Fall 2009

Fault-Tolerance and Recovery in Wireless Sensor Networks

Kevin M. Somervill

Old Dominion University

FAULT-TOLERANCE AND RECOVERY IN WIRELESS SENSOR NETWORKS

by

Kevin M. Somervill
B.S. May 2000, Christopher Newport University

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

MASTER OF SCIENCE

COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
December, 2009

Approved by:

Lee A. Belfore II (Director)

Dimitrie Popescu (Member)

Min Song (Member)

ABSTRACT

FAULT-TOLERANCE AND RECOVERY IN WIRELESS SENSOR NETWORKS

Kevin M. Somervill
Old Dominion University, 2009
Director: Dr. Lee A. Belfore II

The topic of Wireless Sensor Networks (WSNs) has gained considerable attention in the research community due to the variety of applications and interesting challenges in developing and deploying such networks. The typical WSN is significantly energy constrained and often deployed in harsh or even hostile environments, resulting in sensor nodes that are prone to failure. Failing nodes alter the topology of the network resulting in segmented routing paths and lost messages, utlimately reducing network efficiency. These issues spur the desire to develop energy-efficient, Fault-Tolerant (FT) algorithms that enable the network to persist in spite of the failed nodes. This work continues previous research on a class of WSN. Unique contributions center around a proposed FT recovery mechanism to mitigate an observed failure mechanism and increase the aggregate network efficiency. The proposal is explicated through concise terminology relevant to the domain of fault-tolerance and WSNs. The recovery mechanism is evaluated through simulation and the results are presented in a characterization of the effect of the proposal on the network's performance.

ACKNOWLEDGMENTS

The author would like to acknowledge the patience and love of his wife and children. They have each endured several years of missing athletic events and family movie nights during the pursuit of this master's degree. I would also like to acknowledge my advisor for his guidance and commitment to this research.

TABLE OF CONTENTS

Page
LIST OF TABLES vi
LIST OF FIGURES ix

CHAPTERS

I INTRODUCTION 1
  I.1 Problem Statement 1
  I.2 Literature Review 3
  I.3 Thesis Organization 4
II BACKGROUND 6
  II.1 Wireless Sensor Networks 6
  II.2 Applications 7
  II.3 Network Model 7
  II.4 Communications Model 11
  II.5 Fault Tolerance 18
III THEORY 21
  III.1 Fault Model 21
  III.2 Fault Recovery 24
IV RESULTS 29
  IV.1 Simulation 29
  IV.2 Simulation Results 33
V SUMMARY AND FUTURE WORK 61
  V.1 Summary 61
  V.2 Contribution 61
  V.3 Future work 62

BIBLIOGRAPHY 64

APPENDICES

A ADDITIONAL DATA SETS 70
  A.1 Plots for Initial Energy of 200 70
  A.2 Plots for Initial Energy of 400 74

VITA 78

LIST OF TABLES

Page

1 Default simulation parameters. 33
2 Configuration parameters for simulation of a uniformly distributed network varying initial node energy. 34
3 Configuration parameters for simulation of a uniformly distributed network. 38
4 Configuration parameters for simulation of a uniformly distributed network with recovery. 41
5 Average percent increase associated with recovery protocol in a uniformly distributed network with 4 to 12 wedges. 45
6 Configuration parameters for simulation of a proportionally distributed network varying initial node energy. 47
7 Configuration parameters for simulation of a proportionally distributed network. 51
8 Configuration parameters for simulation of a proportionally distributed network with recovery. 55
9 Average percent increase associated with recovery protocol in a proportionally distributed network. 57

LIST OF FIGURES

		Page
1	Network of distributed wireless sensor nodes and central sink node.	8
2	Algorithm for determining the associations of a node to its wedge, corona, and cluster.	10
3	The process of training the network.	10
4	Network communications route within a wedge.	12
5	Communications protocol sequence diagram.	13
6	Illustration of primary (a) and secondary (b) network planes.	25
7	Algorithm for determining the associations of a node to its primary and secondary wedges, as well as its corona, and cluster.	26
8	Partitioned clusters with failed cluster prior to recovery transition.	27
9	Partitioned clusters after fault recovery plane transition.	28
10	Network longevity for uniformly distributed 8-wedge networks with initial energy of 100, 200, 400, and 800 units per node.	35
11	Network longevity for uniformly distributed 4-wedge networks with initial energy of 100, 200, 400, and 800 units per node.	36
12	Network longevity for uniformly distributed 12-wedge networks with initial energy of 100, 200, 400, and 800 units per node.	37
13	Network life for uniform node distribution in single-plane configuration.	39
14	Network communications efficiency for uniform node distribution in single-plane configuration.	40
15	Network energy utilization for uniform node distribution in single-plane configuration.	40
16	Network life for uniform node distribution in dual-plane configuration.	42
17	Network communications efficiency for uniform node distribution in dual-plane configuration.	42
18	Network energy utilization for uniform node distribution in dual-plane configuration.	43
19	Network life for uniform node distribution in single-plane and dual-plane configurations.	45
20	Network communications efficiency for uniform node distribution in single-plane and dual-plane configurations.	46
21	Network energy utilization for uniform node distribution in single-plane and dual-plane configurations.	46
22	Network longevity for proportionally distributed 8-wedge networks with initial energy of 100, 200, 400, and 800 units per node.	49
23	Network longevity for proportionally distributed 4-wedge networks with initial energy of 100, 200, 400, and 800 units per node.	49
24	Network longevity for proportionally distributed 12-wedge networks with initial energy of 100, 200, 400, and 800 units per node.	50
25	Network longevity for large uniformly distributed network.	50

26 Network life for proportional node distribution in single-plane configuration. 51
27 Network communications efficiency for proportional node distribution in single-plane configuration. 52
28 Network energy utilization for proportional node distribution in single-plane configuration. 52
29 Efficiency traces for 4 and 8-wedge uniformly and proportionally distributed network configurations. 54
30 Network life for proportional node distribution in dual-plane configuration. 56
31 Network communications efficiency for proportional node distribution in dual-plane configuration. 56
32 Network energy utilization for proportional node distribution in dual-plane configuration. 57
33 Network life for proportional node distribution in single-plane and dual-plane configurations. 59
34 Network communications efficiency for proportional node distribution in single-plane and dual-plane configurations. 59
35 Network energy utilization for proportional node distribution in single-plane and dual-plane configurations. 60
36 Network with multiple sink nodes. 63
37 Network life for uniform node distributed networks with an initial energy of 200 units. 70
38 Network communications efficiency for uniform node distributed networks with an initial energy of 200 units. 71
39 Network energy utilization for uniform node distributed networks with an initial energy of 200 units. 71
40 Network life for proportional node distributed networks with an initial energy of 200 units. 72
41 Network communications efficiency for proportional node distributed networks with an initial energy of 200 units. 72
42 Network energy utilization for proportional node distributed networks with an initial energy of 200 units. 73
43 Network life for uniform node distributed networks with an initial energy of 400 units. 74
44 Network communications efficiency for uniform node distributed networks with an initial energy of 400 units. 75
45 Network energy utilization for uniform node distributed networks with an initial energy of 400 units. 75
46 Network life for proportional node distributed networks with an initial energy of 400 units. 76
47 Network communications efficiency for proportional node distributed networks with an initial energy of 400 units. 76

48 Network energy utilization for proportional node distributed networks with an initial energy of 400 units. 77

CHAPTER I

INTRODUCTION

Wireless Sensor Networks (WSNs) have enjoyed considerable interest from the research community [1] due to their varied applications and unique challenges. They have found applications in military use for "enemy tracking, battlefield surveillance, and target classification" [2] as well as other applications including traffic monitoring, cross-border infiltration detection, military reconnaissance, habitat monitoring, etc [1]. Due to the low manufacturing costs of WSN nodes, they can be deployed in large numbers yielding challenges in network management such as routing, topology control, and data management protocols. These challenges are only complicated by severe energy constraints and the inherently unreliable nature of wireless communications which have yielded work in increasing network efficiency and augmenting protocols with varying degrees of fault-tolerance. This thesis specifically addresses the application of fault-tolerance to enhance the aggregate efficiency of the WSN described by Wadaa et al. [3] and further studies by Elmiligui [2].

I.1 Problem Statement

The topic of WSN continues to grow as a fertile research area. Efforts continually seek to overcome the complications of reliable, or even fault-tolerant, communications in large wireless networks. Literature, however, lacks rigorous and consistent definitions for terminology relevant to reliability and fault-tolerance as applied to WSN. This, in turn, confuses the problem space when seeking to develop and evaluate protocols for dependable wireless communication networks.

In spite of the considerable work over the past decades to rigorously define the language for fault-tolerance and reliability of components and systems [4], current literature neither defines nor applies fault-tolerance terminology in a manner consistent with counterparts in computing system networks. This leaves the topic mired

This thesis follows the style of *IEEE*.

with vague language and conflicting interpretations of the capabilities of proposed algorithms and techniques.

Specific to the context of wireless network routing, a number of protocols have been developed to address the issue of dependable communications within WSN. Wadaa et al. [3], further investigated by Elmiligui [2], propose a network partitioned with a routing hierarchy and protocol which seeks to minimize the energy consumption of the aggregate network. As discussed by Emiligui in his thesis [2], the network fails due to the depletion of energy in the central ring of nodes around the sink node, leaving the sink node segmented from the remaining viable network nodes. With the existing protocol, at the extinction of the network (when the sink is isolated from the remaining live network nodes), the remaining energy is effectively consumed with zero efficiency because it is no longer available for useful work which negates the premise that their approach minimizes energy consumption within the network. Also, neither works adequately characterize the remaining available energy in the network after the depletion of the central ring.

Two issues have been presented related to the proposed class of WSN. The first is the clear and consistent use of terminology with respect to Fault-Tolerant (FT) and WSN in general. This work addresses this shortcoming by defining the relevant terms of fault-tolerance, reliability, and dependability within the context of WSN. The second issue is specific to a class of WSN [2, 3] and deals with the depletion energy in the central ring around the sink resulting from the current network protocol. This work investigates the characterization of the remaining energy in the network after the extinction of the central ring. Then, a modified protocol is proposed to instill fault-tolerance into the network via reorganization of the network in response to network degradation. The amended protocol, in conjunction with the characterization of the remaining network energy, seeks to extend the useful life of the network, increasing utility and efficiency in terms of longevity and energy expended, minimizing residual energy after network extinction.

I.2 Literature Review

As part of the continued research and development of WSNs, significant work has been invested to address the challenges of hostile environments, limited energy, and unreliable communications over low power wireless links. It has been shown that maximizing network lifetime is NP-hard [5] leaving many to address this through heuristically-based, energy-efficient protocols [5–8]. The challenge of energy-efficient operation is a universal theme in modern electronic and communication systems, a dilemma that is only magnified in WSN applications. Energy-efficient routing is of keen interest [9–13] as evidenced through the development of protocols for FT WSN utilizing routing techniques to manage energy conservation [6]. To reduce the power associated with communications, data transmission is reduced via data aggregation. A FT data aggregation scheme has been proposed along with a hierarchical approach to data aggregation that is resilient to node failures [14].

Others have investigated approaches to fault-tolerance in WSN through routing and network protocols. A k-fold routing protocol [15] uses a subset of the network nodes as routing infrastructure and fault-tolerance is realized through k redundant sets of the infrastructure nodes. A directed flooding approach has been presented [16] as well as the use of Forward Error Correction (FEC) in conjunction with a FT routing protocol to improve retransmission efficiency and hence conserve energy [17]. Four requirements of routing protocols for WSN have been proposed as being energy efficient, fault-tolerant, load balanced, and scalable. To address these, the Weighted Energy Aware Routing (WEAR) protocol selects multiple routing paths determined through heuristic weighting associated with metrics such as node energy and the distance to the destination [18]. An adaptive algorithm is presented by Gregoire and Koren [19] to address route failures during message transit.

Network organization has also been explored in a number of works. A topology control protocol for large WSN is presented in which nodes define topology through weighted neighbor relationships [20]. Chen et al. [21] states that topology is key to "extending the lifetime of the network" and proposes using redundant nodes to provide fault tolerance for the communications backbone. A distributed algorithm which seeks to improve performance through assigning minimum power to network nodes has been proposed [22]. The topology can also be managed through varied transmission strength to achieve k-degree connectivity to facilitate data aggregation and $k-1$ fault tolerance [23, 24]. A similar approach adjusts node transmission range to implement topology control for FT WSN [21].

Other algorithms have studied the use of encoding techniques such as Reed-Solomon encoding [25] to enhance the reliability of the communications. Djukic presents work on using FEC to encode packets that are fragmented and distributed over disjoint routes to improve the reliability of the network [11]. Wang et al. [26,27] present data encoding used to combat channel fading and an error correction code, distributed classification fusion approach using error correcting codes (DCFECC), is employed to mitigate data errors and provide FT communications.

In each of these, energy efficiency and network reliability are key aspects of developing protocols for WSN. Due to the limited power available in sensor nodes for data collection and communication, conserving energy is paramount in extending network life. Eventually, energy resources are depleted leaving the network to manage continued performance in the presence of failed or failing nodes. This in conjunction with the inherent unreliability of wireless communications has endeavored a research community to cultivate FT protocols to improve the probability of successful operation and reduce the inefficiencies in lost or corrupted traffic.

I.3 Thesis Organization

This thesis is organized into five chapters. This first chapter presents the problem statement with a review of previous relevant research. The second chapter provides background for the subject of WSNs and the network model for the class of WSN studied in this work. It also introduces the relevance and terminology of fault-tolerance in the domain of WSN. The third chapter describes a fault in the network model of chapter II and proposes a fault-tolerant recovery mechanism for the described fault. The fourth chapter presents the results from simulating the network model. The fifth chapter concludes with a brief summary of the five chapters and a description of the contributions of this work, closing with a brief listing of future candidate research topics.

CHAPTER II

BACKGROUND

The previous chapter introduces the topic of this thesis which is to address a fault in a class of WSNs by implementing fault-tolerance to improve the aggregate network efficiency. Also presented is a cursory literature review of related works in FT WSNs. This chapter presents the background and terminology to establish a clear context for the theory presented next in Chapter III. The discussion begins with the description of a typical WSN followed by a cursory overview of applications. The network model is then refined as the basis for the application of a proposed recovery mechanism. The topic of fault-tolerance concludes this chapter defining key terminology as well as establishing relevance to the topic WSN.

II.1 Wireless Sensor Networks

A WSN is a collection of dispersed sensors collecting data to provide observability into the physical environment of some area of interest [17]. The sensors are nodes in the network and are comprised of one or more measurement devices, computational resources, a communications component, and a finite power source (i.e. battery) [9,28]. The sensors communicate information via wireless (RF) transmissions to other network elements, routing messages to other nodes or their base station, often referred to as a *sink*. The sink serves as the gateway between the user application and the sensor network.

Typically, the sensors are densely deployed, prone to failure, and limited in power [16]. Power is an especially critical resource as it is difficult to recharge or replace failed nodes [9, 13] after deployment and when the node power is depleted, the node is no longer able to transmit or receive communications [22]. The communications component consumes the most power in the sensor node [28] making efficient communications paramount to sustaining viable operations. In wireless communications, power is proportional to the range of communications by a factor of r^n, where r is the radius of communications between the transmitter and receiver and n is the pathloss exponent in the range of 2 and 5 [29]. From this, it is expected that multihop routing is more efficient than direct transmission from nodes to the sink [5, 28]. It has also been proposed that to conserve power, the WSN should establish a "communication backbone" to minimize the number of routes [15].

II.2 Applications

Wireless Sensor Networks have been extensively researched [1] and have found applications in military use for "enemy tracking, battlefield surveillance, and target classification" [2]. Other applications include traffic monitoring, cross-border infiltration detection, military reconnaissance, habitat monitoring, etc [1]. Due to the low manufacturing costs of WSN nodes, they can be deployed in large numbers which leads to large data sets that must be managed and collected. These large data sets require data aggregation, or data fusion, to reduce both the bandwidth and power consumed by the network by reducing both the data size and the number of transmissions of redundant data to a summary form [1]. To protect the integrity of the network, the communications links are enhanced with FT protocols that increase the reliability of the network with respect to the power required to communicate the data to the user.

II.3 Network Model

The network is comprised of a collection of wireless sensor nodes organized into local groups with an imposed hierarchical communications structure. Within the hierarchy, there are three classifications of nodes that manage and perform: message generation, message distribution, data collection, data aggregation, and data return. The various nodes are dispersed over some area with the topology of a typical network similar to that shown in Figure 1.

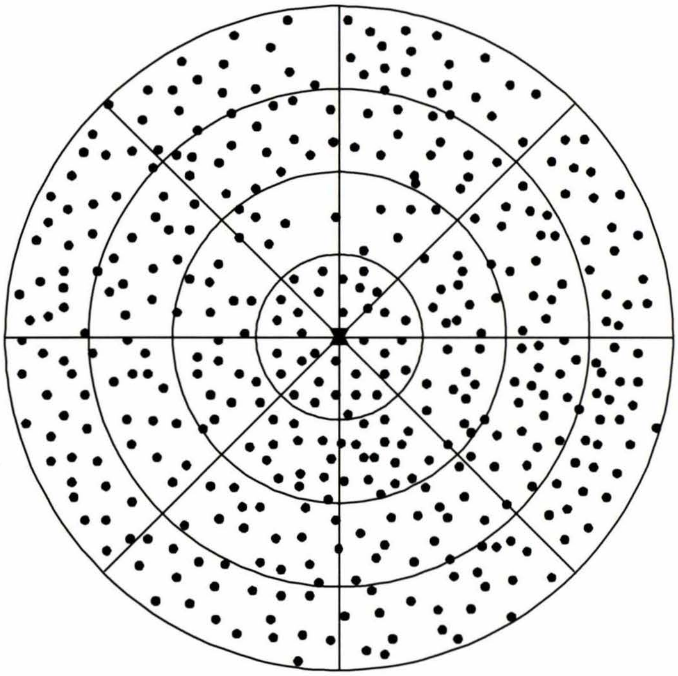


FIG. 1: Network of distributed wireless sensor nodes and central sink node.

