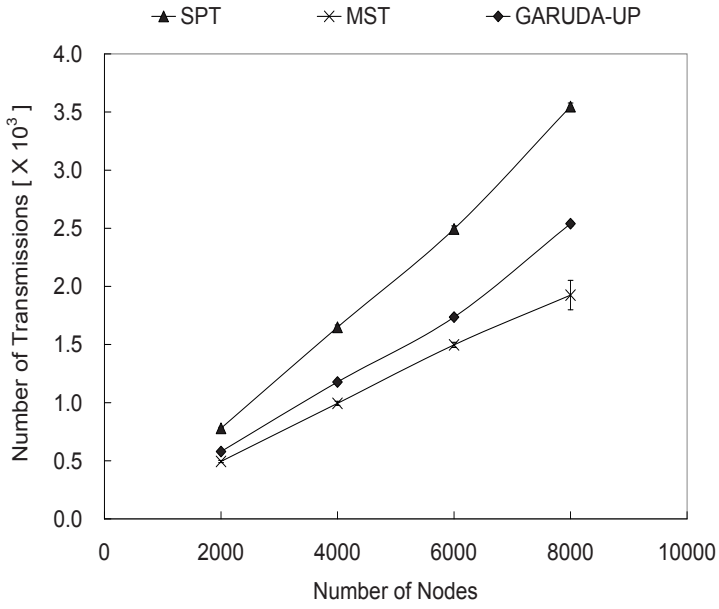


(a) Number of Sources $k = \frac{n}{10}$

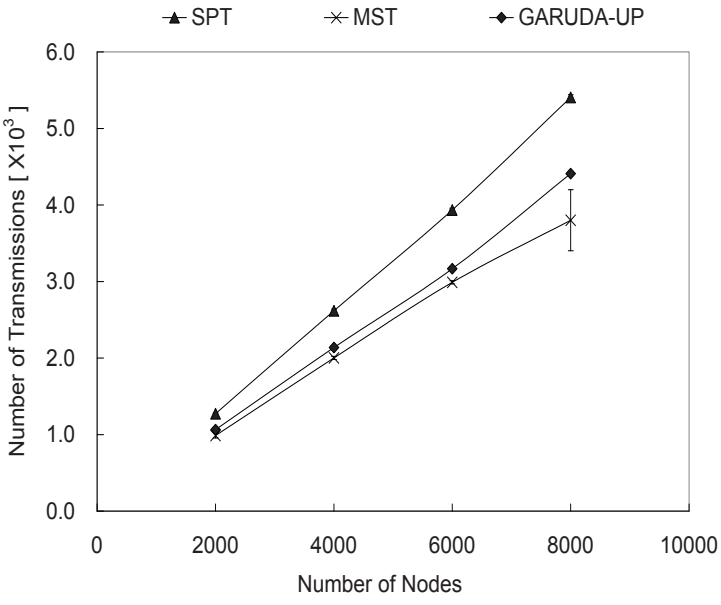


(b) Number of Sources $k = \frac{n}{6}$

Figure 52: Performance Comparison among SPT, MST, and GARUDA-UP for Varying Number of Nodes and Fixing the Ratio of Number of Nodes to that of Sources to 10 and 6



(a) Number of Sources $k = \frac{n}{4}$



(b) Number of Sources $k = \frac{n}{2}$

Figure 53: Performance Comparison among SPT, MST, and GARUDA-UP for Varying Number of Nodes and Fixing the Ratio of Number of Nodes to that of Sources to 4 and 2

5.7.2 Different Node Densities

We first compare the performance of the decentralized GARUDA-UP with those of the SPT and the MST. In this scenario, we assume that data from all sources are correlated perfectly ($\rho = 1$).

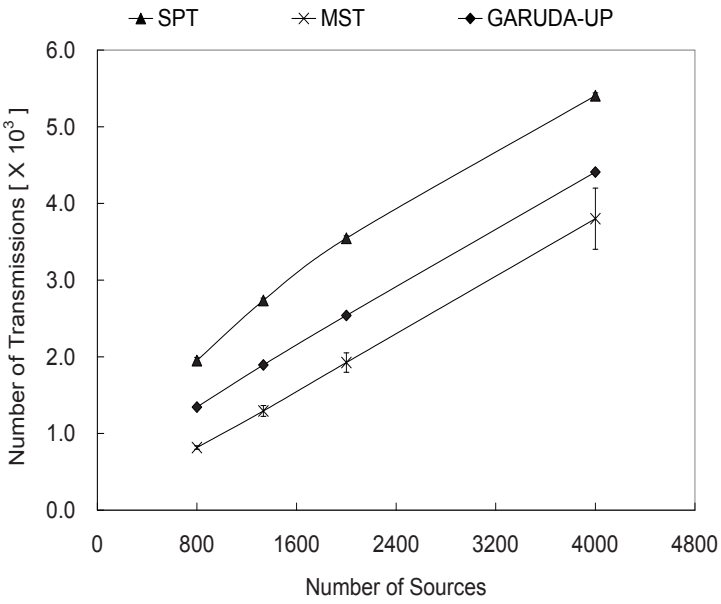
Figure 52 and Figure 53 show the cost of three schemes as a function of number of nodes for different number of sources k. In these simulations, we choose the total number of nodes n as 2000, 4000, 6000, and 8000; and the number of sources k as $\frac{n}{10}$, $\frac{n}{6}$, $\frac{n}{4}$, and $\frac{n}{2}$, respectively. To compare the efficiency of those three schemes, we measure the message complexity, the number of total transmission during aggregation for different schemes.

It can be seen that GARUDA-UP outperforms the SPT scheme under all situations. From the results, we can observe that GARUDA-UP reduces the message complexity from 16% to 35% compared with the SPT with small amount of overhead for constructing the core structure without centralized coordination. Although the MST uses the centralized coordination with high computational complexity, it only can reduce the message complexity from 22% to 50% compared with the SPT. Therefore, from the simulation results, we can say that GARUDA-UP is a good decentralized approximation to the MST.

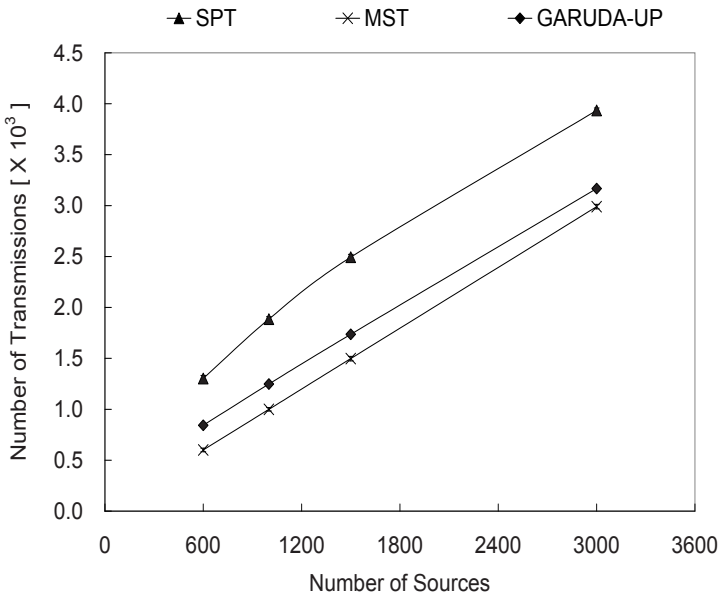
We can also see that the cost of the SPT increases faster than that of the GARUDA-UP approach as the number of nodes increases. This is expected since more number of nodes reduce the efficiency of aggregation in the SPT as the paths chosen by different sources are less likely to overlap. Therefore, GARUDA-UP can be considered as a more scalable decentralized approach as the number of nodes increases. Furthermore, it is observed that the difference between two schemes increases as the ratio of the number of sources to the number of nodes, $\frac{k}{n}$, decreases because more number of sources increase the probability of aggregation for the SPT. As the ratio $\frac{k}{n}$ goes to 1, message complexities of three schemes converge into n transmissions.

5.7.3 Different Source Densities

In Figure 54 and Figure 55, we compare the cost of three schemes: SPT, MST and GARUDA-UP, as a function of number of sources k for different number of nodes n. We also can see that GARUDA-UP still outperforms the SPT and approaches the MST. Based on our analysis of the message complexities of SPT and MST, we expect the difference in costs of the two approaches