II.3.1 Network Topology

The network is partitioned into groups, or clusters, of nodes that cooperate to accomplish the objective of data collection. The process of organizing the nodes into a coherent network is referred to as training where the training protocol used in this work is described by Wadaa et al. [3]. The reference network is divided into congruent arcs, referred to as wedge angles or simply wedges, and then into equally spaced concentric circles, referred to as coronas. The network division is illustrated in Figure 1 where the dots represent nodes in the network, the radial lines establish wedge boundaries, and the concentric circles denote corona boundaries. The sink is shown at the center of the network. The wedge angles, θ_ω, are assumed to be uniform as well as the radial distance between the corona boundaries which is referred to as the corona stride, $\mathcal{C}_s$. Assuming $\mathcal{N}_\omega$ is the number of wedges and $\mathcal{N}_\mathcal{C}$ is the number of coronas, θ_ω and $\mathcal{C}_s$ are defined as

$$\theta_\omega = \frac{2\pi}{\mathcal{N}_\omega} \tag{1}$$

$$\mathcal{C}_s = \frac{r_{Network}}{\mathcal{N}_C} \tag{2}$$

where $r_{Network}$ is the radius of the network coverage area. A wedge is denoted by $\omega_j = j, j \in [1, \mathcal{N}_\omega]$ and the coronas are denoted by $\mathcal{C}_i = i, i \in [1, \mathcal{N}_C]$. For a given node n, with polar coordinates (r_n, θ_n), its wedge $\omega(n)$ and corona, $\mathcal{C}(n)$, are given by

$$\omega(n) = \left\lfloor \frac{\theta_n}{\theta_\omega} \right\rfloor + 1 \tag{3}$$

$$\mathcal{C}(n) = \left\lceil \frac{r_n}{\mathcal{C}_s} \right\rceil \quad , r_n > 0 \tag{4}$$

The nodes co-located within a wedge ω_j and corona $\mathcal{C}_i$ are logically grouped and referred to as a cluster. A cluster can be identified by its wedge and corona coordinates or by an index. These are denoted as either $cluster_{j,i}$ or $cluster_k$, respectively. The index, k is determined by

$$k = [(\omega_j - 1)\mathcal{N}_C] + \mathcal{C}_i, \tag{5}$$

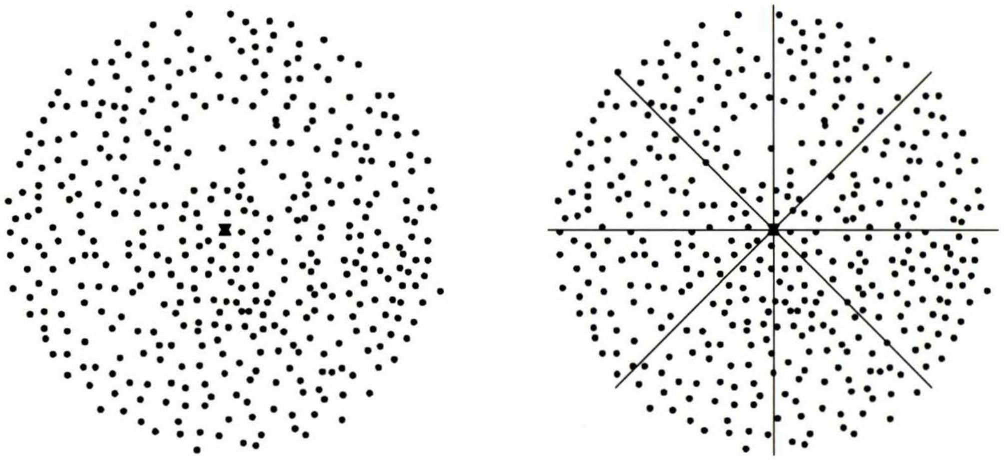
II.3.2 Network Training

At the center of the network, the sink performs the network training using a directional antenna and varying the transmission energy to determine the direction and distance of the nodes, respectively. The general algorithm is given in Figure 2. Initially the network is an untrained, disorganized distribution of nodes as shown in Figure 3(a). The sink begins the training process by initiating communication with nodes in one direction at minimum transmission strength. Transmission energy is gradually increased and only those nodes within range can communicate. This locates the network nodes within a given wedge as illustrated in Figure 3(b). Gradually increasing the transmission range enables the sink to resolve each of the nodes within the wedge to one of the coronas. This process is then repeated for each of the wedges until the network topology resembles that shown in Figure 1.

The distribution of nodes within a network area can be random and for the purposes of this study, the node distribution is assumed to be either uniform over the network area, or equally among clusters. In practical applications, the deployment mechanism will determine the actual distribution and the applicability of this work.

for all $\omega_j : j \in [1, \mathcal{N}_\omega]$ **do**
  for all $\mathcal{C}_i : i \in [1, \mathcal{C}_i]$ **do**
    for all $n : n \in [\omega_j \cap \mathcal{C}_i]$ **do**
      $\omega(n) \Leftarrow \left\lfloor \frac{\theta_n}{\theta_\omega} \right\rfloor + 1$
      $\mathcal{C}(n) \Leftarrow \left\lceil \frac{r_n}{\mathcal{C}_s} \right\rceil$
      $cluster(n) \Leftarrow [(\omega(n) - 1)\mathcal{N}_\mathcal{C}] + \mathcal{C}(n)$
    end for
  end for
end for

FIG. 2: Algorithm for determining the associations of a node to its wedge, corona, and cluster.



(a) Untrained network. (b) Partially trained network.

FIG. 3: The process of training the network.

II.3.3 Network Hierarchy

The network is organized hierarchically and function is determined by the role of the node in that hierarchy. There are two types of nodes in the network serving three basic functions. The first type is referred to as the sink node and it is located at the center of the network. The sink node is assumed to have unlimited energy storage and serves as the gateway to the network. It is responsible for the initiation of all data traffic and is the destination of aggregated data within the network. The remaining nodes are distributed around the sink node in collections forming clusters as previously described. These nodes are referred to as cluster nodes and have limited energy storage. The cluster nodes periodically arbitrate for function where one node in each cluster performs the function of cluster head and the rest serve as data nodes. The cluster head receives data requests from the sink and broadcasts the request to the cluster. The cluster head is responsible for collecting the data from the nodes and returning it to the sink. A secondary function of the cluster head is to relay communications between clusters. These are either outbound requests from the sink, or inbound replies to the sink. The data nodes respond to data requests broadcast by their cluster head. On request, the data nodes sense the local environment and return the sample data to the cluster head. A more detailed description of the protocol follows.

II.4 Communications Model

The network implements a simple communications protocol in which all communications originate, and if successful, terminate at the sink node. Communications are constrained to propagate between adjacent clusters within the same wedge as shown in Figure 4. That is, messages cannot cross wedge boundaries, instead, they propagate from the sink through adjacent clusters to the destination (target) cluster. In the figure, messages route from the sink (shown as a square in the bottom right) to the intermediate cluster heads (shown as solid circles). The message is eventually received by the target cluster head and distributed to the cluster's data nodes (shown as circles). In the event a cluster fails to complete a request, the message is lost. The following sections provide a more detailed description of the communications protocol, a traffic model, and the energy consumption model for the network.

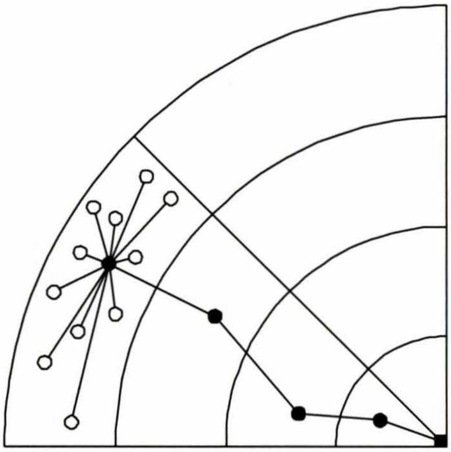


FIG. 4: Network communications route within a wedge.

II.4.1 Protocol Overview

The process of completing a data collection task is depicted in the sequence diagram in Figure 5. The sink node initiates the process by issuing a *Receive Request* message to the appropriate wedge. The request is destined for a cluster selected by the sink as the target cluster to perform the data collection task. This request is received by the cluster head in the first corona, referred to as the original cluster. In the diagram, the request is not destined for the original cluster and hence the *Receive Request* message is forwarded to the cluster head in the next corona away from the sink. The *Receive Request* message is forwarded by the cluster heads of adjacent clusters until it is received by the cluster head of the target cluster. The cluster head of the target cluster then transmits a *Broadcast* to the cluster requesting for cluster nodes to participate in a data collection task. Live nodes in the cluster reply with a *Receive Interest* message to acknowledge receipt of the *Broadcast* and participate in the data collection task. The data request is characterized with a Quality of Service (QoS) parameter which determines the number of samples to be collected and hence the number of nodes that should participate in the data collection. The cluster head attempts to fulfill the QoS requirement by responding to a sufficient

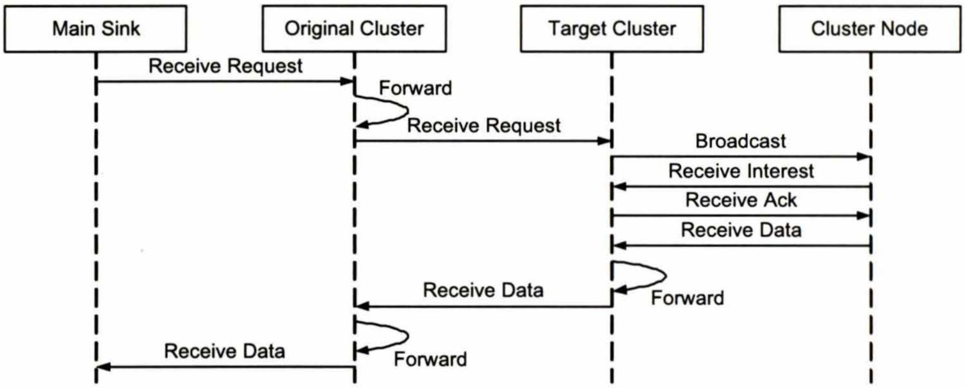


FIG. 5: Communications protocol sequence diagram.

set of data nodes that have indicated interest. The data nodes are included in the actual data collection task by receiving a *Receive Acknowledge* message from the cluster head. The data nodes then respond with *Receive Data* messages to the cluster head. When all of the acknowledged data nodes have responded with data, the QoS requirement has been fulfilled. The aggregated data forms a data product which is sent by the cluster head to the sink. The data product is forwarded as necessary by intermediate clusters until it is received by the sink and the task is completed.

II.4.2 Traffic Model

With the network topology and communications protocol described, the traffic model can now be constructed. A uniform distribution of tasks to the clusters has been assumed yielding a probability that a given cluster k is the target of a task as

$$P\{k\} = \frac{1}{\mathcal{N}_\omega \mathcal{N}_C} \tag{6}$$

The arrival rate of messages to the network is λ_m, but considering that communications are confined to a single wedge, it is convenient to distinguish the arrival rate to a wedge as λ_ω which is defined as

$$\lambda_\omega = \frac{\lambda_m}{\mathcal{N}_\omega} \tag{7}$$

Cluster traffic is comprised of tasks destined for the cluster and communications that are forwarded by the cluster. The arrival rate of messages to cluster k is denoted with λ_k and is the aggregate of the arrival rates for tasks to the cluster, λ_{tasks}, and forwards through the cluster, $\lambda_{forward}$ The arrival rate λ_k is given by

$$\lambda_k = \lambda_{tasks} + \lambda_{forward} \tag{8}$$

The first term, λ_{tasks}, is given by

$$\lambda_{tasks} = P\{k\}\lambda_m = \frac{\lambda_m}{\mathcal{N}_\omega \mathcal{N}_C} = \frac{\lambda_\omega}{\mathcal{N}_C} \tag{9}$$

Forward traffic is encountered by the cluster for both messages outbound from the sink and messages inbound to the sink. Forwarded messages are either destined for, or returned from, clusters further away from the sink. Let $\mathcal{C}(k)$ denote the corona index of k, the total number of clusters further from the sink than cluster k is given by $\mathcal{N}_C - \mathcal{C}(k)$. The arrival rate of forwards is then given by

$$\lambda_{forward} = 2\frac{(\mathcal{N}_C - \mathcal{C}(k))}{\mathcal{N}_C}\lambda_\omega \tag{10}$$

Collecting these terms together yields an aggregate arrival rate to a cluster as

$$\lambda_k = \frac{\lambda_\omega}{\mathcal{N}_C} + \frac{2(\mathcal{N}_C - \mathcal{C}(k))\lambda_\omega}{\mathcal{N}_C} \tag{11}$$

II.4.3 Node Resources

As noted above, there are two classes of nodes that serve to implement the described network topology and protocols. The main sink is responsible for managing the data tasks in the network and providing collected data to an external user. To realize this, the main sink has unlimited power storage such as a hardwired connection or solar power. The main sink also has sufficient communications range to enable the network training described in § II.3.2, although this range is only used for training purposes. To support the described training protocol, the main sink has a directional antenna. To monitor the health of the network, the main sink maintains buffers to record which clusters or wedges are responsive to task requests and those that are not.

The distributed cluster nodes support two roles in the network, cluster head and data node; however, each is capable of supporting either role. That is, there is no hardware differentiation between the cluster head and its subordinate data nodes. The cluster nodes have limited power storage such as a battery, but no mechanism to replenish energy storage once deployed. They also have an omni-directional antenna with limited communications range. This enables the broadcast capability while serving as a cluster head. Other requirements for serving as a cluster head include buffers for storing the QoS requirement, a list of registered nodes, the accumulated data, and routing information for adjacent clusters.

II.4.4 Energy Consumption

To model the life expectancy and energy efficiency of the network model, the power consumption has to be considered. This includes quantifying the power associated with each of the communications necessary to negotiate the protocol illustrated in Figure 5. In the model, the sink has unlimited power and as such, is not penalized for transmitting or receiving messages. The cluster nodes have finite energy storage such that once depleted the cluster node is considered to have failed. At this point, the node is referred to as dead and no longer provides data collection or communication services. A cluster is dead when all of the nodes in the cluster are dead. The following equations account for the power expended by cluster nodes during the various communications of data collection task.

As described above, the process of completing a task includes a number of communications from the sink node initiating the task, the relaying of the request to the target cluster, performing the data collection, and finally returning the data product to the sink [2]. In this process, the penalty to transmit a message is P_{Tx} and is differentiated from the penalty to receive a message, P_{Rx}. The penalty to forward a message is P_{Fw} which includes the power to receive (P_{Rx}) and retransmit (P_{Tx}) the message.

$$P_{Fw} = P_{Rx} + P_{Tx} \tag{12}$$

From this, the power to initiate a request from the sink to the target cluster is the product of the forward penalty and the number of times the message is forwarded.

$$P_{Init} = (\mathcal{C}_t - 1)P_{Fw} + P_{Rx} \tag{13}$$

In (13), $\mathcal{C}_t$ is the corona of the target cluster and is in the range of $[1, \mathcal{N}_\mathcal{C}]$. The argument is $\mathcal{C}_t - 1$ because the target cluster does not forward the message. The additional receipt penalty term accounts for the target cluster after finally receiving the request. When the message is received by the target cluster, the message is broadcast by the cluster head to the data nodes. The penalty to broadcast a *Receive Request* message to a cluster is the same as transmitting from one cluster head to another, P_{Tx}. Hence the penalty of a *Broadcast* is $P_{Broadcast} = P_{Tx}$.

The penalty incurred by the data node to receive the *Broadcast* and send a *Receive Interest* message is equivalent to P_{Fw} for each live data node in the cluster. The cluster head then receives interest from each of the live nodes in the cluster with a penalty to the cluster head to receive the interest being $(Live(C_t) - 1)P_{Rx}$ where $(Live(C_t) - 1)$ is the number of live nodes in the target cluster C_t excluding the cluster head. The total penalty to the cluster for the assessment of interest is

$$P_{Interest} = (Live(C_t) - 1)(P_{Fw} + P_{Rx}) \tag{14}$$

For a given message, m, the QoS constraint is $qos(m)$. In fulfilling the QoS constraint, the penalty for the acknowledgement of interest is

$$P_{Ack} = qos(m)(P_{Tx} + P_{Rx}) = qos(m)P_{Fw} \tag{15}$$

which includes both the cluster head transmission and the data nodes' reception of the *Receive Acknowledge* message. The energy consumed in sending and receiving the data is again proportional to the QoS constraint $qos(m)$ as

$$P_{Data} = qos(m)(P_{Tx} + P_{Rx}) = qos(m)P_{Fw} \tag{16}$$

The penalty for the cluster for performing the task can be aggregated as P_{Task} where

$$P_{Task} = P_{Broadcast} + P_{Interest} + P_{Ack} + P_{Data} \tag{17}$$

The data is then forwarded back to the sink to complete the task with a penalty equivalent to

$$P_{Comp} = \mathcal{C}_t P_{Tx} + (\mathcal{C}_t - 1)P_{Rx} \tag{18}$$

The return data is transmitted by each cluster from the target cluster to the sink however the sink is not penalized for receiving the data as indicated by the $\mathcal{C}_t - 1$ term. This yields a cumulative penalty of

$$P_{Total} = P_{Init} + P_{Task} + P_{Comp} \tag{19}$$

where the first and last terms represent the overhead of propagating the requests and results, respectively, and the remaining term is the power expended by the cluster servicing the request.

II.5 Fault Tolerance

The topic of fault-tolerance in WSN has been extensively researched [6, 8, 9, 11, 13–20, 22–28, 30–35] covering topics of routing and topology control in the context of power efficiency. The sensor nodes may be deployed in harsh or hostile environments leaving the nodes potentially vulnerable to environmentally induced failure or attack [17]. As a result, sensor nodes may be easily damaged or depleted of energy [26] altering the network topology and fragmenting routing paths. This dynamic characteristic of the network is especially critical to routing protocols [16] where energy is lost in transmitting along failed routing paths. As noted above, sensor nodes are not readily replaced or recharged and hence the networks and employed protocols must complete their objectives in the presence of one or more failed nodes. This clearly establishes the value of employing mechanisms and protocols that persist correctly after the onset of network failures. This characteristic is referred to as *fault-tolerance.*

Although significant research has been invested in the topic of FT WSN, terminology has been poorly addressed. This section attempts to remedy this in part by reviewing definitions of the relevant terminology as defined in the general computing community. Avizienis et al. provides an excellent exposition on the progression of the terminology over several decades [4] and is referenced for much of the discussion throughout this section. These definitions then clarify the domain and application of fault-tolerance as part of the proposed recovery mechanism described in chapter III. In the following paragraphs, key terminology, shown in **boldface** characters, is defined.

To begin the discourse on terminology, we first define *reliability* and *fault-tolerance.* **Reliability** reflects a functional units ability to meet performance specifications over a specified period of time, and this is often expressed as a probability or a mean time to failure (MTTF). **Fault-tolerance** is the quality or ability of a functional unit to perform a required task in the presence of some number of faults or errors [2]. The distinction is that the former is an attribute relating to the performance period until a fault is encountered while the latter is with regard to the system's performance in the presence of one or more failed components.

Fault-tolerance is applied to increase the reliability of a system [36]. Avizienis et al. expand the domain of the topic to *dependability* which encompasses availability, reliability, safety, integrity, and maintainability [4]. In this discussion, **availability** is the readiness of a system to provide a service. **Reliability** is "continuity of correct service" [4] or the probability of survival [36], both of which coincide with the previous definition for reliability. **Safety** is the absence of hazards to the user and environment, but is not considered directly relevant to this work as the users are not expected to be in close proximity to the WSN. Also, the safety implications of a WSN are expected to be an aspect of the intended use of a specific WSN and not inherent in WSN in general. Webster's Dictionary [37] defines **integrity** as "an unimpaired condition" which is consistent with Avizienis defining it as the "absence of improper system alterations" [4]. Finally, **maintainability** is the quality of being modified or repaired which is not considered to be part of normal WSN operations or use given the description provided above. Of these, availability, reliability, and integrity are left for consideration.

The three remaining key components of dependability each relate to the correct operation of the system of interest, WSN in this discussion, and lead us to define failures, faults, and errors. As noted above, nodes in a WSN are prone to failure where **failure** is defined as the deviation in the actual performance from the expected or correct performance of the system. Correct operation of the WSN is to respond to task requests with the specified QoS. A failure is when the network does not respond. A **fault** is a defect in the system resulting in an incorrect state [38]. Faults in the reference WSN are insufficient power in the nodes in the first corona, insufficient nodes to support QoS, and collisions. The first is the primary focus of this work, the second is a side effect of either exhaustion or topology, and the last is a function of arrival rate and task distribution. An alternative fault to insufficient energy in the nodes is that the apportionment of nodes to clusters is not suitable for the distribution of traffic. Butler states that "a rigorous definition of an error is non-trivial" however he continues that an error is the "manifestation of the fault in some visible state of the system" [38]. In this work, **error** is the deviation in the state of a system from the correct state, consistent with both Butler and Avizienis [4, 38]. The error in the reference network is lack of response as the system fails.

Avizienis et al. [4] proposes that dependability can be addressed through fault prevention, fault-tolerance, fault removal, and fault forecasting. Fault prevention is the *a priori* preclusion of the occurrence of faults in a system and is considered "almost impossible" to achieve [36]. Fault-tolerance accounts for the expectation of faults through the incorporation of redundancy to mitigate the faults and prevent system failure [36]. Fault removal is the process of modifying the system to correct the fault and is considered maintenance and not germane to this discussion for the reasons noted above. Fault forecasting predicts the occurrence of faults and assesses the consequences but does not mitigate faults and is also beyond the scope of this work. This yields fault-tolerance as the logical pursuit in attempting to improve the reliability, and hence dependability, of a WSN.