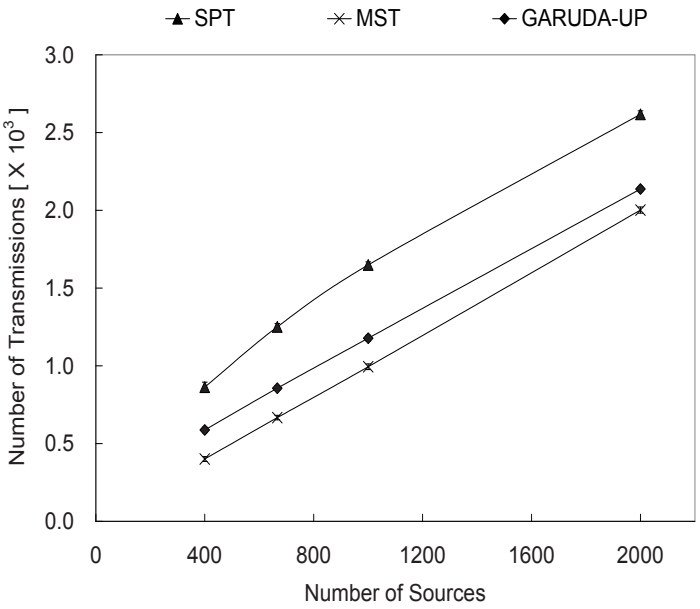


(a) Number of Nodes $n = 8000$

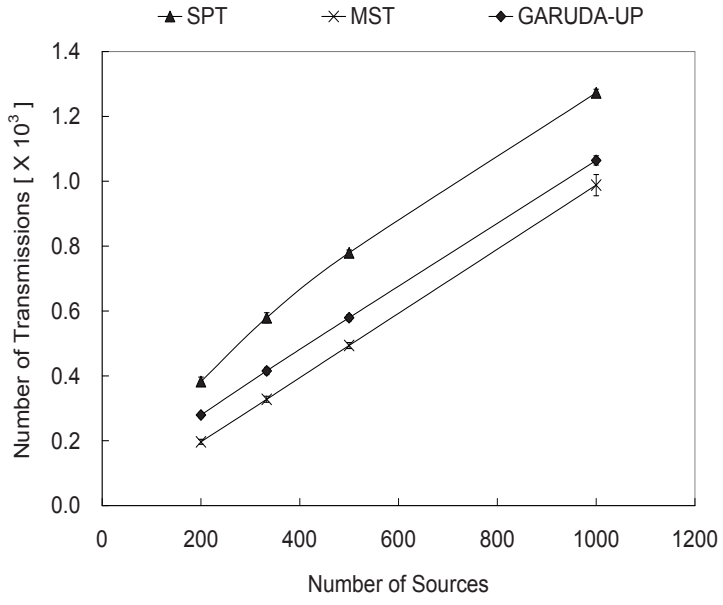


(b) Number of Nodes $n = 6000$

Figure 54: Performance Comparison among SPT, MST, and GARUDA-UP for Varying Number of Sources and Fixing Number of Nodes to 8000 and 6000



(a) Number of Nodes $n = 4000$



(b) Number of Nodes $n = 2000$

Figure 55: Performance Comparison among SPT, MST, and GARUDA-UP for Varying Number of Sources and Fixing Number of Nodes to 4000 and 2000

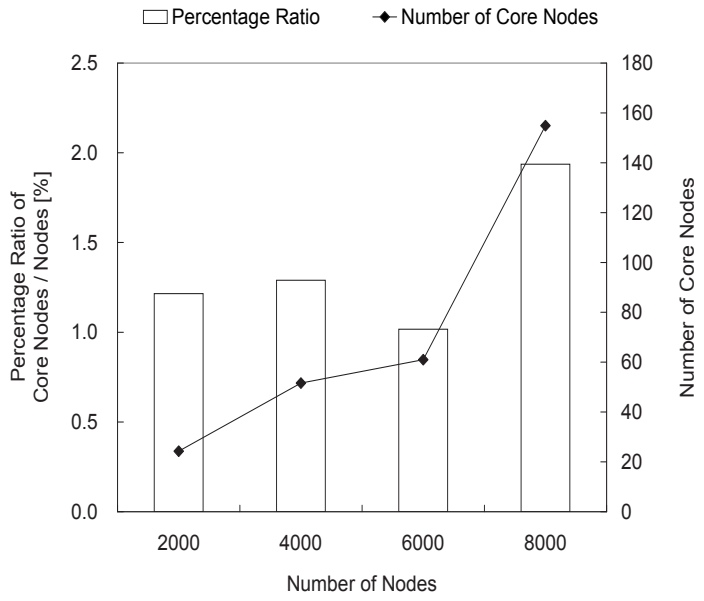


Figure 56: Percentage Ratio of the Number of Core Nodes to the Number of Nodes in GARUDA-UP Simulations

to increase up to a certain k and converge to 0 as $k \to n$. Figure 54 and Figure 55 show that the maximum difference occurs when $k = \frac{n}{4}$. When k is larger than $\frac{n}{4}$ and approaches n, the costs of both schemes will converge to n because each node aggregates all the downstream data and transmits exactly once.

5.7.4 Number of Core Nodes

To analyze the reasons of GARUDA-UP's outperforming the SPT and approximating to the MST, we observe the number of core nodes of GARUDA-UP. Among two parameters on GARUDA-UP's message complexity: the number of source nodes k and the number of core nodes n_c, the number of core nodes n_c is the only parameter that changes the message complexity of GARUDA-UP since k is a given value. Figure 56 shows the number of core nodes selected in simulations varying the number of nodes n from 2000 to 8000. It shows that the percentage ratio of $\frac{n_c}{k}$ is below 2% over all scenarios from 2,000 to 8,000 nodes and less than the number of sources k. In the simulations, we assume that the number of sources is larger than 10% of nodes. Therefore, in message complexity of GARUDA-UP, a dominating factor is the number of sources k of which message complexity order is lower than the order of the SPT, $O(min(k\sqrt{n}, n))$ (see Section 5.2).

5.8 Summary

In this section we compare the proposed GARUDA-UP scheme with the decentralized SPT and the centralized MST schemes. Simulation results indicate that GARUDA-UP outperforms the SPT substantially in terms of delivery cost for all environments. The cardinal reason for GARUDA-UP's superior performance over that of the SPT is because GARUDA-UP approximates the MST using heuristics to choose the Steiner points from the minimum dominating set.

CHAPTER VI

CONCLUSION

This thesis has addressed the energy efficiency in wireless sensor networks at three different layers: (i) topology control layer that decides transmission power level to construct efficient network topology in terms of energy consumption as well as throughput performance; (ii) downstream data delivery that brings loss-sensitive data to areas of interest reliably and efficiently in terms of latency and energy consumption; and (iii) upstream correlated data delivery that gathers redundant information with minimum energy consumption in terms of data transmission.