CHAPTER III

THEORY

The purpose of this research is to model a fault recovery mechanism in the context of the wireless sensor network proposed by Wadaa et al. [3]. The previous chapter presents the network model and the communications protocol as well key terminology for discussing the observed failure mechanism inherent in the proposed topology. The network is comprised of a distribution of wireless sensor nodes communicating with a central node serving to collect data from the network. The network suffers from exhaustion of power in node clusters closest to the central node resulting in accelerated loss of function, or network failure. A recovery mechanism is proposed to address this and enhance the communications efficiency of the network. The following sections present the the failure mechanism and the proposed recovery protocol.

III.1 Fault Model

The described network suffers from an intrinsic failure mechanism, or fault, resulting from the topology, routing protocol, and finite energy storage in the cluster nodes. The *fault* is an artifact of the central sink concentrating traffic through surrounding clusters in the first corona. This results in a *failure* observed as a loss of service from the affected wedges. The observance of the failure is made through the lack of response, and hence a lack of data, which is an *error*. Given that there is a single sink node, any message destined for a cluster not adjacent to the sink must be forwarded by the clusters between the sink and the target cluster. In other words, all message traffic is concentrated through the inner-most clusters in the first corona. This is significant because it establishes a disproportionate traffic load which increases the rate of energy depletion in the first corona compared to outer coronas. These establish that while all the clusters potentially process the same number of requests, the inner clusters also consume additional energy as a result of conveying traffic on behalf of the outer clusters [2]. Using the arrival rate first defined in (11), note that λ_{tasks} is independent of the corona of the target cluster; however, $\lambda_{forward}$ is not. For a cluster in the first corona, this results in

$$\lambda_{C_1} = \frac{\lambda_\omega}{\mathcal{N}_C} + \frac{2(\mathcal{N}_C - 1)\lambda_\omega}{\mathcal{N}_C} = \lambda_\omega \frac{2\mathcal{N}_C - 1}{\mathcal{N}_C} \tag{20}$$

For a cluster in the outermost corona, there is no forward traffic and the arrival rate reduces to

$$\lambda_{C_\mathcal{N}} = \frac{\lambda_\omega}{\mathcal{N}_C} \tag{21}$$

From (11), the percentage of traffic to a cluster representing the task load is given by

$$\text{Task load} = \frac{\lambda_{tasks}}{\lambda_k} = \frac{1}{2(\mathcal{N}_C - \mathcal{C}(k)) + 1} \tag{22}$$

The percentage of traffic load associated with forwarded traffic is then

$$\text{Forward load} = 1 - \text{Taskload} = \frac{2(\mathcal{N}_C - \mathcal{C}(k))}{2(\mathcal{N}_C - \mathcal{C}(k)) + 1} \tag{23}$$

This demonstrates that as $\mathcal{C}(k)$ increases, the load associated forwarded traffic quickly exceeds the load for communications directed to the cluster.

In the case where nodes are uniformly distributed throughout the network, this failure mechanism is compounded by disproportionate energy available in each cluster. The amount of energy stored in a cluster is proportional to the number of nodes in the cluster, which in turn is proportional to the area covered by the cluster. The area of a cluster, noted as A_c, is given by the difference of the area of the arc bounded by the cluster's outer radius and the area of the arc bounded by its inner radius. For clusters in the first corona, the area of the inner arc is 0 and the area is given by

$$A_c = \frac{1}{2}\theta_\omega(\mathcal{C}_s)^2 \tag{24}$$

For a cluster k, not in the first corona,

$$\begin{aligned} A_c &= \frac{1}{2}\theta_\omega(\mathcal{C}_s\mathcal{C}(k))^2 - \frac{1}{2}\theta_\omega(\mathcal{C}_s(\mathcal{C}(k)-1))^2 \\ A_c &= \theta_\omega\mathcal{C}_s^2(\mathcal{C}(k)-1/2) \end{aligned} \tag{25}$$

where $\mathcal{C}(k)$ is again the corona of cluster k. Note that for the case where the cluster is in the first corona, $\mathcal{C}(k) = 1$ and (25) equates to the (24) making it generally applicable to this network topology. This yields clusters in a wedge increasing in area, and hence power, at a rate of $(\mathcal{C}(k) - 1/2), \mathcal{C}(k) \geq 1$.

Both of these network characteristics lead to the exhaustion of power in the clusters in the first corona. Thus, while the inner clusters are initially at a power disadvantage, the concentration of traffic through them results in premature mortality for the clusters in the first corona. Noting that the first corona is the conduit to transport communications between the sink and outer clusters, once the first corona cluster has failed, the remaining live clusters in the wedge are inaccessible, thus causing the entire wedge to have effectively failed. As the wedges fail, the remaining *live* wedges experience an increase in the arrival rate of the messages for a constant λ_m. Let $\mathcal{N}'_\omega$ denote the number of *live* wedges remaining in the network, then λ'_ω is the arrival rate of messages to the remaining wedges and is given by

$$\lambda'_\omega = \frac{\lambda_m}{\mathcal{N}'_\omega} \tag{26}$$

where

$$\lim_{\mathcal{N}'_\omega \to 0} \lambda'_\omega = \lambda_m \tag{27}$$

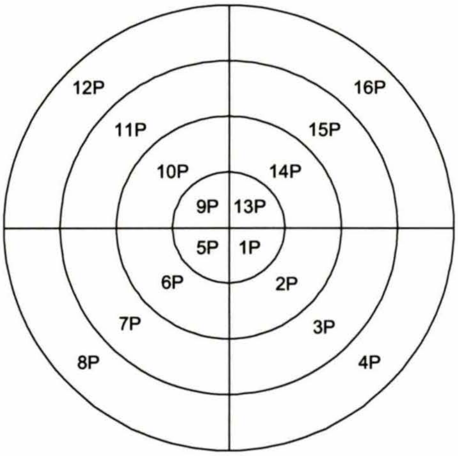
As $\mathcal{N}'_\omega$ approaches 0, the network suffers a cascading failure as traffic is concentrated on the remaining accessible clusters. During this process, the remaining energy in the isolated outer clusters is lost reducing the aggregate network communications efficiency.

III.2 Fault Recovery

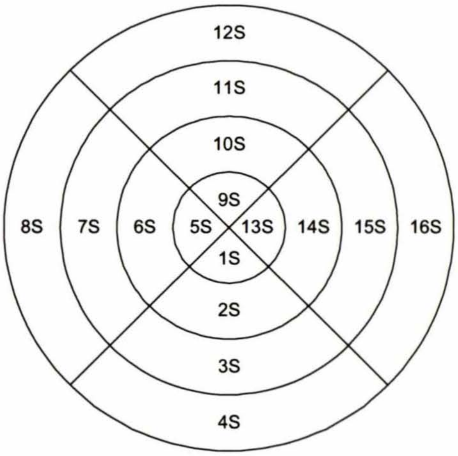
To address the described fault and increase network efficiency, a protocol is proposed to provide access to isolated outer clusters after an inner cluster has failed. The proposed protocol augments the training algorithm to enable a recovery mechanism that utilizes the nodes of neighboring clusters to regain connectivity with outer clusters in a failed wedge. This is made possible by configuring the network as multiple virtual planes that are similarly partitioned but the wedge angles are out of phase by θ_{ω^*} where

$$\theta_{\omega^*} = \frac{\theta_\omega}{\rho}, \rho \geq 1 \tag{28}$$

In (28), ρ is the number of virtual planes implemented as part of the recovery mechanism and the case where $\rho = 1$ realizes the original training described in § II.3.2. For the purpose of discussion, the following assumes ρ to be 2 yielding a two plane configuration. In the two-plane configuration, the virtual planes are referred to as primary and secondary where the primary virtual plane is identical to that described in § II.3.2. This is illustrated in Figure 6 where the primary partitioning is shown in (a) and the secondary partitioning is shown in (b). During initial network training, cluster nodes are mapped into both planes and after clusters fail, a recovery process is initiated in which routing tables in the remaining cluster nodes are updated to reflect the topology of the alternate plane. While the nodes are not physically relocated, they are reorganized such that approximately half of each cluster transitions membership to an adjacent cluster in the same corona of the adjacent wedge. Using Figure 6 as an example, some nodes in cluster 3P will remain in cluster 3 as members of 3S when transitioning to the secondary plane. The rest will become members of cluster 15S. This results in a dead cluster potentially receiving live nodes from a neighboring cluster providing renewed connectivity to clusters in that wedge. More detailed descriptions of the modified training algorithm and the recovery process follow.



(a) Primary plane.



(b) Secondary plane.

FIG. 6: Illustration of primary (a) and secondary (b) network planes.

III.2.1 Modified Training

To support the proposed recovery protocol, the training of the network is modified to realize the two virtual planes. The virtual planes are out of phase by θ_{ω^*} yielding non-congruent views of the network as a means of pseudo-redundancy. The network is now comprised of a primary set of wedges, ω_j, and a secondary set of wedges, ω'_j, where each wedge is subtended by two wedges $\omega^*_m, m \in [1, 2\mathcal{N}_\omega]$. This is a specific form of the the more general $\omega^*_m, m \in [1, \rho\mathcal{N}_\omega]$ where $\rho = 2$. For $\omega^*(n)$ denoting membership of node n in subwedge ω^*_m and n is located with (r_n, θ_n), ω^*_m is defined as

$$\omega^*(n) = \left\lfloor \frac{\theta_n}{\theta_{\omega^*}} \right\rfloor + 1 \tag{29}$$

The algorithm for training the network is updated in Figure 7 to account for θ_{ω^*}. In this process, each node is allocated to primary ($\omega(n)$) and secondary ($\omega'(n)$) wedges and then to primary and secondary clusters.

III.2.2 Recovery Process

The recovery mechanism is the process of switching between virtual planes. In the event of a cluster failure in the first corona, the network initiates a transition

for all $\omega_j^* : j \in [1, \rho\mathcal{N}_\omega], \rho = 2$ **do**
  for all $\mathcal{C}_i : i \in [1, \mathcal{C}_i]$ **do**
    for all $n : n \in [\omega_j^* \cap \mathcal{C}_i]$ **do**

$$\omega(n) \Leftarrow \left\lceil \frac{\omega^*(n)}{2} \right\rceil$$

$$\omega'(n) \Leftarrow \left[\left\lceil \frac{\omega^*(n)-1}{2} \right\rceil \pmod{\mathcal{N}_\omega} \right] + 1$$

$$\mathcal{C}(n) \Leftarrow \left\lceil \frac{r_n}{\mathcal{C}_s} \right\rceil$$

$$primary\_cluster(n) \Leftarrow [(\omega(n) - 1)\mathcal{N}_\mathcal{C}] + \mathcal{C}(n)$$

$$secondary\_cluster(n) \Leftarrow [(\omega'(n) - 1)\mathcal{N}_\mathcal{C}] + \mathcal{C}(n)$$

    end for
  end for
end for

FIG. 7: Algorithm for determining the associations of a node to its primary and secondary wedges, as well as its corona, and cluster.

in network organization to the secondary plane making it the active configuration. After the transition to the newly active configuration, the nodes use the cluster id for the active configuration and the clusters each perform the cluster head selection process. The cluster heads then utilize the active network configuration for routing and data collection.

The recovery process is illustrated in Figures 8 and 9, showing the network state before and after the recovery transition, respectively. The figures illustrate a network, similar to that shown in Figure 6, trained with a primary and secondary plane denoted with 'P' and 'S', respectively. In Figure 8, node 216 is the head of cluster 2P processing task request 0. The requested QoS is two with cluster nodes 85 and 65 supporting the request. Cluster 6 has failed with no live cluster nodes to perform functions of cluster head for data collection or communications relay. Figure 9 illustrates the clusters after a recovery transition. Cluster 2 transitions from the primary plane as cluster 2P to the secondary plane as cluster 2S. The nodes 402, 65, and 92 are no longer in cluster 2 although node 65 has not completed its task. Communications between node 65 and the new cluster head of cluster 2S are permitted only to complete the current task. The cluster head function transitions

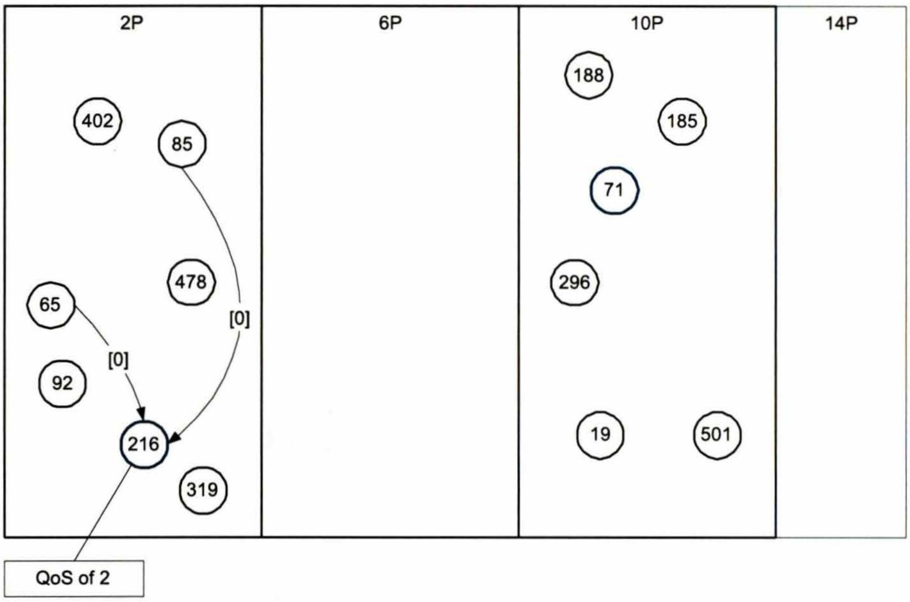


FIG. 8: Partitioned clusters with failed cluster prior to recovery transition.

from node 216 to node 478 and node 478 utilizes the routing table associated with the now active secondary plane of cluster 2. After the recovery transition, cluster 6 transitions from 6P to 6S and subsequently is revived to support traffic. In the figure, cluster 6S is processing task request 1 with a QoS of 3. During the next recovery transition, the clusters will transition back to the primary plane and cluster 6 will return to a failed state.

III.2.3 Hypothesis

The proposed recovery mechanism is expected to improve the network performance over the original protocol. Recovery offers the opportunity to re-establish communications with clusters in a failed wedge and extend the useful life of outer clusters in those wedges. This continued use of the outer clusters for task processing yields a greater network efficiency . It is expected that the inner clusters will experience "brown-out" phases of intermittent communications instead of permanent failures which is considered an acceptable alternative.

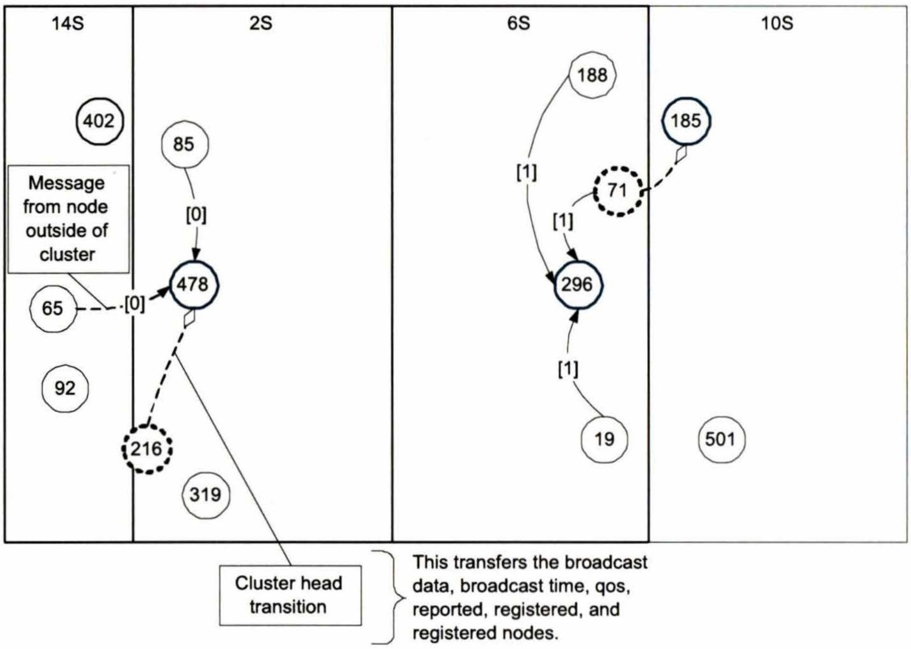


FIG. 9: Partitioned clusters after fault recovery plane transition.

CHAPTER IV

RESULTS

The previous chapters present the background and context of a proposed fault recovery protocol for WSNs. Chapter II develops the network model including topology and communications protocols. Chapter III describes the observed fault and proposes a recovery mechanism. The proposed recovery scheme has been simulated in order to evaluate the hypothesis. This chapter first describes the simulation followed by a presentation of the resultant data.

IV.1 Simulation

The described protocol, WSN Algorithm Simulator (WASim), is implemented in Java. WASim is a discrete event simulator implementing a virtual WSN that can be readily configured for a number of network characteristics such as the number of nodes, node energy, and node distribution. While WASim has been initially developed to simulate a network that has been trained using the protocol described by Wadaa et al. [3], it could easily be extended to support other topologies and protocols by replacing a small subset of classes (i.e. Java files) in the program structure. The following briefly describes the structure and operation of the simulator as well as the configuration of the simulated network to establish a clear context for the data results presented in § IV.2.

IV.1.1 Program Structure

The WASim program utilizes an object-oriented approach to implement both the network elements as well as the simulation framework. The program Main serves to provide a program entrance point and parse configuration options as well as allocate and configure the primary components of the program, the simulator and network object instances. The simulator is responsible for managing time and processing the events in the simulation. The simulator generates all traffic in the network which is routed through the sink to the clusters. To facilitate the scheduling and processing of events, the simulator also manages a buffer of currently active messages which dictates the operations to be executed by network elements. While this approach lacks the fidelity of modeling buffers in the nodes storing messages between transmissions, it facilitates observability during testing of the simulation without detracting from the intent and performance of the simulation. The network contains the network nodes stored as an array of Node objects as well as an array of Cluster objects which serve to organize the nodes. In each cluster, as described previously, one node serves the function of cluster head while the rest facilitate task requests as data nodes. Periodically, the network initiates an election process in which a new cluster head is selected, at random, from the available live nodes in each cluster. The remaining nodes operate as data nodes until selected to be the cluster head during a subsequent election process.

IV.1.2 Traffic Generation and Routing

The traffic generation model is simple, yet fundamental to the operation and performance of the network. All traffic is originated from the sink node as a request for a target cluster to collect data. To simulate this functionality, the simulator takes as a configuration parameter, a message arrival rate λ_m which specifies the number of requests to generate and schedule each simulation cycle. Then, for each simulation cycle, λ_m messages are introduced into the network as requests from the sink to be routed to the target cluster. Each request issued by the sink includes both a destination address and a QoS requirement.

The target cluster is selected randomly from a list of live wedges. The simulator maintains a list of live wedges that is first populated when the nodes are distributed and the network is trained. Then, as clusters in the first corona fail, the sink can no longer communicate with other clusters in the wedge and the wedge is considered dead and removed from the list. In practical applications, this could be implemented by simply monitoring for loss of communication with clusters in the first corona, which would indicate loss of the entire wedge.

During execution of a simulation cycle, the simulator conveys the buffered messages from the source node to the destination node. As the messages are received by a node, the node performs a routing function based on the type of message and its destination address. For cluster heads, the routing function is either to forward messages to the next cluster or broadcast a request for data collection to its cluster. The data nodes do not perform routing functions. Rather, if a received message is a request, an interest message is returned to the cluster head. If an acknowledgment is received, data is returned. The cluster head acknowledges enough interest requests to satisfy the QoS requirement for the task and when sufficient data has been collected, the aggregated data product (an average in the case of the simulation) is returned to the sink, routed by intermediate cluster heads.

For each transmission and reception, the node deducts either P_{Tx} or P_{Rx}, respectively, from its energy store. The values for the P_{Tx} and P_{Rx} parameters default to 1 and 0.55, respectively; however either can be set through configuration parameters. The default values were selected to represent typical energy losses due to transmitter and receiver electronics, amplifier dissipation, size of the message (in bits), and the distance the message traverses [2]. These values have been normalized to a unit transmit energy as a simplification. The ratio between the transmit and receive energy has been maintained for consistency with the prior work by Elmiligui [2]. As the simulation proceeds, the node's energy is depleted and the node fails. The cluster and network are then *notified* as a simplification to support resource tracking and statistics.

Periodic communication losses are network failures and degrade communications efficiency. In the simulation, there are four faults resulting in the loss of a message: insufficient QoS, a collision, data node failure, and cluster head failure. Each received message includes a QoS requirement that may exceed the capacity of the target cluster. In this case, the message is dropped due to insufficient nodes available to support the requested QoS. This results in a "fail silent" data error which is easily detected by the sink and user application. A collision occurs when a request is received by a cluster head that has a previous task still pending. The cluster head has no way to determine whether the pending request will complete, but favors it and drops the new request. All requests to the cluster are dropped until the pending request is completed or cleared, however messages will still route through the cluster. For data node failures, a loss can occur if the data node fails during the period between sending interest and returning data. In this situation, the cluster head is expecting a response from a failed node leaving the cluster head blocked with a pending task. Any subsequent task received by the blocked cluster head results in a collision until the pending request is cleared. During the next election process, the task information is transferred to the new cluster head indicating that a request is pending, subsequently blocking the new cluster head. A pending request is forwarded at most one time before being dropped, freeing the cluster to service subsequent task requests. For the cluster head failure, all communications to and through the cluster are lost until the next election process. During the election process, a new cluster head is selected, but the interrupted tasks are not continued.

IV.1.3 Configuration Options

The simulation can be configured via a number of command line options and compile time parameters, each of which affect the performance and operation of the network. The default values are listed in Table 1, of which only a few were modified during the performance of the simulations. The first three parameters configure the energy for the network and are normalized to a unit transmission penalty. The initial energy is 100 times the normalized transmission penalty while the receipt penalty is 0.55, maintaining a ratio consistent with the work performed by Elmiligui [2]. The distribution option selects between uniform (0) and proportional (1) distributions. The former evenly distributes nodes over the network area while the latter equally distributes nodes among the clusters. The wedge and corona count options set the organization of the network. The recovery option enables the proposed recovery mechanism in the simulation while the iteration option determines the number of times the simulation is executed. In each configuration, the simulation is run until the sink cannot communicate with any clusters and the network has failed.

TABLE 1: Default simulation parameters.

Parameter	Label	Default Value	Description
Initial energy	e	100	Energy stored in the node at the start of the simulation
Transmit Penalty	t	1	Penalty to transmit a message
Receive Penalty	r	0.55	Penalty to receive a message
Nodes	N	512	Nodes in the network
Distribution	D	0 (uniform)	Set the node distribution
Wedge count	w	8	Number of wedges in the network
Corona count	c	4	Number of wedges in the network
Recovery	R	0 (no)	Boolean to select recovery
Iterations	i	1	Times to repeat the simulation

IV.2 Simulation Results

The simulation results quantify the performance of the recovery mechanism in the context of both uniformly and proportionally distributed networks. The primary parameters varied during the performance of the simulations are:

- the network size in nodes varied from 512 to 8192;
- the nodes distributed uniformly and proportionally; and
- the wedge count varied from 4 to 12 wedges

Each of these parameters are varied for configurations with and without recovery generating data for the proposed evaluation criteria of network longevity, communications efficiency, and utilization. The data is presented in the following sections with a comparative performance analysis of the proposed recovery mechanism.

IV.2.1 Characterization of a Uniformly Distributed Network

The first network distribution evaluated is the uniform distribution in which nodes are distributed uniformly over the service area of the network. The simulation data for a single virtual plane is presented with the evaluation criteria of network longevity (simulation cycles), communications efficiency, and energy utilization which are then used to characterize the aggregate network efficiency. This is followed by a presentation of the data for two virtual planes implementing the recovery protocol. A comparative analysis between the two data sets is then presented concluding this section.

IV.2.1.1 Network Response to Initial Energy

This section presents data results characterizing the effect of initial energy on the longevity of the uniformly distributed single-plane network. Most notably, the energy is varied from 100 units to 800 units for wedge counts of 4, 8, and 12. The complete list of simulation parameters is given in Table 2.

TABLE 2: Configuration parameters for simulation of a uniformly distributed network varying initial node energy.

Parameter	Label	Value
Initial energy	e	100, 200, 400, 800
Nodes	N	512-8192
Distribution	D	0 (uniform)
Wedge count	w	4, 8, 12
Recovery	R	0 (no)
Iterations	i	100

The first metric discussed is network longevity with the plot for the eight wedge configuration shown in Figure 10. The plot shows the average network lifetime, in simulation cycles, for initial energies of 100, 200, 400, and 800 units per cluster node. This is plotted against the size of the network varying from a 512 node network to an 8192 node network. Increasing either the initial energy in the nodes or the

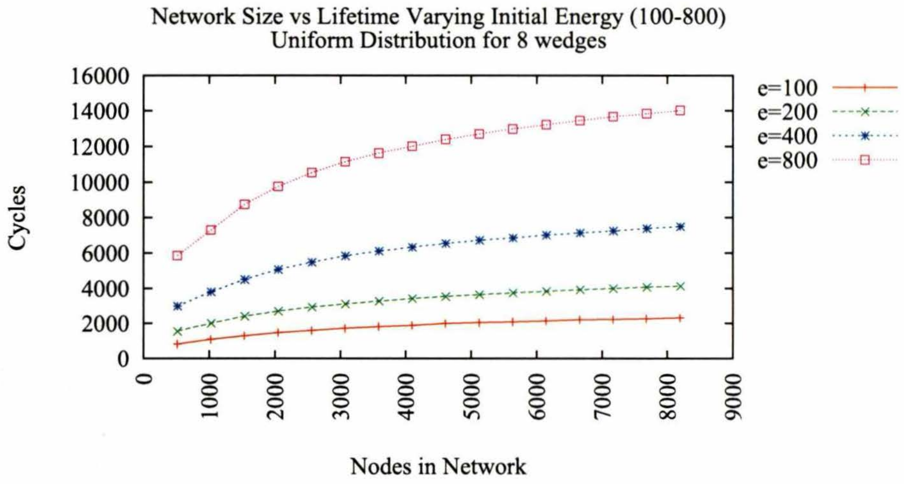


FIG. 10: Network longevity for uniformly distributed 8-wedge networks with initial energy of 100, 200, 400, and 800 units per node.

size of the network increases the networks initial energy capacity. As illustrated in Figure 10, this results in extending network life. The increase in network life is nearly linear with increasing initial node energy. Note though, that as the network become significantly larger (several thousand nodes) the benefit tapers off approaching an apparent upper bound in the operational life of the network. In all test cases, for large networks, only the clusters in the first corona failed prior to network failure, which is the impetus for this research. Figures 11 and 12 demonstrate a similar trend for configurations of 4 and 12 wedges, respectively. These indicate that while the longevity of the network depends on the initial amount of energy in the nodes, the general trend is consistent and independent of the initial energy and the number of wedges. This is noteworthy because the proposed mechanism is applicable to the end of the operational period beginning at the onset of failure. With this relationship established, any of the configurations can be used to evaluate the performance of the proposed recovery mechanism for a uniformly distributed WSN. For the remainder of the data presented, the simulation is performed with an initial energy of 100 units (e=100), unless otherwise noted. This allows the discussion to focus on the

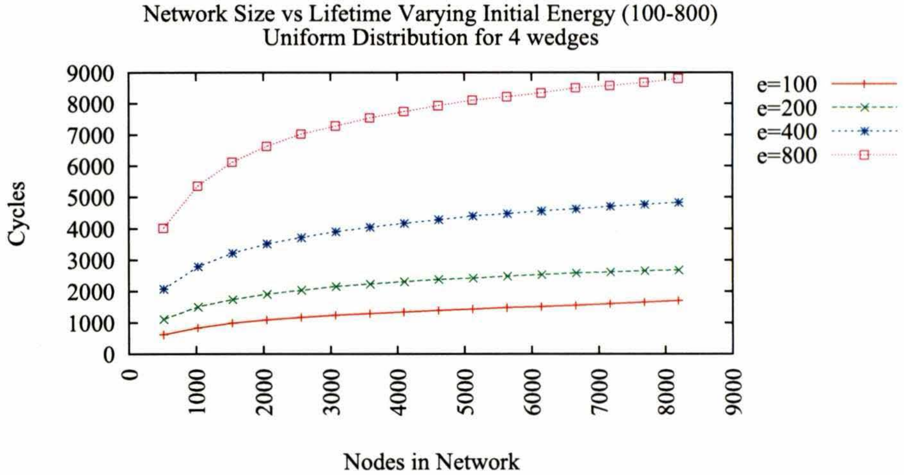


FIG. 11: Network longevity for uniformly distributed 4-wedge networks with initial energy of 100, 200, 400, and 800 units per node.

end-of-life portion of the network operation as it transitions from operational to failure.

IV.2.1.2 The Uniformly Distributed Single-Plane Network

This section presents the configuration and results for a uniformly distributed network without the proposed recovery mechanism. Table 3 lists the configuration parameters for the results presented in this section with the primary parameters of interest being the variance of wedge counts from 4 to 12 and nodes from 512 to 8192. The default values are presented in § IV.1.3 and are not listed.

Figure 13 depicts the variation of network size and wedges plotted against the resultant average cycle life. Again, the trend of diminishing returns of adding nodes to the network is observed. It is interesting to note that the operational life of the network significantly increases in response to varying the number of wedges. This is due to two factors: 1) the burden on the cluster head of communicating with data nodes, and 2) the reduced traffic per wedge. In the former case, the size of each cluster decreases as the number of wedges increases for a given network size. This

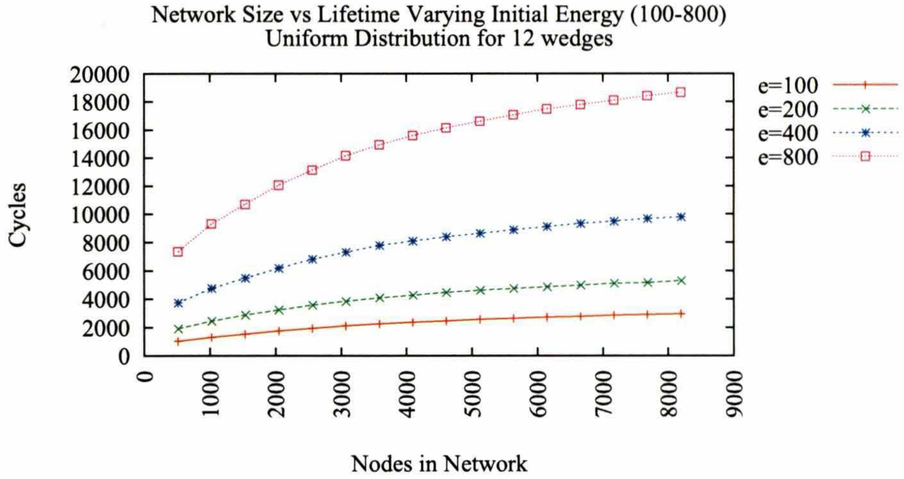


FIG. 12: Network longevity for uniformly distributed 12-wedge networks with initial energy of 100, 200, 400, and 800 units per node.

in turn reduces the number of messages received by a cluster head from data nodes interested in supporting a data task. The latter case is the result of traffic from the main sink being randomly distributed to the network clusters. As the number of wedges increases, the probability that a message is destined for a given wedge decreases, reducing the average load on the wedge and increasing its operational life.

At this point, the performance of the network is discussed in terms of the communications efficiency and energy utilization. The communications efficiency, $\mathcal{E}_c$, is the ratio of the number of completed tasks, m_c, to the product of the total number of attempts, which is the product of the message arrival rate λ_m and the duration of time of operation t. For this analysis, the t is the life of the network yielding a communications efficiency given by

$$\mathcal{E}_c = \frac{m_c}{\lambda_m t} \tag{30}$$

The energy utilization is the percent of energy consumed during the operational life of the network. The data for these metrics are plotted in Figures 14 and 15. The

TABLE 3: Configuration parameters for simulation of a uniformly distributed network.

Parameter	Label	Value
Initial energy	e	100
Nodes	N	512-8192
Distribution	D	0 (uniform)
Wedge count	w	4-12
Recovery	R	0 (no)
Iterations	i	100

plots represent the communications efficiency and energy utilization for a uniformly distributed network with organization varying from 4 to 12 wedges.

In Figure 14, it can be seen that efficiency tends to increase with the number of wedges but not with the number of nodes. For lower wedge count configurations, the probability of a collision increases, resulting in dropped messages and reduced network efficiency. As the network becomes larger, clusters suffer from two failure mechanisms that result in degraded efficiency. The first is due to the large number of data nodes overwhelming the cluster head during tasks and accelerating the cluster head's exhaustion leaving the cluster unable to service requests until a new cluster head is selected during the next election cycle. The second is due to the increasing probability that a data node that has responded to participate in a task is selected by the cluster head and fails prior to returning data, preventing the task's completion. Each of these contribute to decreased network efficiency.

Figure 15 depicts the average percentage of energy utilized by the uniformly distributed network for wedge counts varying from 4 to 12 wedges. As shown, the utilization is also dependent on the wedge configuration as well as the number of nodes in the network. Each of the wedge count configurations quickly increases to its respective upper bounds. For the case where the wedge count is 4, the utilization reaches a maximum and then slowly decreases. This is related to the decrease in network efficiency and is attributed to the incompletion of most of the tasks.

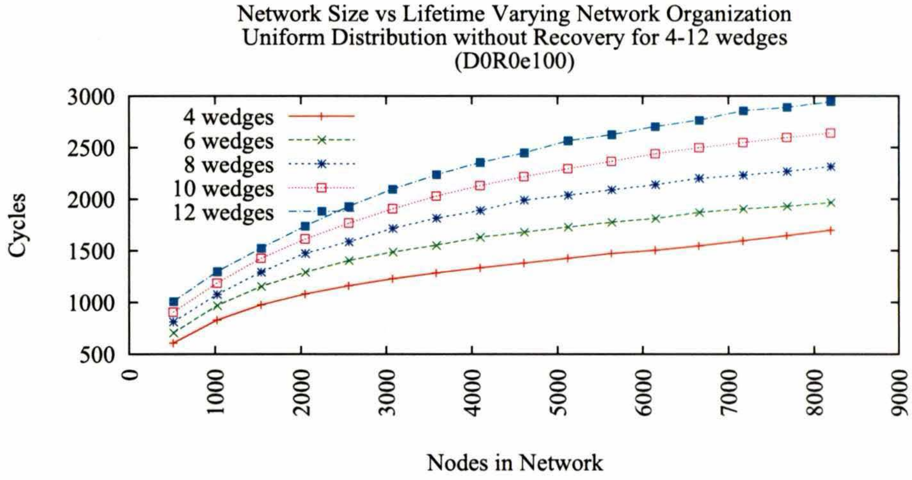


FIG. 13: Network life for uniform node distribution in single-plane configuration.

Note that beyond the 1024 node network, less than 50% of the messages complete due to either collisions or node failures during a task, either during the collection phase or the return routing. As the number of wedges in the network increases, the utilization increases at a decreasing rate with the number of nodes. In fact, for the network divided into 12 wedges, the range of 512 through 2560 nodes enjoys the lowest utilization and highest efficiency. For the larger networks, the utilization increases both with network size and the number of wedges; they do so, however, at a decreasing rate. For the case where the network is 8192 nodes, the benefits of increasing the number of wedges from 10 to 12 result in increases in the life of the network by 12% and efficiency by 21% at the expense of less than 1% in energy utilization.

IV.2.1.3 Recovery in a Uniformly Distributed Two-Plane Network

In this section, the results are presented for the simulation of a two-plane network with recovery. The following paragraphs introduce the results describing the network longevity, efficiency, and energy utilization. The simulation parameters are listed in

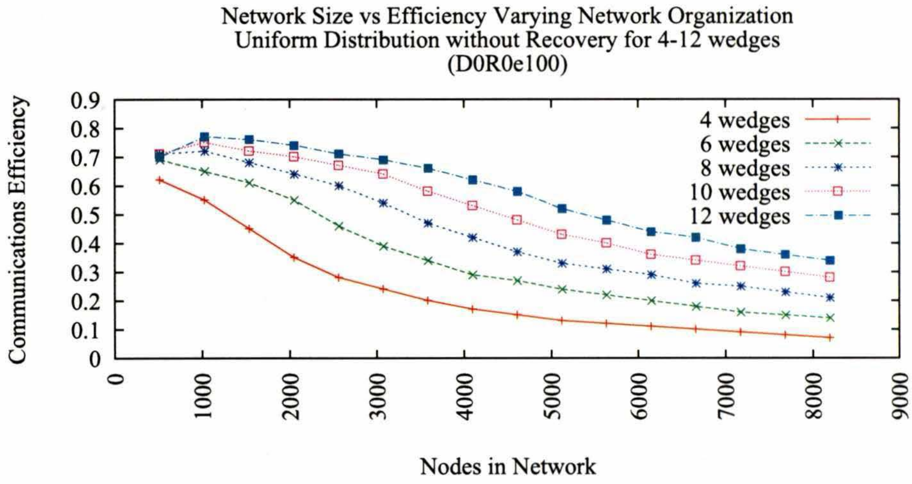


FIG. 14: Network communications efficiency for uniform node distribution in single-plane configuration.

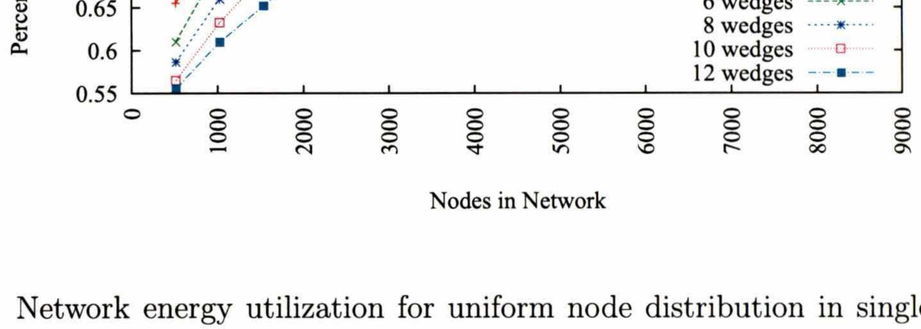


FIG. 15: Network energy utilization for uniform node distribution in single-plane configuration.

TABLE 4: Configuration parameters for simulation of a uniformly distributed network with recovery.

Parameter	Label	Value
Initial energy	e	100
Nodes	N	512-8192
Distribution	D	0 (uniform)
Wedge count	w	4-12
Recovery	R	1 (yes)
Iterations	i	100

Table 4 with only the recovery option changing from the previous configuration.

The resulting cycle life, efficiency, and utilization for the recovery protocol applied to the uniform distribution are shown in Figures 16, 17, and 18, respectively. In each of these, the trends are identical to those described in § IV.2.1. This is because the implementation of the recovery mechanism does not affect the general communications protocols. Furthermore, Figure 16 demonstrates the benefit of increasing the wedge counts. As observed for the single-plane configuration, the communications efficiency decreases as the network becomes large as shown in Figure 17. The average utilization, shown in Figure 18, is consistent with the data in Figure 15. From each of these, the 12-wedge configuration has the longest operational period with the highest communications efficiency. Also note, that for large networks, the average utilization is close to that for the 8 and 10-wedge configuration.