6.1 Contributions

Topology Control:

- Investigated the optimal topology in terms of energy consumption and throughput; and verified that the static topology control using minimum transmission power cannot always maximize the throughput per unit energy under all kinds of environment in typical wireless sensor networks through extensive simulations.
- Presented a theoretical throughput model representing the relationship between throughput and transmission power to motivate an adaptive topology control scheme based on different environments, e.g., traffic load and node density.
- Proposed three adaptive topology control schemes: (i) ATC-CP which synchronizes all nodes to use a common power to support symmetric MAC and routing algorithms; (ii) ATC-IP which allows each node to use an individual power to eliminate coordination overheads; and (iii) ATC-MS which harmonizes transmission power of nodes within local areas to adapt power to the optimal power in the area with the minimum coordination overhead.

- Compared the performance of three adaptive topology control schemes with the static topology control schemes using minimum and maximum transmission powers; and showed that ATC-MS outperforms the others.

Reliable Downstream Data Delivery:

- Motivated the reliability of downstream data delivery through simulations and identified different delivery semantics
- Formulated the reliable data delivery problem theoretically using the minimum set cover problem and transformed it to the minimum dominating set (MDS) problem for a practical and feasible standpoint.
- Proposed GARUDA-DN consisting of (i) the core to approximate the MDS; (ii) WFP pulses to tackle a new challenge, lost-all-packet problem; (iii) two-phase recovery to reduce possibility of collision as well as utilize the broadcast nature of wireless networks; and (iv) A-map to prevent error propagation.
- Evaluated performance of GARUDA-DN with other previous schemes; and showed that GARUDA-DN performs substantially better both in terms of latency and the number of retransmissions.
- Extended GARUDA-DN to support different delivery semantics; and showed the feasibility of the solution through extensive simulations.

Upstream Correlated Data Aggregation:

- Formulated the perfectly correlated data aggregation problem for upstream data traffic using the Steiner minimum tree (SMT) and showed the upper bound for message complexity.
- Proposed a decentralized and simple aggregation scheme, GARUDA-UP, integrating the shortest path tree (SPT) and the minimum dominating set to approximate the optimal solution, the SMT.

- Compared performance of the GARUDA-UP approach with the SPT and the SMT through simulations; showed that GARUDA-UP outperforms the SPT and closely approaches the centralized approximation of the SMT with less computational complexity and without global coordination.

6.2 Future Works

There are two main directions that one can extend this work in the future. The first is to improve the proposed scheme, GARUDA-UP, using heuristics to approach the Steiner tree closely. For example, in addition to the shortest path tree and the minimum dominating set, one can exploit the characteristics of the minimum spanning tree or minimum set cover with small amount of overhead and distributed coordination.

The other way is to find an optimal solution for the upstream data aggregation problem assuming correlation factor between 0 and 1; and then design an approximation solution for the general aggregation problem in a decentralized fashion so that one can implement it over wireless sensor networks.

REFERENCES

[1] AGARWAL, S., KRISHNAMURTHY, S. V., KATZ, R. H., and DAO, S. K., "Distributed Power Control in Ad-hoc Wireless Networks," in *Personal Indoor Mobile Radio Conference (PIMRC)*, pp. F59–F66, Oct. 2001.

[2] A.GOEL and D.ESTRIN, "Simultaneous Optimization for Concave Cost: Single Sink Aggregation or Single Source Buy-at-Bulk," in *ACM-SIAM Symposium on Discrete Algorithm*, 2003.

[3] AGRE, J. and CLARE, L., "An integrated architecture for cooperative sensing networks," *IEEE Computer Magazine*, vol. 33, pp. 106–108, May 2000.

[4] AKYILDIZ, I., SU, W., SANKARASUBRAMANIAM, Y., and CAYIRCI, E., "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, pp. 102–116, Aug. 2002.

[5] ALON, N. and SPENCER, J., *The Probabilistic Method*. J. Wiley and Sons, 1992.

[6] BERTSEKAS, D. and GALLAGER, R., *Data Networks*. USA: Prentice Hall, Inc., 1992.

[7] BONNET, P., GEHRKE, J., and SESHADRI, P., "Querying the physical world," *IEEE Personal Communications*, vol. 7, pp. 10–15, Oct. 2000.

[8] BROCH, J., JOHNSON, D. B., and MALTZ, D. A., "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks." Internet Draft draft-ietf-manet-dsr-03.txt, Oct. 1999.

[9] BULUSU, N., HEIDEMANN, J., and ESTRIN, D., "Gps-less low cost outdoor localization for very small devices," *IEEE Personal Communications Magazine*, vol. 7, pp. 28–34, Oct. 2000.

[10] BYERS, J. W., LUBY, M., MITZENMACHER, M., and REGE, A., "A digital fountain approach to reliable distribution of bulk data," in *SIGCOMM*, Oct. 1998.

[11] CERPA, A., ELSON, J., ESTRIN, D., GIROD, L., HAMILTON, M., and ZHAO, J., "Habitat monitoring: Application driver for wireless communications technology," in *the 2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, (Annapolis, USA), Apr. 2001.