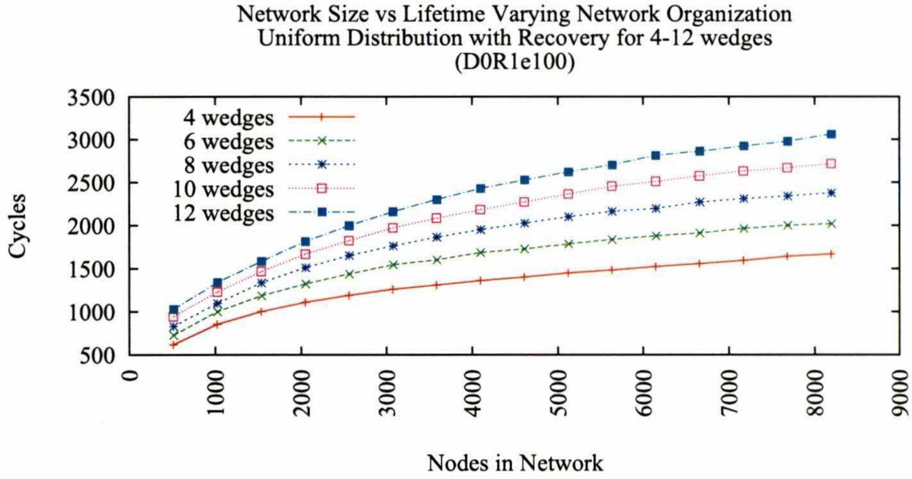


FIG. 16: Network life for uniform node distribution in dual-plane configuration.

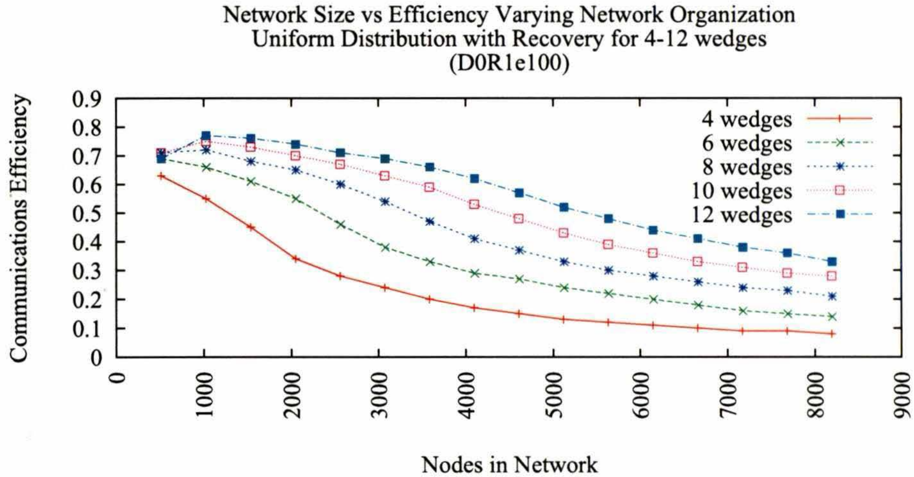


FIG. 17: Network communications efficiency for uniform node distribution in dual-plane configuration.

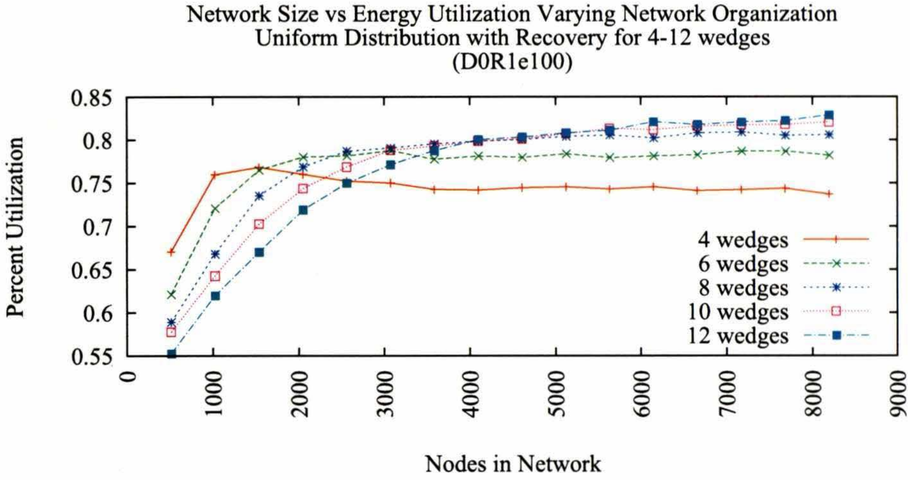


FIG. 18: Network energy utilization for uniform node distribution in dual-plane configuration.

IV.2.2 Analysis of the Uniform Distribution

Figures 19, 20, and 21 plot the recovery protocol data with the corresponding data from the single-plane configuration for the uniform distribution. Figure 19 indicates a consistent increase in network life as a result of the recovery mechanism. The network communications efficiency is shown in Figure 20 to be marginally affected with the efficiency decreasing by at most 0.7%. For the constraining case of 4 wedges, the efficiency is seen to increase by 1.6%, but is still less than 10% for the large networks. For the network configurations of 6 or more wedges, there is a clear increase in the operational life of the network. Recall that this represents the portion of the operational life during the onset of network failure. The resulting increase yields only a few percent increase ranging from an average of 1.4% for the 4-wedge configuration to an average of 3.2% for the 12-wedge configuration. The plots in Figures 20 and 21 indicate that the effect on the efficiency is negligible, but at the expense of increased utilization. The data from the plots are summarized in Table 5 as relative increases in the average performance metric for the various wedge configurations. The recovery mechanism incurs an average increase of 2.4% in utilization in the 12-wedge configuration to achieve an average increase of 3.2% in network life. However, the remaining energy in the network after network failure is effectively lost. The recovery protocol prolonged operational life of the network providing access to the outer clusters without degrading the efficiency of communications. This result indicates an improvement, albeit a small one, in the aggregate network efficiency.

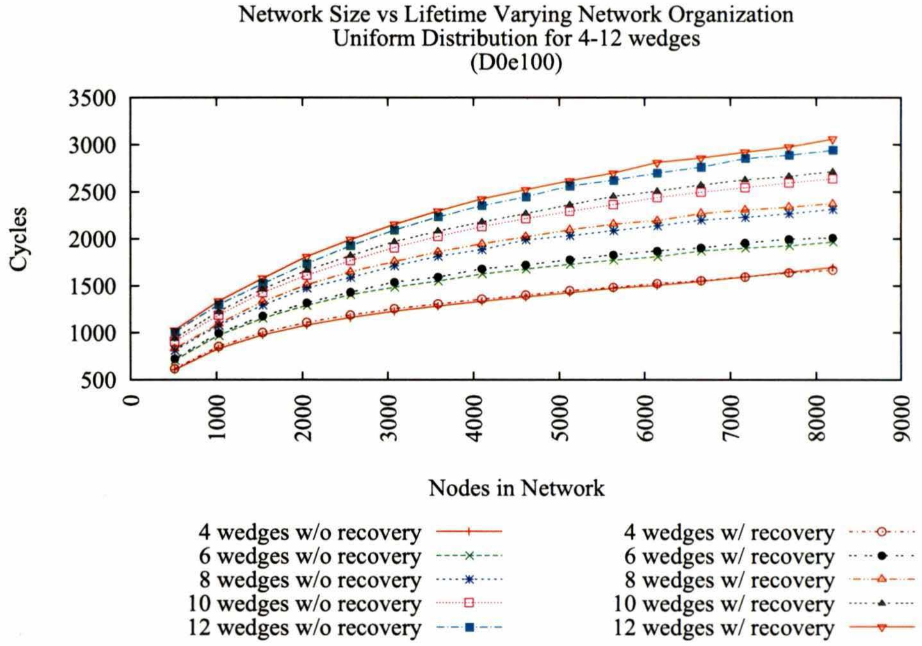


FIG. 19: Network life for uniform node distribution in single-plane and dual-plane configurations.

TABLE 5: Average percent increase associated with recovery protocol in a uniformly distributed network with 4 to 12 wedges.

Wedges	Cycles	Utilization	Efficiency
4	1.4%	3.2%	1.6%
6	3.0%	3.1%	-0.3%
8	2.9%	2.7%	-0.7%
10	3.1%	2.5%	-0.7%
12	3.2%	2.4%	-0.5%

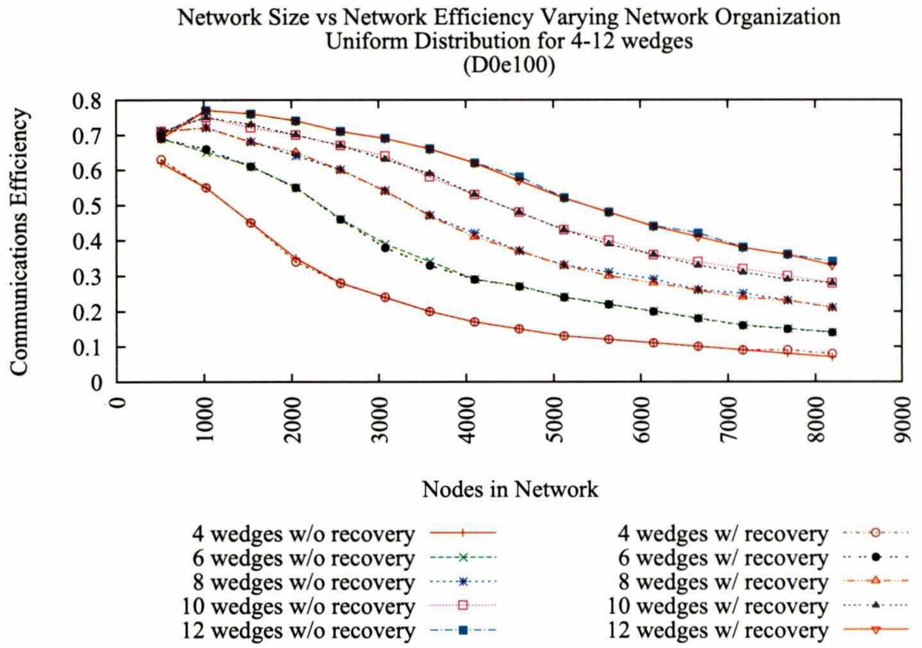


FIG. 20: Network communications efficiency for uniform node distribution in single-plane and dual-plane configurations.

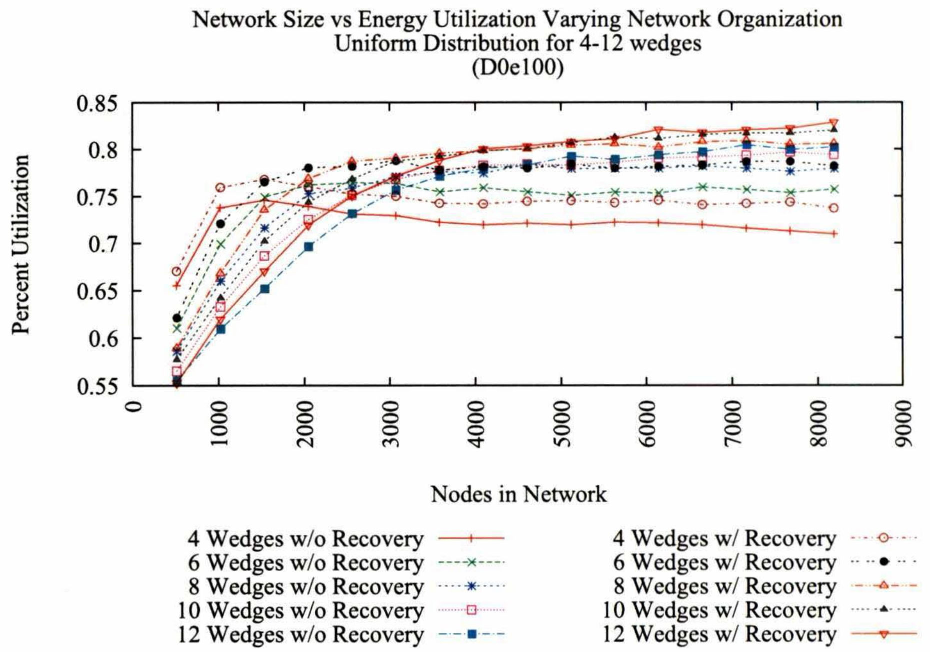


FIG. 21: Network energy utilization for uniform node distribution in single-plane and dual-plane configurations.

IV.2.3 Characterization of a Proportionally Distributed Network

This section evaluates the proportionally distributed network model where nodes are distributed equally among the clusters in the network. The simulation data for a single virtual plane is presented with the evaluation criteria of network longevity, communications efficiency, and energy utilization. The data is then presented for two virtual planes implementing the recovery protocol. These are followed with an analysis of the recovery mechanism in a proportionally distributed network.

IV.2.3.1 Network Response to Initial Energy

As with the uniformly distributed network, the initial energy must be considered. This section presents data results characterizing the effect of initial energy on the longevity of the proportionally distributed single-plane network. Most notably, the energy is varied from 100 units to 800 units for wedge counts of 4, 8, and 12. The complete list of simulation parameters is given in Table 6.

TABLE 6: Configuration parameters for simulation of a proportionally distributed network varying initial node energy.

Parameter	Label	Value
Initial energy	e	100, 200, 400, 800
Nodes	N	512-8192
Distribution	D	1 (proportional)
Wedge count	w	4, 8, 12
Recovery	R	0 (no)
Iterations	i	100

For the network with proportionally distributed nodes, the trends are not consistent with the uniform distribution for the low energy and low wedge count configurations. Figure 22 displays similar trends as those shown above in Figure 10 with an increase in network lifetime as the amount of initial energy increases. Examining the 4-wedge configuration in Figure 23, a strange phenomenon occurs for large network sizes beyond 2000 nodes. This is suspected to represent a threshold of linearity for the network configuration. In this configuration, the initial energy is overly constraining for such a large network. The fewer wedges results in overwhelmingly large clusters that inundate the cluster head and drastically reduce the efficiency of network communications. The result is degraded performance for an unexpectedly prolonged period of operation.

Figure 24 depicts the data for the 12-wedge configuration which appears to adhere to the previous trends with the network benefiting from increases in either initial energy or the number of wedges. The proportional distribution of nodes results in more energy in the first corona, postponing the exhaustion of the inner corona and the resultant failure of the network due to segregation of the network sink from the outer clusters. The two configurations of 8 and 12 wedges indicate a consistent trend as observed for the uniform distribution. The large network with 4 wedges however, does not. Re-examining the uniform distribution, Figure 25 shows the results for a uniformly distributed, 4-wedge configuration varying the size of the network from 1024 to 16384 nodes. It seems that given a sufficiently large number of nodes, the network life becomes linearly proportional to the number of nodes in the network; however, it is effectively useless with an average communications efficiency less than 10% (see § IV.2.1). This, contrasted with the observed longevity realized by increasing the wedge count, makes the large network partitioned into 4 wedges an impractical option. The data is included for completeness.

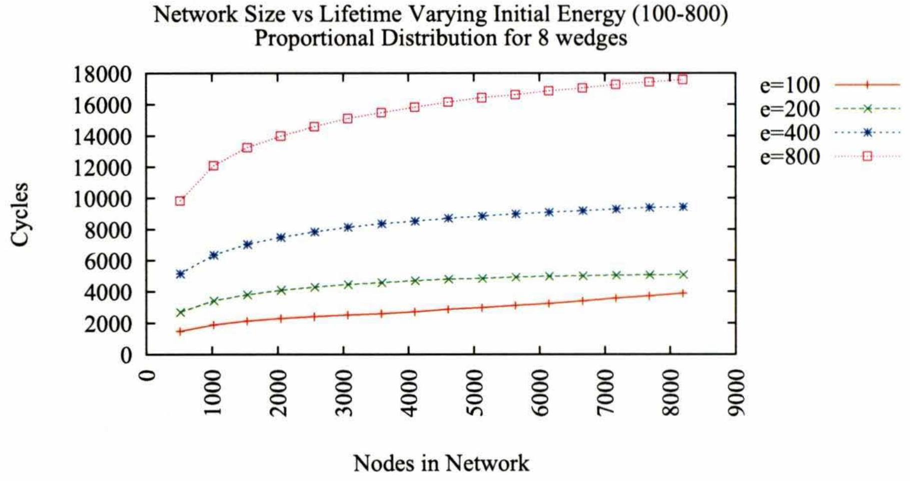


FIG. 22: Network longevity for proportionally distributed 8-wedge networks with initial energy of 100, 200, 400, and 800 units per node.

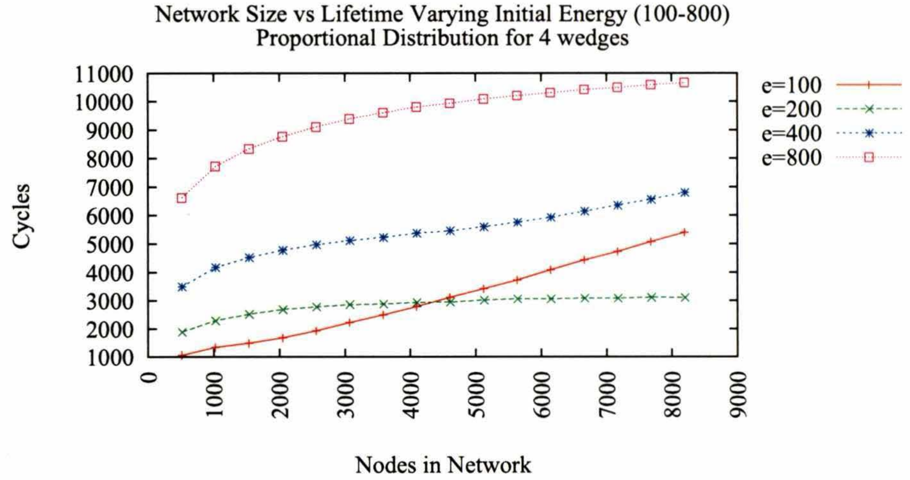


FIG. 23: Network longevity for proportionally distributed 4-wedge networks with initial energy of 100, 200, 400, and 800 units per node.

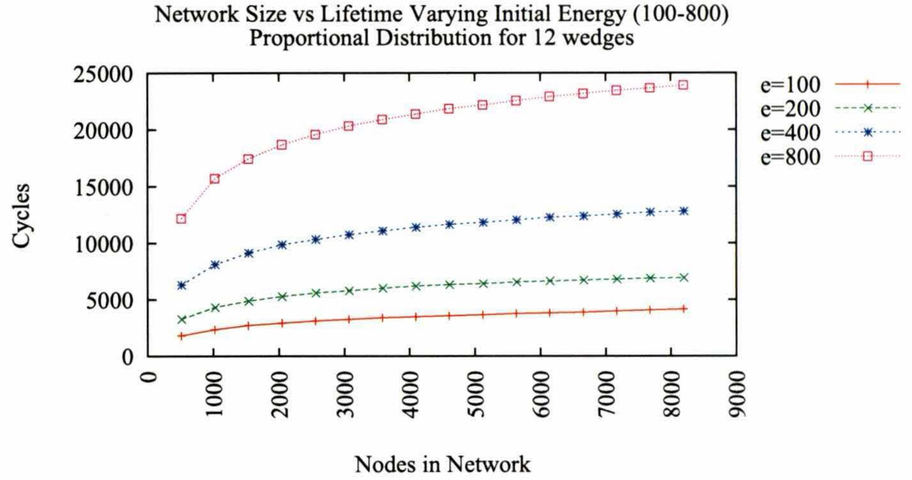


FIG. 24: Network longevity for proportionally distributed 12-wedge networks with initial energy of 100, 200, 400, and 800 units per node.