[12] CHEN, B., JAMIESON, K., BALAKRISHNAN, H., and MORRIS, R., "Span: An Energy Efficient Coordination Algorithm for Topology Maintenanc in Ad Hoc Wireless Networks," in *Proceedings of ACM MOBICOM*, July 2001.

[13] CHENG, X., NARAHARI, B., SIMHA, R., CHENG, M. X., and LIU, D., "Strong minimum energy topology in wireless sensor networks: np-completeness and heuristics," *IEEE Transactions on Mobile Computing*, vol. 2, pp. 248–256, July 2003.

[14] COVER, T. and THOMAS, J., *Elements of Information Theory, ser. Wiley Series in Telecommunications*. John Wiley and Sons, Inc., 1991.

[15] CRAMER, J., SCHOLTZ, R., and WIN, M., "On the analysis of uwb communication channels," in *IEEE MILCOM*, (Atlantic City, USA), pp. 1191–1195, Oct. 1999.

[16] CRISTESCU, R., BEFERULL-LOZANO, B., and VETTERLI, M., "On Network Correlated Data Gathering," in *INFOCOM*, (Hong Kong), Mar. 2004.

[17] CRISTESCU, R. and VETTERLI, M., "Power Efficient Gathering of Correlated Data: Optimization,NP-Completeness and Heuristics," in *The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, June 2003.

[18] DEPARTMENT OF COMPUTER SCIENCE, R. U., "The CMU Monarch Project. The CMU monarch projects wireless and mobility extension to ns." http://www.monarch.cs.rice.edu/cmu-ns.html.

[19] DUBOIS-FERRIERE, H. and ESTRIN, D., "Efficient and practical query scoping in sensor networks," Tech. Rep. 39, EPFL and UCLA, April 2004.

[20] FALL, K. and VARDHAN, K., "ns notes and documentation." http://www-mash.cs.berkeley.edu/ns/, 1999.

[21] FEIGE, U., "A threshold of ln n for approximating set cover," *Journal of ACM*, vol. 45, pp. 634–652, Apr. 1998.

[22] F.HARARY, *Graph Theory*. Addison Wesley Publishing Co., Oct. 1969.

[23] FINN, G., "Routing and addressing problems in large metropolitan-scale internetworks," Tech. Rep. ISI/RR-87-180, USC/ISI, March 1987.

[24] FLOYD, S., JACOBSON, V., LIU, C., MCCANNE, S., and ZHANG, L., "A reliable multicast framework for light-weight sessions and application level framing," *IEEE/ACM Transaction on Networking*, vol. 5, pp. 784–803, Dec. 1997.

[25] GANDHI, R., PARTHASARATHY, S., and MISHRA, A., "Minimizing Broadcast Latency and Redunancy in Ad Hoc Networks," in *Proc. ACM MOBIHOC '03*, (Annapolis, USA), pp. 222–232, June 2003.

[26] GAREY, M. R. and JOHNSON, D. S., *Computers and Intractability, A Guide to the Theory of NP-completeness*. Freeman, 1979.

[27] GILBERT, E. N. and POLLAK, H. O., "Steiner Minimal Trees," in *SIAM J. Applied Math*, vol. 16, pp. 1–20, 1968.

[28] GOPALSAMY, T., SINGHAL, M., PANDA, D., and SADAYAPPAN, P., "A reliable multicast algorithm for mobile ad hoc networks," in *Proceedings of 22nd International Conference on Distributed Computing Systems*, (Vienna, Austria), pp. 563–570, July 2002.

[29] GROUP, I. M. W., "Mobile Ad-hoc Networking." http://www.ietf.org/html.charters/manet-charter.html.

[30] GUPTA, P. and KUMAR, P. R., "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. IT-46, pp. 388–404, Mar. 2000.

[31] HAYES, T. P., "Randomly coloring graphs of girth at least five," in *35th ACM Symposium on Theory of Computing*, (San Diego, USA), pp. 269–278, June 2003.

[32] HEINZELMAN, W. R., KULIK, J., and BALAKRISHNAN, H., "Adaptive protocols for information dissemination in wireless sensor networks," in *MOBICOM*, pp. 174–185, Aug 1999.

[33] HWANG, F., RICHARDS, D., and WINTER, P., *The Steiner Tree Problem*. North-Holland, 1992.

[34] IEEE STANDARDS BOARD, "802 Part 11: Wireless LAN Medium Access Control (MAC) and physical layer (PHY) specifications," Mar. 1999.

[35] INTANAGONIWAWAT, C., ESTRIN, D., GOVINDAN, R., and HEIDEMANN, J., "Impact of Network Density on Data Aggregation in Wireless Sensor Networks," in *International Conference on Distributed Computing Systems (ICDCS'02)*, (Vienna, Austria), July 2002.

[36] INTANAGONWIWAT, C., GOVINDAN, R., and ESTRIN, D., "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *MOBICOM*, (Boston, USA), pp. 56–67, Aug. 2000.

[37] JOHNSON, D. S., "Approximation algorithms for combinatorial problems," *Journal of Computer and System Sciences*, vol. 9, pp. 256–278, Dec. 1974.

[38] KAHN, J., KATZ, R., and PISTER, K., "Next century challenges: mobile networking for smart dust," in *ACM MOBICOM*, (Washington, USA), pp. 271–278, Aug. 1999.