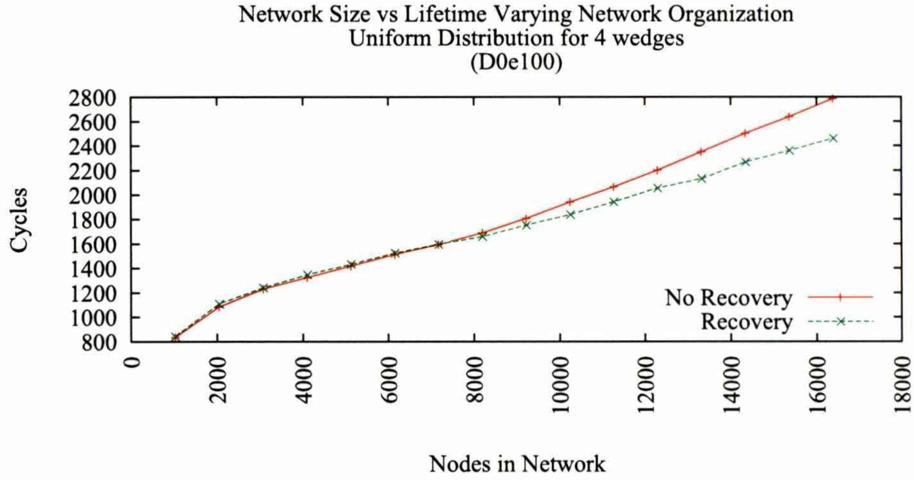


FIG. 25: Network longevity for large uniformly distributed network.

IV.2.3.2 The Proportionally Distributed Single-Plane Network

TABLE 7: Configuration parameters for simulation of a proportionally distributed network.

Parameter	Label	Value
Initial energy	e	100
Nodes	N	512-8192
Distribution	D	1 (proportional)
Wedge count	w	4-12
Recovery	R	0 (no)
Iterations	i	100

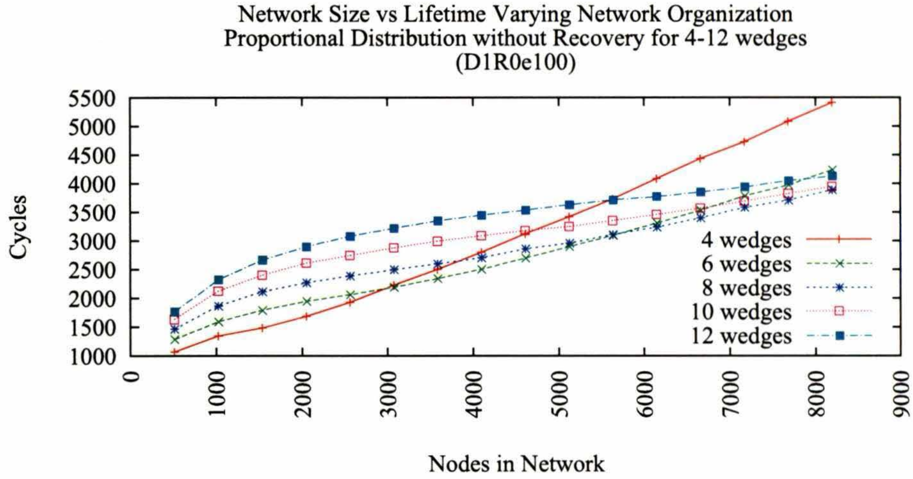


FIG. 26: Network life for proportional node distribution in single-plane configuration.

In characterizing the proportionally distributed network, the single virtual plane configuration is considered first. The parameters for the simulation are listed in Table 7 with the distribution set to proportional and the recovery protocol disabled for wedge counts varied from 4 to 12. Examining the network response for these configurations, Figure 26 appears to support the theory of a linearity speculated in § IV.2.3.1. Obviously the 4-wedge and 6-wedge cases follow this trend while the 8 and 10-wedge cases appear to begin to enter the linear region.

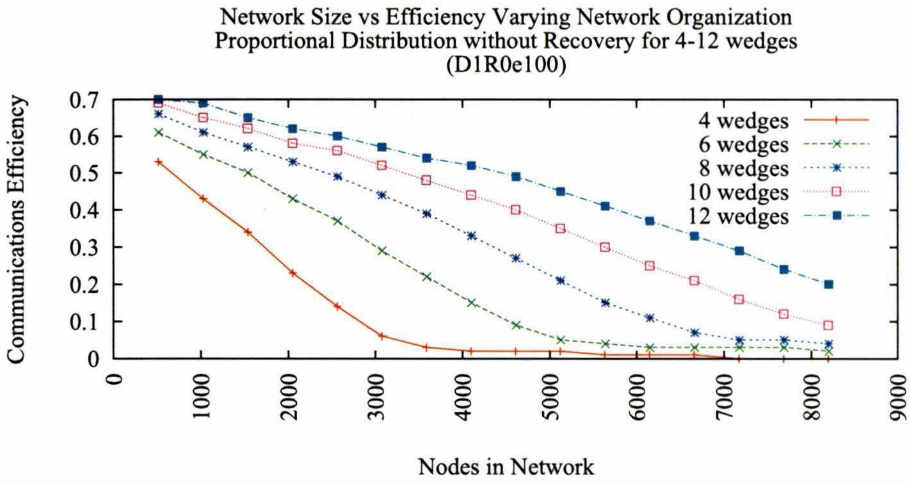


FIG. 27: Network communications efficiency for proportional node distribution in single-plane configuration.

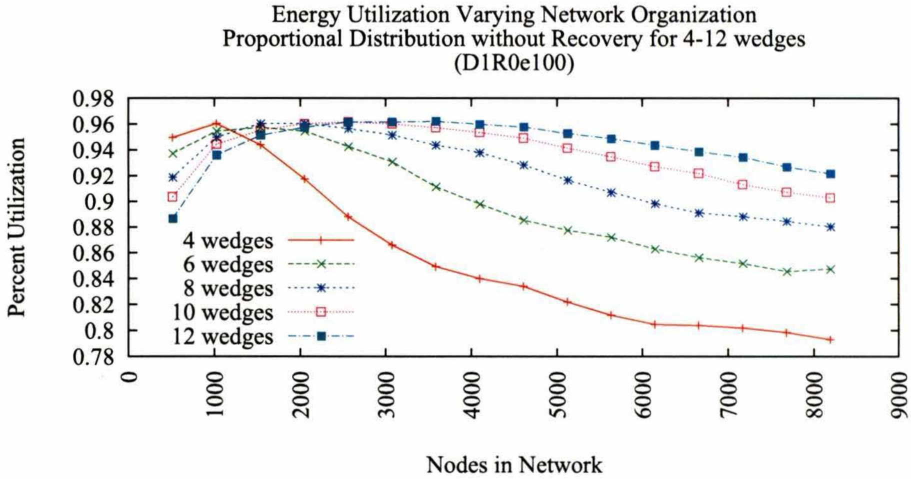


FIG. 28: Network energy utilization for proportional node distribution in single-plane configuration.

Figure 27 illustrates the degradation in communications efficiency as the number of nodes gets large. Comparing the plots for the proportionally distributed network to those of the uniformly distributed network in Figures 13, 14, and 15, there is a significant reduction in efficiency for comparable configurations. The efficiency shown in Figure 27 and the utilization shown in Figure 28 both indicate a severe degradation in performance for the 4-wedge configuration, evidenced by reductions in both communications efficiency and utilization. It is clear that while the large network is operational for a long period of time, the efficiency of the 4-wedge network is negligible beyond 3000 nodes and as well for the 6-wedge configuration beyond 5000 nodes. In fact, Figure 29 illustrates the issue for the 4096 node case. In this plot, each data point represents the completion of a task at the time given on the x-axis. This allows the efficiency to be traced over the operational life of 4 and 8-wedge networks in uniform and proportional distributions. In the figure, both of the proportionally distributed networks operate longer than their uniformly distributed counterparts. However, there are significant efficiency losses due to the size of the clusters, especially for the 4-wedge case. This is due to failures in the network, especially the cluster heads. When a cluster head fails, the cluster is no longer able to route communications until a new cluster head is selected. If the failed cluster head is in one of the inner clusters, the entire wedge is temporarily lost further diminishing the already low communications efficiency. This also explains the curiously prolonged life for the low wedge count configuration. For the uniformly distributed network, the communications are more consistent though there are periodic communications losses near the end of operational life and the inner corona soon fails due to the limited number of nodes to support prolonged activity.

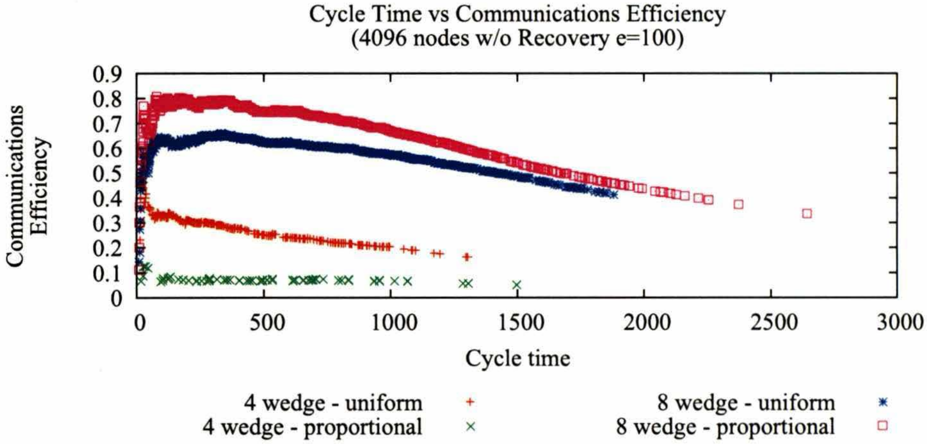


FIG. 29: Efficiency traces for 4 and 8-wedge uniformly and proportionally distributed network configurations.

IV.2.3.3 Recovery in a Proportionally Distributed Two-Plane Network

In this section, the results are presented for the simulation of the recovery mechanism in a proportionally distributed network with two virtual-planes. The following paragraphs introduce the results describing the network longevity, efficiency, and energy utilization. The simulation parameters are listed in Table 8.

TABLE 8: Configuration parameters for simulation of a proportionally distributed network with recovery.

Parameter	Label	Value
Initial energy	e	100
Nodes	N	512-8192
Distribution	D	1 (proportional)
Wedge count	w	4-12
Recovery	R	1 (yes)
Iterations	i	100

For the proportionally distributed network, the results are again similar to the single-plane configuration without recovery. The network life is plotted in Figure 30 demonstrating the same trends. For the 4 and 6-wedge configurations, the network experiences a significantly different trend than for those with higher wedge counts. This is suspected to be the result of the phenomenon described in § IV.2.3.1. Considering the efficiency depicted in Figure 31, the 4 and 6-wedge configurations are not considered plausible configurations for a network with more than 3000 or 5000 nodes, respectively. Figure 32 contains a plot of the utilization of energy for the various wedge configurations. As with the single-plane configuration, there is a dramatic reduction in utilization for the 4-wedge configuration due to the low communications efficiency. As the number of wedges approaches 12, the utilization becomes less dependent on the number of nodes in the network. The efficiency, which improves as the number of wedges increases, still decreases as the network becomes large. This indicates that the network would likely benefit from greater segmentation from increasing wedge counts or possibly corona counts, the latter of which has not been explored by this work.

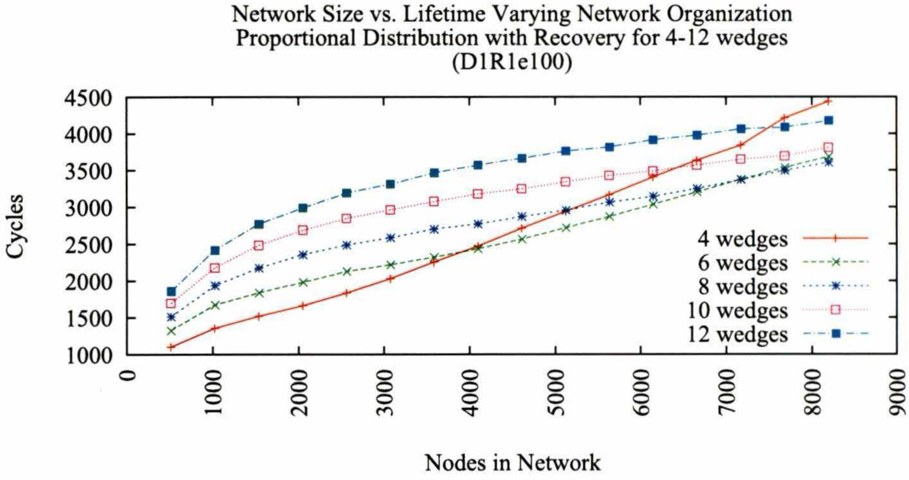


FIG. 30: Network life for proportional node distribution in dual-plane configuration.

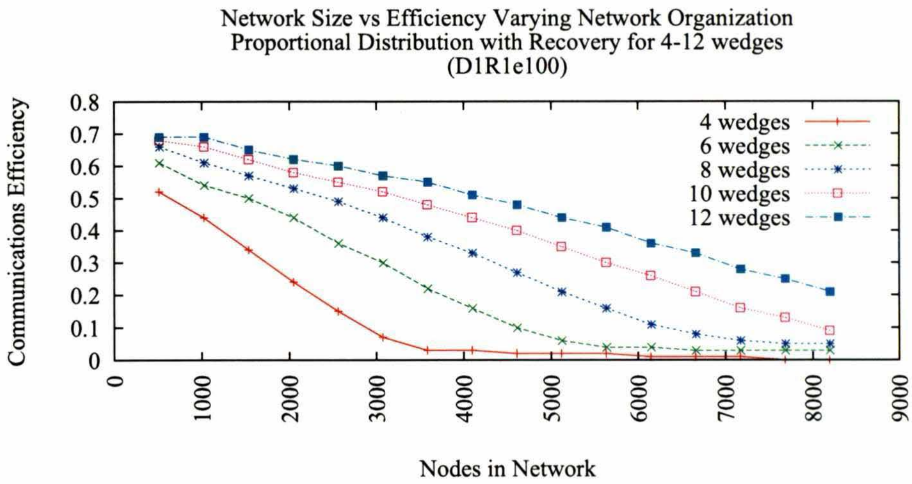


FIG. 31: Network communications efficiency for proportional node distribution in dual-plane configuration.

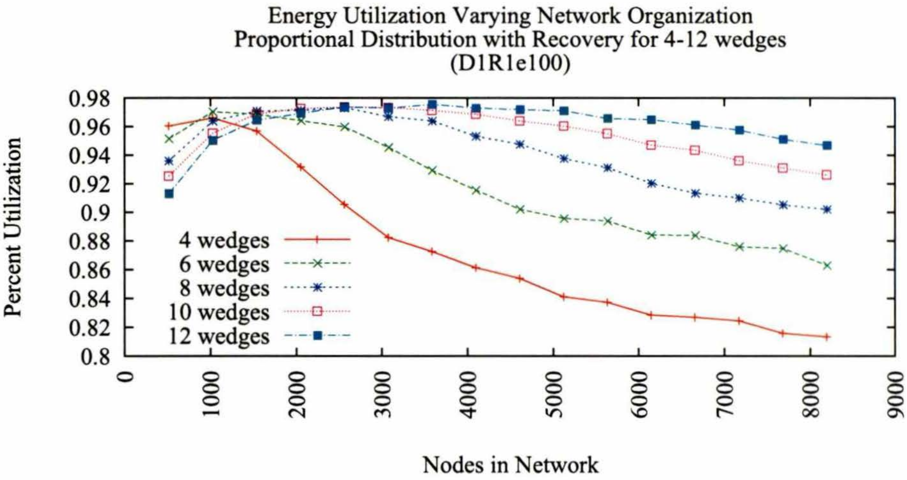


FIG. 32: Network energy utilization for proportional node distribution in dual-plane configuration.

IV.2.4 Analysis of the Proportional Distribution

TABLE 9: Average percent increase associated with recovery protocol in a proportionally distributed network.

Wedges	Cycles	Utilization	Efficiency
4	-9.9%	2.2%	11.2%
6	-3.9%	2.0%	7.6%
8	-0.2%	2.0%	4.0%
10	1.7%	1.8%	0.7%
12	3.1%	1.8%	-0.2%

When compared to the performance of the single-plane configuration for proportional distribution, the results are not as straight forward. Table 9 lists the average improvements for the three metrics of longevity, utilization, and efficiency. From this, it appears that the trend is for the recovery mechanism to improve the overall network performance when used in conjunction with higher wedge counts. The table appears to indicate that as the number of wedges increases, the benefit of the recovery mechanism is realized in a longer operational life of the network with a significant improvement in efficiency. This data is misleading as the previous sections have indicated that at least the first two wedge configurations are not useful for large networks and skew the trend in the data. The complete data sets are plotted in Figures 33, 34, and 35 for cycle life, communications efficiency, and energy utilization, respectively. These illustrate that for any of the node sizes, the recovery mechanism marginally improves the longevity of the network and that increasing the wedge count provides the optimal performance. Table 9 indicates that there is an overall improvement in network performance resulting from the recovery protocol for the 10 and 12-wedge configurations. The data also reflects the observed issues with large networks and low wedge counts. From the plots, it can be seen that there is a noticeable and consistent improvement for smaller networks in general. This indicates that as the network enlarges, the wedge count should increase to maintain a viable network configuration. Also, given a sufficient ratio of wedge count to network size, the recovery mechanism provides a benefit as the network begins to fail, resulting in an increased life of the network as well as an improvement in the aggregate network efficiency.

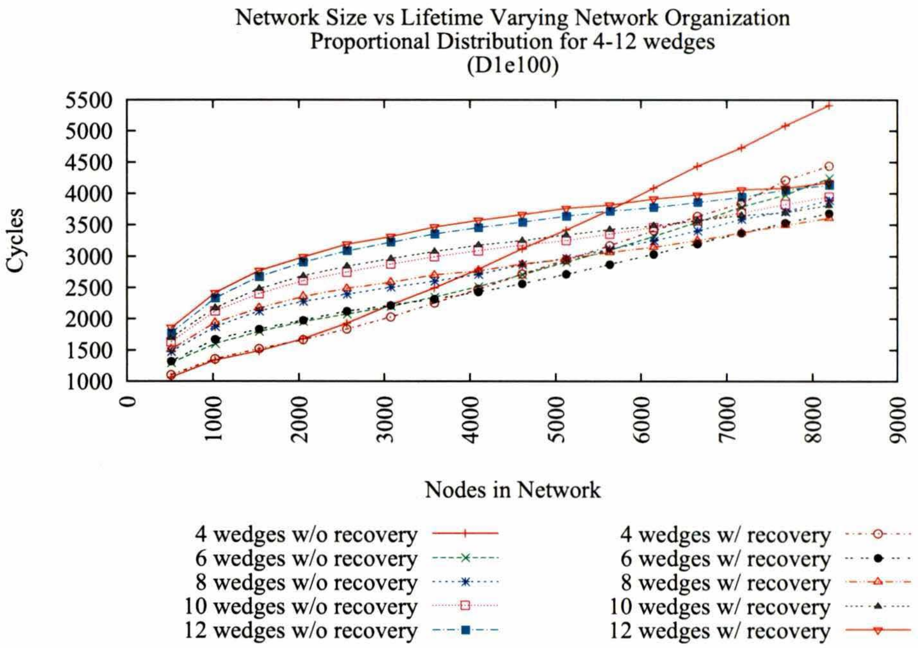


FIG. 33: Network life for proportional node distribution in single-plane and dual-plane configurations.

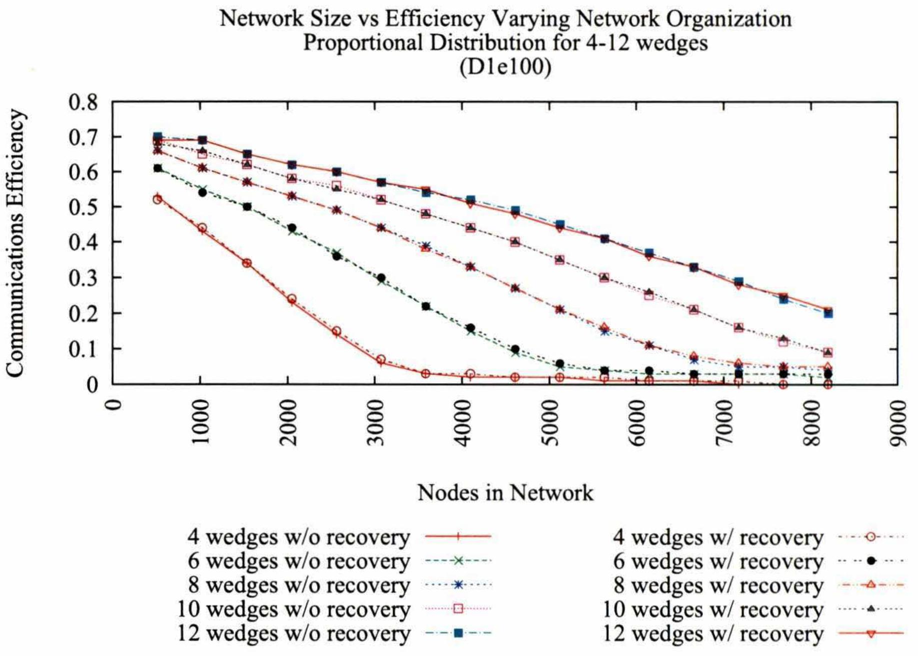


FIG. 34: Network communications efficiency for proportional node distribution in single-plane and dual-plane configurations.