[39] KARGER, D. R. and MINKOFF, M., "Building Steiner Trees with Incomplete Global Knowledge," in *Proceedings of the 41th Annual IEEE Symposium on Foundations of Computer Science*, 2000.

[40] KARP, B. and KUNG, H., "Greedy perimeter stateless routing for wireless networks," in *MOBICOM*, (Boston, USA), pp. 243–254, Aug. 2000.

[41] KARP, R. M., *Reducability among Combinatorial Problems, Complexity of Computer Computations*. New York: R. E. Miller and J. W. Thatcher, Plenum Press, 1972.

[42] KARP, R. M., "Reducibility among combinatorial problems," *Complexity of Computer Computations*, pp. 85–103, May 1972.

[43] KAWADIA, V. and KUMAR, P., "Power Control and Clustering in Ad Hoc Networks," in *INFOCOM*, (San Francisco, USA), Mar. 2003.

[44] KAWADIA, V., NARAYANASWAMY, S., R. ROZOVSKY, R. S. S., and KUMAR, P., "Protocols for Media Access Control and Power Control in Wireless Networks," in *Proceedings of the 40th IEEE Conference on Decision and Control*, (Orlando, FL, USA), pp. pp.1935–1940, Dec. 2001.

[45] KIM, D., TOH, C.-K., and CHOI, Y., "RODA: A new dynamic routing protocol using dual paths to support asymmetric links in mobile ad hoc network," in *Proceedings of Computer Communications and Networks*, (Las Vegas, U.S.A), pp. 4–8, Oct. 2000.

[46] LEE, S.-J., GERLA, M., and CHIANG, C.-C., "On-demand multicasting routing protocol," in *Proceedings of IEEE WCNC*, (LA, USA), pp. 1298–1302, Sept. 1999.

[47] LI, D. and CHERITON, D. R., "Oters (on-tree efficient recovery using subcasting)," in *6th International Conference on Network Protocols*, pp. 237–245, Oct. 1998.

[48] LI, D. and CHERITON, D. R., "Evaluating the utility of fec with reliable multicast," in *7th International Conference on Network Protocols*, pp. 97–105, Nov. 1999.

[49] LIN, S. and COSTELLO, D. J., *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, Oct. 1983.

[50] LUO, H., YE, F., CHENG, J., LU, S., and ZHANG, L., "TTDD - A Two-Tier Data Dissemination for Large-scale Sensor Networks," in *MOBICOM*, 2002.

[51] M. SANCHEZ, P. M. and HAAS, Z. J., "Determination of Critical Transmission Range in Ad-Hoc Networks," in *Proceedings of IEEE Workshop on Multiaccess Mobility and Teletraffic for Wireless Communications*, (Venice, Italy), Oct. 1999.

[52] MONKS, J., BHARGHAVAN, V., and HU, W., "A Power Controlled Multiple Access Protocol for Wireless Packet Networks," in *Proceedings of IEEE INFOCOM*, (Anchorage, USA), pp. 22–26, Apr. 2001.

[53] MOTE, M., "Mica, mica2 motes & sensors datasheets." http://www.xbow.com/Products/Wireless_Sensor_Networks.htm.

[54] NI, S.-Y., TSENG, Y.-C., CHEN, Y.-S., and SHEU, J.-P., "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of ACM MOBICOM*, (Seattle, USA), pp. 151–162, Aug. 1999.

[55] NICULESCU, D. and NATH, B., "Dv based positioning in ad-hoc networks," *Telecommunication Systems*.

[56] PARK, S.-J. and SIVAKUMAR, R., "Load Sensitive Transmission Power Control in Wireless Ad-hoc Networks," in *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, (Taipei, Taiwan), pp. 42–46, Nov. 2002.

[57] PARK, S.-J. and SIVAKUMAR, R., "Quantitative Analysis of Transmission Power Control in Wireless Ad-hoc Networks," in *Proceedings of International Workshop on Ad Hoc Networking*, (Vancouver, Canada), pp. 56–63, Aug. 2002.

[58] PENG, W. and LU, X., "Efficient broadcast in mobile ad hoc networks using connected dominating sets," *Journal of Software*, vol. 12, pp. 529–536, Dec. 1999.

[59] PERKINS, C. and ROYER, E., "Ad-hoc on demand distance vector routing," in *WMCSA*, (New Orleans, USA), Feb. 1999.

[60] PERRIG, A., SZEWCZYK, R., V. WEN, D. C., and TYGAR, J., "Spins: Security protocols for sensor networks," *Wireless Networks*, vol. 8, pp. 521–534.

[61] PORRET, A., MELLY, T., ENZ, C., and VITTOZ, E., "A low-power low-voltage transceiver architecture suitable for wireless distributed sensors network," in *IEEE International Symposium on Circuits and Systems*, (Geneva), pp. 56–59, Apr. 2000.

[62] POTTIE, G. and KAISER, W., "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, pp. 551–558, May 2000.

[63] PRABHAKAR, B., UYSAL-BIYIKOGLU, E., and GAMAL, A. E., "Energy-efficient Transmission over a Wireless Link via Lazy Packet Scheduling ," in *Proceedings of IEEE INFOCOM*, (Anchorage, USA), pp. 386 –394, Apr. 2001.