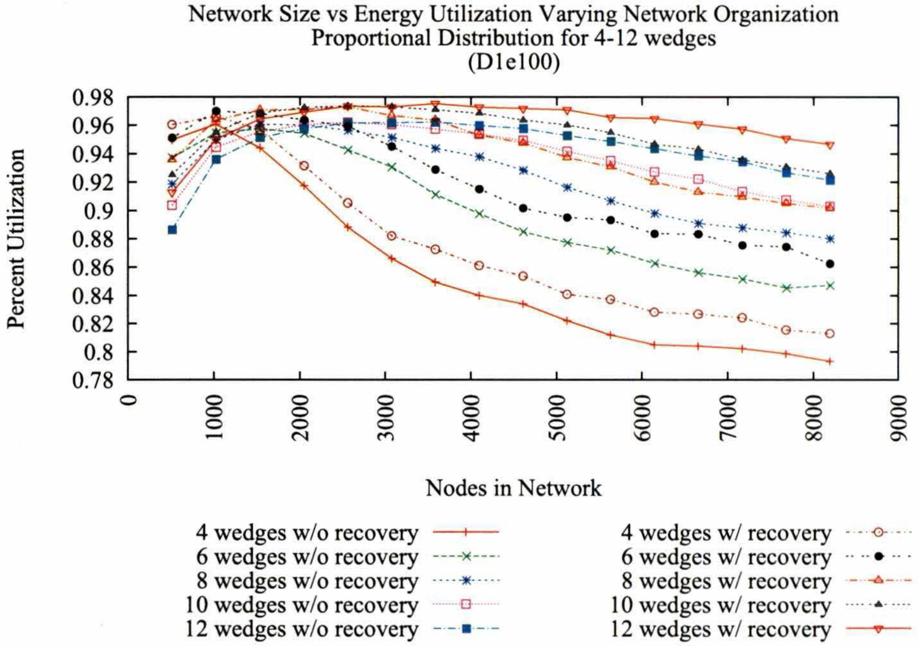


FIG. 35: Network energy utilization for proportional node distribution in single-plane and dual-plane configurations.

CHAPTER V

SUMMARY AND FUTURE WORK

V.1 Summary

This research investigated and evaluated a proposed fault-recovery mechanism in the context of a self-training wireless sensor network [3]. The proposed mechanism augments the training protocol to instill fault-tolerance and enhance operational efficiency. The context and validity of the problem space is established through a comprehensive review of relevant research and applications of fault-tolerance in WSN. This is followed by a brief discourse on the terminology of fault-tolerance in the context of WSN to establish the language of the problem space.

The theory for the recovery mechanism has been presented with the hypothesis that it extends the operational life of the network, in turn, increasing the effective network efficiency. The results have been presented for the performance of the recovery mechanism in both uniformly distributed and proportionally distributed networks. The average communications efficiency has not been observed to be significantly affected; however, the general trend indicates that in most configurations, recovery does realize some modicum of fault-recovery, extending the useful life, if only marginally. This yields the benefit of an improved overall network efficiency.

In summary, a fault-tolerant recovery protocol has been presented. Appropriate terminology in the context of fault-tolerance have been defined and applied to an observed fault in the reference network model. Through simulation, the proposed recovery mechanism has been evaluated and demonstrated to improve the aggregate efficiency of the network during the transition from operational to a failed state.

V.2 Contribution

This work has made several unique contributions to the initial network model developed and documented by Wadaa et al. [3] describing an algorithm for network training. Study of this network training and topology is continued by Elmiligui in his thesis [2] which develops both an analytical model as well as a simulation to determine the probability of failure in a given wedge. This work expands on the network model to incorporate and evaluate the application of fault-tolerance to mitigate an observed fault resultant from the prescribed training. This work makes five unique contributions to the study of FT WSNs and the maturation of the aforementioned network topology. The relevant terminology for fault-tolerance with respect to WSN is defined. A recovery algorithm is developed to mitigate an observed fault in the network model. The original training protocol [3] is amended to account for the proposed recovery mechanism. A characterization of energy utilization and aggregate network efficiency is presented for the proposed augmentation. Finally, the simulation first developed by Elmiligui [2] is significantly extended to enable the evaluation of the proposed recovery mechanism. The contributions to the simulation implement modifications to:

- the original training algorithm [3]
- the cluster head election process
- data structures to account for recovery mechanism
- instrumentation to generate performance statistics

V.3 Future work

During the development and execution of this thesis, several opportunities have been observed for derivative and future research. The first topic is related to managing the exhaustion of clusters in the first corona. The second investigates techniques in sub-clustering to manage large clusters. A third investigates benefits of task queuing in the cluster heads to avoid the losses associated with collisions. The fourth topic proposes a QoS+n protocol to mitigate the failed data node scenario.

The first topic for potential future research attempts to address the dilemma of exhaustion of the first corona. The premise is to augment the network with multiple

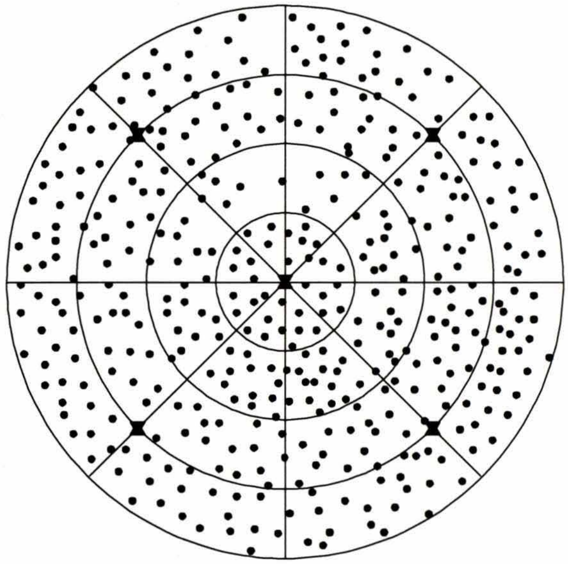


FIG. 36: Network with multiple sink nodes.

nodes that could support the function of the sink node. These nodes would either have power in great excess of standard data nodes and would alternate performing the sink function. As an example, given the network topology shown in Figure 1 in §III, there could be four alternate sinks in the upper left, upper right, lower left, and lower right as shown in Figure 36. The distributed sinks would temporarily manage network traffic, migrating that responsibility either upon the detection of some performance threshold or at scheduled intervals. In this way, the focus of communications is made transient, regulating the effects of traffic localization around a fixed point in the network. Similar work has been presented in [11] which utilizes the concept of distributed sinks via "proxy nodes" that collect data from the network and deliver it to the sink. The concept of mobile sinks is studied in [39] which also attempts to address the depletion of energy around the sink.

The second topic for future research investigates sub-clustering in large networks. As seen in the network model, outer clusters can be extremely large which unduly burdens the cluster head. Also, energy is expended by nodes actively attempting to participate in broadcasted requests even though many will not be acknowledged to support data collection. The cluster could be partitioned into sub-clusters configured *a priori*, by QoS requirements, or by performance tracking where sub-clusters selectively respond to requests. The selection process could be round-robin by iterating through the sub-clusters. The transition of a cluster from quiescent to active could be managed through the cluster head election process, or simply take turns responding to task requests. Another approach could reserve cluster nodes and have them not participate until active nodes start to fail. The quiescent nodes would conserve energy until needed, extending the useful life of the cluster and minimizing the load on the cluster head.

The third topic addresses a simplification in the network model employed in this work. The clusters and network performance suffered from task collisions which could be mitigated given modifications to the cluster nodes. The cluster head could employ some mechanism for queuing task requests such that multiple tasks could be concurrently active in a cluster. Consideration would have to be given to flushing the queue of stale or blocked requests as well as transferring the queue to subsequent cluster heads. This would be expected to significantly improve the efficiency of network communications .

The final topic for future work evaluates an extended QoS scheme. In the current network model, a failed data node can result in a cluster being blocked and the task failing, degrading communications and network performance. The cluster head registers only enough *interested* data nodes to satisfy the specified QoS for the current task. The cluster head could register a additional nodes, n of them, to reduce the probability that a failed node would prevent a task from completing.

BIBLIOGRAPHY

[1] P. Sridhar, A. M. Madni, and M. Jamshidi, "Hierarchical aggregation and intelligent monitoring and control in fault-tolerant wireless sensor networks," *Systems Journal, IEEE*, vol. 1, pp. 38–54, Sept. 2007.

[2] A. A. Elmiligui, "Fault modeling in wireless sensor networks," Master's Thesis, Old Dominion University, Dec. 2006.

[3] A. Wadaa, S. Olariu, L. Wilson, M. Eltoweissy, and K. Jones, "Training a wireless sensor network," *MONET*, vol. 10, no. 1-2, pp. 151–168, 2005.

[4] A. Avizienis, J.-C. Laprie, B. Randell, and C. E. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans. Dependable Sec. Comput*, vol. 1, no. 1, pp. 11–33, 2004.

[5] J. Park and S. Sahni, "An online heuristic for maximum lifetime routing in wireless sensor networks," *Computers, IEEE Transactions on*, vol. 55, pp. 1048–1056, Aug. 2006.

[6] A. Datta, "A fault-tolerant protocol for energy-efficient permutation routing in wireless networks," *Computers, IEEE Transactions on*, vol. 54, pp. 1409–1421, Nov. 2005.

[7] M. Zimmerling, "An energy-efficient routing protocol for linear wireless sensor networks," in *Informatiktage*, vol. S-6 of *LNI*, pp. 239–242, GI, 2008.

[8] Q. Xu and Y. Wang, "Solving reliable coverage in fault tolerant energy efficient wireless sensor network," in *Pervasive Computing and Applications, 2006 1st International Symposium on*, pp. 791–796, Aug. 3-5, 2006.

[9] D. Kandris, P. Tsioumas, A. Tzes, N. Pantazis, and D. Vergados, "Hierarchical energy efficient routing in wireless sensor networks," in *Control and Automation, 2008 16th Mediterranean Conference on*, pp. 1856–1861, June 2008.

[10] H. Yu, L. Wei, and K. Zhenhua, "Study on energy efficient hierarchical routing protocols of wireless sensor network," in *Information Engineering, 2009. ICIE '09. WASE International Conference on*, vol. 1, pp. 325–328, July 2009.

[11] P. Djukic and S. Valaee, "Minimum energy fault tolerant sensor networks," in *Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004. IEEE*, pp. 22–26, Nov. 29 - Dec. 1, 2004.

[12] Q. Liang, L. Wang, and Q. Ren, "Fault-tolerant and energy efficient cross-layer design for wireless sensor networks," *Int. J. of Sensor Networks*, vol. 2, pp. 248–257, Apr. 11, 2007.

[13] G. Khanna, S. Bagchi, and Y.-S. Wu, "Fault tolerant energy aware data dissemination protocol in sensor networks," in *Dependable Systems and Networks, 2004 International Conference on*, pp. 795–804, June 28 - July 1, 2004.

[14] M. Larrea, C. Martin, and J. Astrain, "Hierarchical and fault-tolerant data aggregation in wireless sensor networks," in *Wireless Pervasive Computing, 2007. ISWPC '07. 2nd International Symposium on*, pp. 531–536, Feb. 5-7, 2007.

[15] D. Bein, "Fault-tolerant k-fold pivot routing in wireless sensor networks," in *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, pp. 494–500, Jan. 7-10, 2008.

[16] R. Farivar, M. Fazeli, and S. Miremadi, "Directed flooding: A fault-tolerant routing protocol for wireless sensor networks," in *Systems Communications, 2005. Proceedings*, pp. 395–399, Aug. 2005.

[17] Z. Xiong, Z. Yang, W. Liu, and Z. Feng, "A Lightweight FEC Algorithm for Fault Tolerant Routing in Wireless Sensor Networks," in *Wireless Communications, Networking and Mobile Computing, 2006. WiCOM 2006.International Conference on*, pp. 1–4, Sept. 2006.

[18] K. Sha, J. Du, and W. Shi, "WEAR: a balanced, fault-tolerant, energy-aware routing protocol in WSNs," *Int. J. of Sensor Networks*, vol. 1, pp. 156–168, Jan. 12, 2007.

[19] M. Gregoire and I. Koren, "An adaptive algorithm for fault tolerant re-routing in wireless sensor networks," in *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on*, pp. 542–547, Mar. 19-23, 2007.

[20] L. Wang, H. Jin, J. Dang, and Y. Jin, "A fault tolerant topology control algorithm for large-scale sensor networks," in *Parallel and Distributed Computing, Applications and Technologies, 2007. PDCAT '07. Eighth International Conference on*, pp. 407–412, Dec. 3-6, 2007.

[21] C.-W. Chen, K.-F. Ssu, and H. Jiau, "Fault-tolerant topology control with adjustable transmission ranges in wireless sensor networks," in *Dependable Computing, 2007. PRDC 2007. 13th Pacific Rim International Symposium on*, pp. 131–138, Dec. 17-19, 2007.

[22] I. Saha, L. Sambasivan, S. Ghosh, and R. Patro, "Distributed fault tolerant topology control in wireless ad-hoc sensor networks," in *Wireless and Optical Communications Networks, 2006 IFIP International Conference on*, Feb. 2007.

[23] M. Cardei, S. Yang, and J. Wu, "Fault-tolerant topology control for heterogeneous wireless sensor networks," in *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*, pp. 1–9, Oct. 2007.

[24] M. Cardei, S. Yang, and J. Wu, "Algorithms for fault-tolerant topology in heterogeneous wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst*, vol. 19, no. 4, pp. 545–558, 2008.

[25] S. Ali, A. Fakoorian, and H. Taheri, "Optimum Reed-Solomon Erasure Coding in Fault Tolerant Sensor Networks," in *Wireless Communication Systems, 2007. ISWCS 2007. 4th International Symposium on*, pp. 6–10, Oct. 16-19, 2007.

[26] T.-Y. Wang, Y. S. Han, and P. Varshney, "A combined decision fusion and channel coding scheme for fault-tolerant classification in wireless sensor networks," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, vol. 2, pp. 1073–1076, May 2004.

[27] T.-Y. Wang, Y. S. Han, B. Chen, and P. K. Varshney, "A combined decision fusion and channel coding scheme for distributed fault-tolerant classification in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 7, pp. 1695–1705, 2006.

[28] A. Boukerche and A. Martirosyan, "An energy-aware and fault tolerant intercluster communication based protocol for wireless sensor networks," in *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, pp. 1164–1168, Nov. 2007.

[29] S. Haykin and M. Moher, *Modern Wireless Communication.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2004.

[30] Y. Chen and S. Son, "A fault tolerant topology control in wireless sensor networks," in *Computer Systems and Applications, 2005. The 3rd ACS/IEEE International Conference on*, pp. 57–64, Jan. 6, 2005.

[31] S. Gobriel, S. Khattab, D. Mosse, J. Brustoloni, and R. Melhem, "Ridesharing: Fault tolerant aggregation in sensor networks using corrective actions," in *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, vol. 2, pp. 595–604, Sept. 2006.

[32] Y. Wang and H. Wu, "Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN): A New Paradigm for Pervasive Information Gathering," *Mobile Computing, IEEE Transactions on*, vol. 6, pp. 1021–1034, Sept. 2007.

[33] S. Lee, U. Jang, and J. Park, "Fast fault-tolerant time synchronization for wireless sensor networks," in *Object Oriented Real-Time Distributed Computing*

(ISORC), 2008 11th IEEE International Symposium on, pp. 178–185, May 5-7, 2008.

[34] W. L. Lee, A. Datta, and R. Cardell-Oliver, "FlexiTP: A flexible-schedule-based TDMA protocol for fault-tolerant and energy-efficient wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst*, vol. 19, no. 6, pp. 851–864, 2008.

[35] S. Seareesavetrat, C. Pornavalai, and R. Varakulsiripunth, "A light-weight fault-tolerant time synchronization for wireless sensor networks," in *ITS Telecommunications, 2008. ITST 2008. 8th International Conference on*, pp. 182–186, Oct. 2008.

[36] P. K. Lala, *Self-checking and fault-tolerant digital design.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001.

[37] *Webster's College Dictionary.* Barnes & Noble Books, 2003.

[38] R. Butler, "A Primer on Architectural Level Fault Tolerance," Tech. Rep. TM-2008-215108, NASA Langley Research Center, Hampton, VA, Feb. 2008.

[39] M. Marta and M. Cardei, "Using sink mobility to increase wireless sensor networks lifetime," in *World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a*, pp. 1–10, June 23-26, 2008.

APPENDIX A

ADDITIONAL DATA SETS

A.1 Plots for Initial Energy of 200

The following plots depict the results for simulations with an initial energy of 200 units.

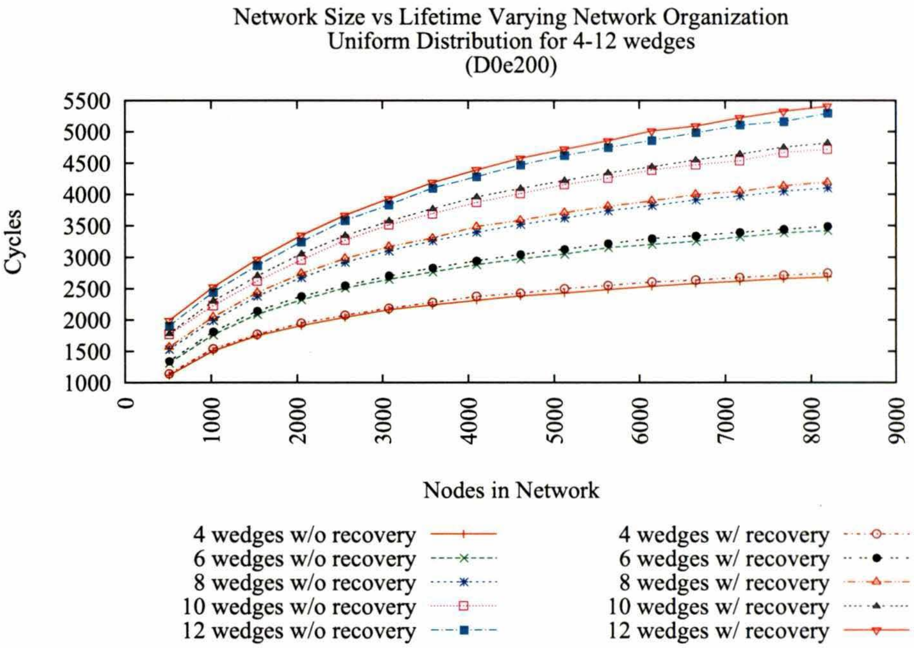


FIG. 37: Network life for uniform node distributed networks with an initial energy of 200 units.