[64] RAMANATHAN, R. and R-HAIN, R., "Topology Control of Multihop Wireless Networks using Transmit Power Adjustment," in *Proceedings of IEEE INFOCOM*, pp. 404–413, Mar. 2000.

[65] RATNASAMY, S., KARP, B., YIN, L., YU, F., ESTRIN, D., GOVINDAN, R., and SHENKER, S., "GHT - A Geographic Hash-Table for Datacentric Storage," in *In First ACM International Workshop on Wireless Sensor Networks and their Applications*, 2002.

[66] RAY, S., UNGRANGSI, R., PELLEGRINI, F. D., TRACHTENBERG, A., and STAROBINSKI, D., "Robust location detection in emergency sensor networks," in *INFOCOM*, (San Francisco, USA), Mar. 2003.

[67] RIZZO, L. and VICISANO, L., "A reliable multicast data distribution protocol based on software fec techniques(rmdp)," in *IEEE Workshop on High-Performance Comm. Systems(HPCS'97)*, pp. 115–124, Apr. 1997.

[68] RODOPLU, V. and MENG, T., "Minimum Energy Mobile Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1333–1344, 1999.

[69] ROYER, E. M. and PERKINS, C. E., "Multicast operation of the ad-hoc on-demand distance vector routing protocol," in *mobicom*, (Seattle, USA), pp. 207–218, Aug. 1999.

[70] ROYER, E. M. and PERKINS, C. E., "Transmission Range Effects on AODV Multicast Communication," *ACM Mobile Networks and Applications special issue on Multipoint Communication in Wireless Mobile Networks*, vol. 7, pp. 455–470, Dec. 2002.

[71] SANKARASUBRAMANIAM, Y., AKAN, O. B., and AKYILIDIZ, I. F., "Esrt: Event-to-sink reliable transport in wireless sensor networks," in *ACM MOBIHOC*, pp. 177–188, Nov. 2003.

[72] SANTI, P., BLOUGH, D., and VAINSTEIN, F., "A probabilistic analysis for the range assignment problem in ad hoc networks," (Long Beach, USA), pp. 212–220, Oct. 2001.

[73] SAVARESE, C., RABAEY, J., and BEUTEL, J., "Locationing in distributed ad-hoc wireless sensor networks," in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2001)*, (Salt Lake City, USA), May 2001.

[74] SAVVIDES, A. and SRIVASTAVA, M. B., "A Distributed COmputation Platform for Wireless Embedded Sensing," in *Proc. of ICCD*, (Freiburg, Germany), 2002.

[75] SCHURGERS, C., TSIATSIS, V., and SRIVASTAVA, M., "STEM: Topology Management for Energy Efficient Sensor Networks," in *IEEE Aerospace Conference*, (Big Sky, USA), pp. 78–89, Mar. 2002.

[76] SHANG, Y., RUML, W., Y.ZHANG, and FROMHERZ, M., "Localization from mere connectivity," in *4th ACM Intl. Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc)*, (Annapolis, USA), June 2003.

[77] SHEN, C., SRISATHAPORNPHAT, C., and JAIKAEO, C., "Sensor information networking architecture and applications," *IEEE Personal Communications*, pp. 52–59, Aug. 2001.

[78] SHIS, E., CHO, S., ICKES, N., MIN, R., SINHA, A., WANG, A., and ACHANDRAKASAN, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," in *MOBICOM*, (Rome, Italy), pp. 272–286, July 2001.

[79] SINGH, S. and RAGHAVENDRA, C., "PAMAS: Power Aware Multi-access Protocol with Signalling for Ad hoc Networks," in *Proceedings of ACM MOBICOM*, (Dallas, USA), pp. 5–26, July 1998.

[80] SINGH, S., WOO, M., and RAGHAVENDRA, C. S., "Power-aware routing in mobile ad hoc networks," in *ACM MOBICOM*, pp. 181–190, 1998.

[81] SIVAKUMAR, R., SINHA, P., and BHARGHAVAN, V., "CEDAR: a Core-Extraction Distributed Ad hoc Routing algorithm," *IEEE Journal on Selected Areas in Communications (Special Issue on Ad-hoc Routing)*, vol. 17, pp. 1454–1465, Aug. 1999.

[82] SRIVASTAVA, M., MUNTZ, R., and POTKONJAK, M., "Smart kindergarten: Sensor-based wireless networks for smart developmental problem-solving environments," in *ACM MOBICOM*, (Rome, Italy), pp. 132–138, July 2001.

[83] STANN, F. and HEIDEMANN, J., "Rmst: Reliable data transport in sensor networks," in *Proceedings of the First International Workshop on Sensor Net Protocols and Applications*, (Anchorage, USA), pp. 345–353, Apr. 2003.

[84] STEERE, D., BAPTISTA, A., MCNAMEE, D., PU, C., and WALPOLE, J., "Research challenges in environmental observation and forecasting systems," in *MOBICOM*, (Boston, USA), pp. 292–299, Aug. 2000.

[85] TAKAGI, H. and KLEINROCK, L., "Optimal transmission ranges for randomly distributed packet radio terminals," *IEEE Trans. on Communication*, vol. 32, pp. 246–257, Mar. 1984.

[86] TAKAHASHI, H. and MATSUYAMA, A., "An approximate solution for the steiner problem in graphs," *Math. Japonica 24*, vol. 24, pp. 573–577, Jan. 1980.

[87] TANENBAUM, A. S., *Computer Network*. Prentice Hall, 1996.

[88] TANG, K. and GERLA, M., "Mac reliable broadcast in ad hoc networks," in *Proc. IEEE MILCOM*, (Virginia, USA), pp. 1008–1013, Aug. 2001.

[89] TILAK, S., ABU-GHAZALEH, N. B., and HEINZELMAN, W., "A taxonomy of wirelss micro-sensor network models," *Mobile Computing and Communications Review*, vol. 6, no. 2, pp. 28–36, 2001.

[90] VARDHAN, S., WILCZYNCKI, M., POTTIE, G., and KAISER, W., "Wireless integrated network sensors (wins): Distributed in situ sensing for mission and flight systems," in *IEEE Aerospace Conference*, (USA), pp. 459–463, Aug. 2000.

[91] VAZIRANI, V. V., *Approximation Algorithms*. Springer, May 2001.

[92] WAN, C.-Y., CAMPBELL, A., and KRISHNAMURTHY, L., "PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks," in *Proc. ACM International Workshop on Sensor Networks and Architectures*, (Atlanta, USA), pp. 1–11, Sept. 2002.

[93] WAN, C.-Y., EISENMAN, S. B., and CAMPBELL, A. T., "Coda: Congestion detection and avoidance in sensor networks," in *Proceedings of ACM SenSys*, (Orlando, USA), Nov. 2003.

[94] WANG, N.-Y. B. and CHANG, R.-C., "An Upper Bound for the Average Length of the Euclidean Minimum Spanning Tree," in *J. Computer Math*, vol. 30, pp. 1–12, 1989.

[95] WANG, Y., LI, X.-Y., WAN, P.-J., and FRIEDER, O., "Sparse Power Efficient Topology for Wireless Networks," *Journal of Parallel and Distributed Computing*, 2002.

[96] WARNEKE, B., LIEBOWITZ, B., and PISTER, K., "Smart dust: Communicating with a cubic-millimeter computer," *IEEE Computer*, vol. 8, pp. 2–9, Jan. 2001.

[97] WATTENHOFER, R., LI, L., BAHL, P., and WANG, Y.-M., "Distributed Topology Control for Ad-hoc Networks," in *Proceedings of IEEE INFOCOM*, (Anchorage, USA), pp. 22–26, Apr. 2001.

[98] WILLIAMS, B. and CAMP, T., "Comparison of Broadcasting Techniques for Mobile Adhoc Networks," in *ACM MOBIHOC*, (Lausanne, Switzerland), pp. 194–205, June 2002.

[99] WOO, A. and CULLER, D., "A transmission control scheme for media access in sensor networks," in *MOBICOM*, (Rome, Italy), pp. 221–235, July 2001.

[100] YAO, K., HUDSON, R., REED, C., CHEN, D., and LORENZELLI, F., "Blind beamforming on a randomly distributed sensor array system," *IEEE Journal of Selected Areas in Communications*, vol. 16, pp. 1555–1567, Oct. 1998.

[101] YE, F., LUO, H., CHENG, J., LU, S., and ZHANG, L., "A two-tier data dissemination model for large-scale wireless sensor networks," in *MOBICOM*, (Atlanta, USA), pp. 148–159, Sept. 2002.

[102] YE, W., HEIDEMANN, J., and ESTRIN, D., "An energy-efficient mac protocol for wireless sensor networks," in *Proceedings of IEEE INFOCOM*, (New York, USA), pp. 1567–1576, June 2002.

[103] YE, Z., KRISHNAMURTHY, S., and TRIPATHI, S., "A framework for reliable routing in mobile ad hoc networks," in *roceedings of IEEE INFOCOM*, (San Francisco, USA), pp. 270–280, Mar. 2003.

[104] YUAN, W., KRISHNAMURTHY, S. V., and TRIPATHI, S. K., "Synchronization of Multiple Levels of Data Fusion in Wireless Sensor Networks," 2003.

[105] ZELIKOVSKY, A., "Better Approximation Bounds for the Network and Euclidean Steiner Tree Problems," in *Tech. Rep. CS-96-06, University of Virginia, Charlottesville*, (VA, USA), 1996.

VITA

Seung-Jong Park received his B.S. degree in Computer Science from the Korea University and M.S. degree in Computer Science from the Korea Advanced Institute of Science and Technology (KAIST) in 1993 and 1995, respectively. He worked for Shinsegi Telecomm (a CDMA wireless cellular provider) in Korea as a research and technical staff from 1995 to 2000. Since 2000, He has been a doctoral student in the school of Electrical and Computer Engineering at the Georgia Institute of Technology. His research interests are wireless network protocols, mobile computing, and network QoS (Quality of Service) in the areas of wireless cellular, ad-hoc, and sensor networks.