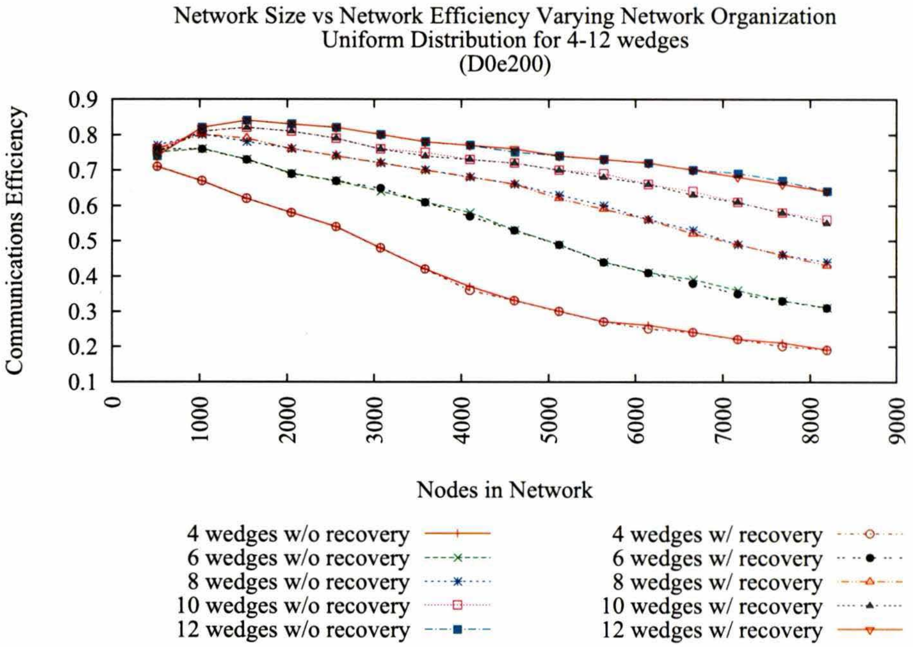


FIG. 38: Network communications efficiency for uniform node distributed networks with an initial energy of 200 units.

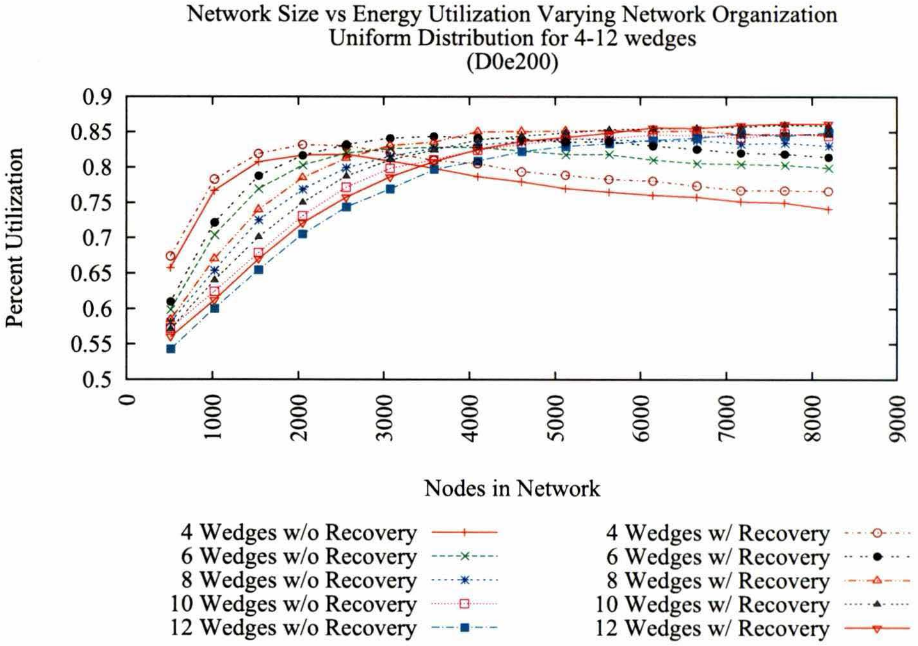


FIG. 39: Network energy utilization for uniform node distributed networks with an initial energy of 200 units.

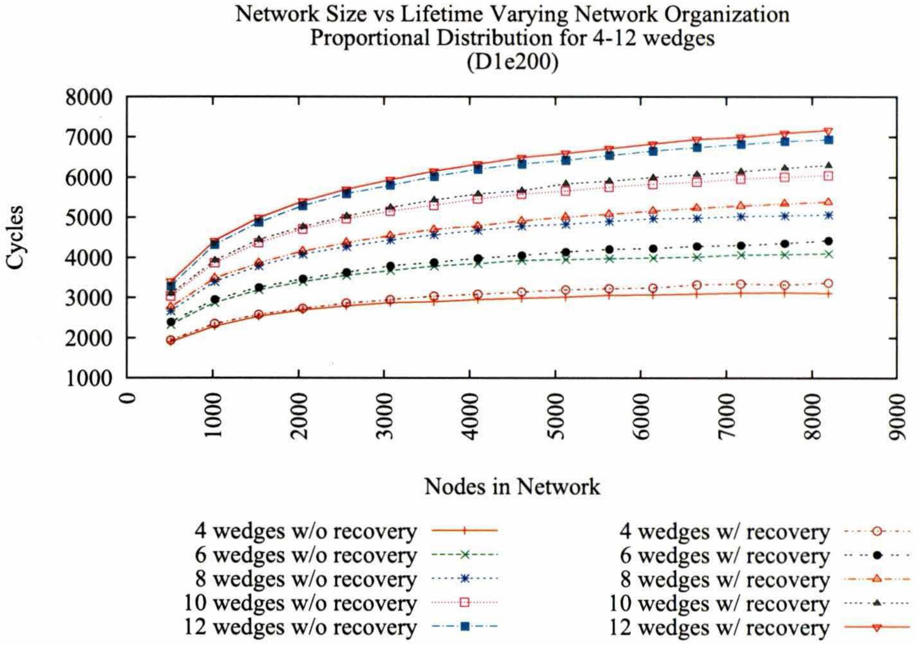


FIG. 40: Network life for proportional node distributed networks with an initial energy of 200 units.

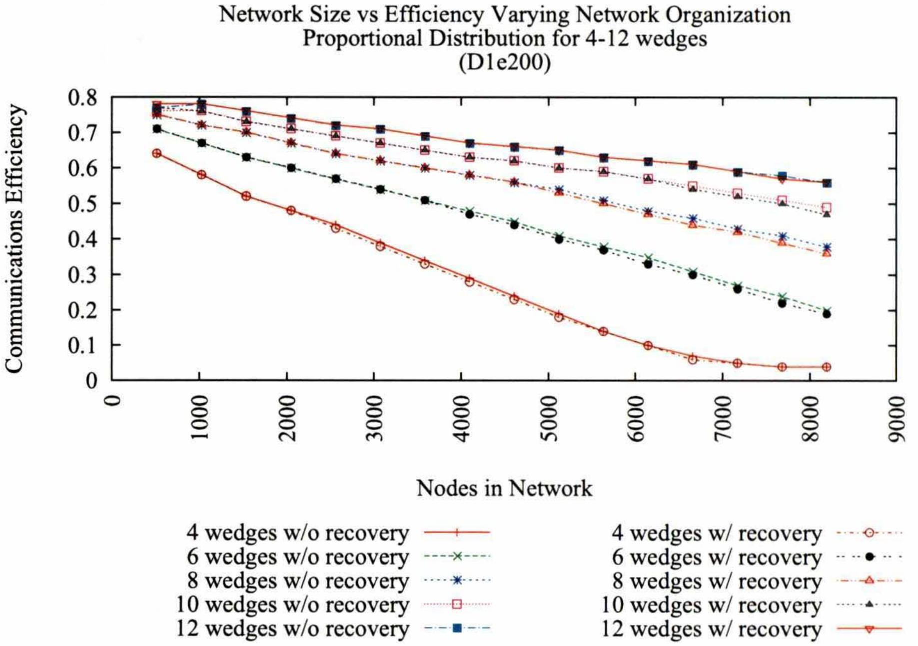


FIG. 41: Network communications efficiency for proportional node distributed networks with an initial energy of 200 units.

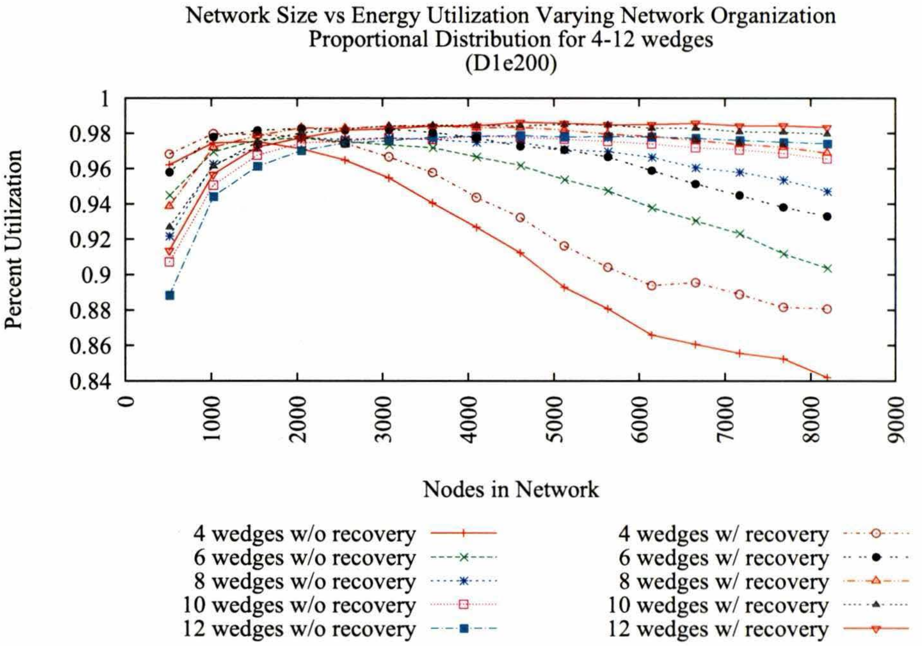


FIG. 42: Network energy utilization for proportional node distributed networks with an initial energy of 200 units.

A.2 Plots for Initial Energy of 400

The following plots depict the results for simulations with an initial energy of 400 units.

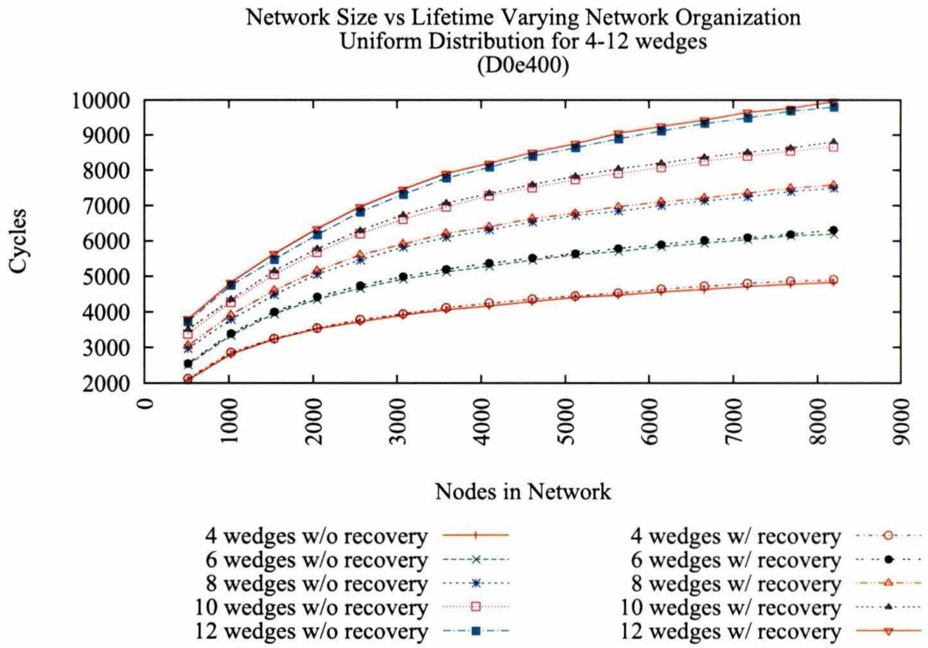


FIG. 43: Network life for uniform node distributed networks with an initial energy of 400 units.

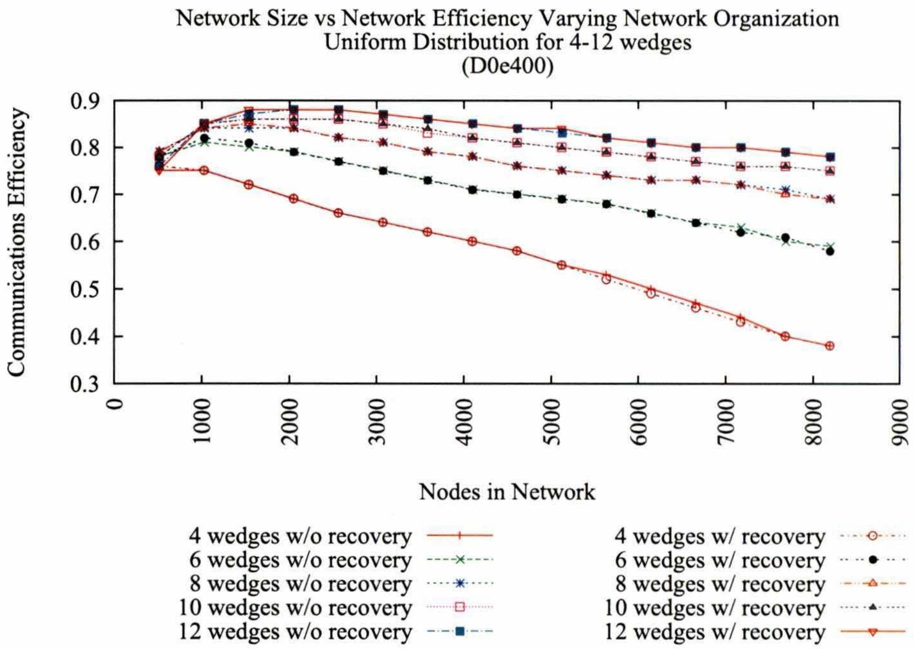


FIG. 44: Network communications efficiency for uniform node distributed networks with an initial energy of 400 units.

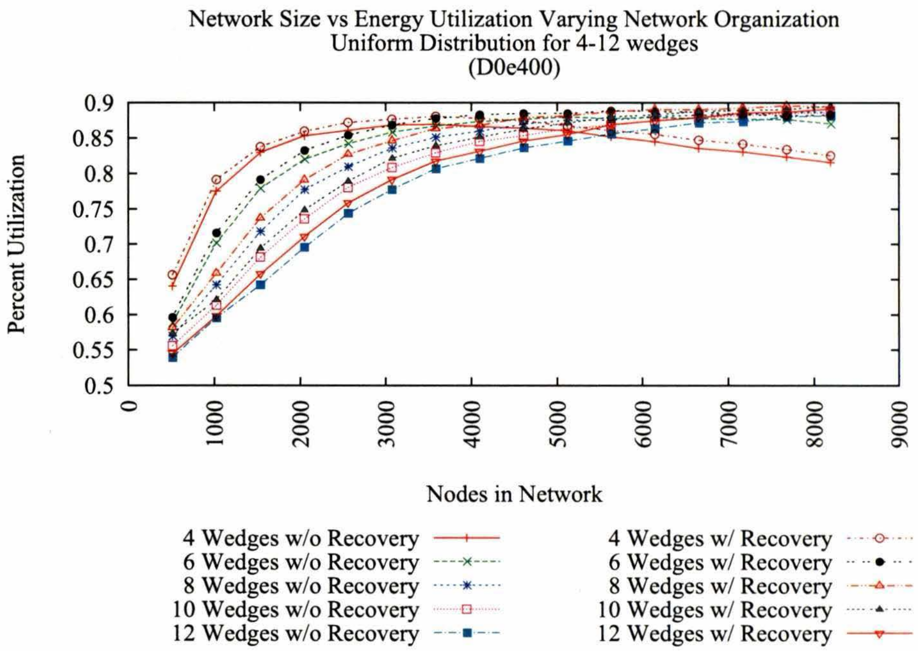


FIG. 45: Network energy utilization for uniform node distributed networks with an initial energy of 400 units.

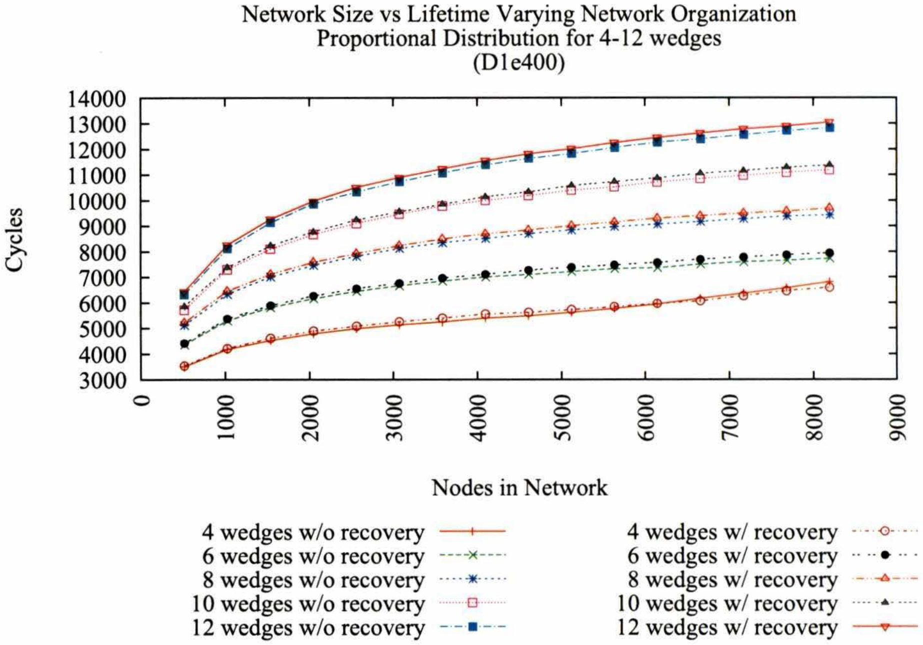


FIG. 46: Network life for proportional node distributed networks with an initial energy of 400 units.

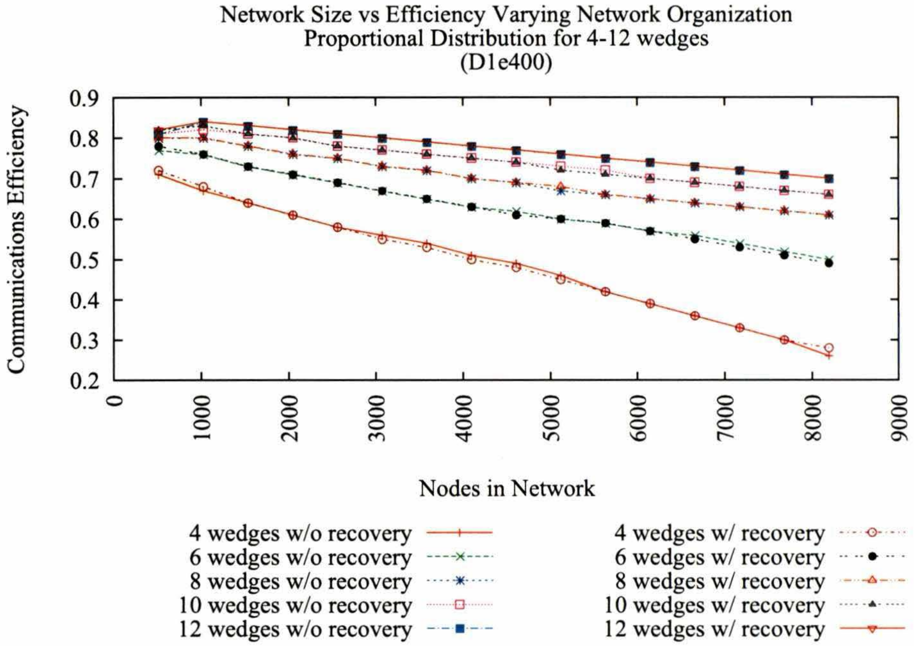


FIG. 47: Network communications efficiency for proportional node distributed networks with an initial energy of 400 units.

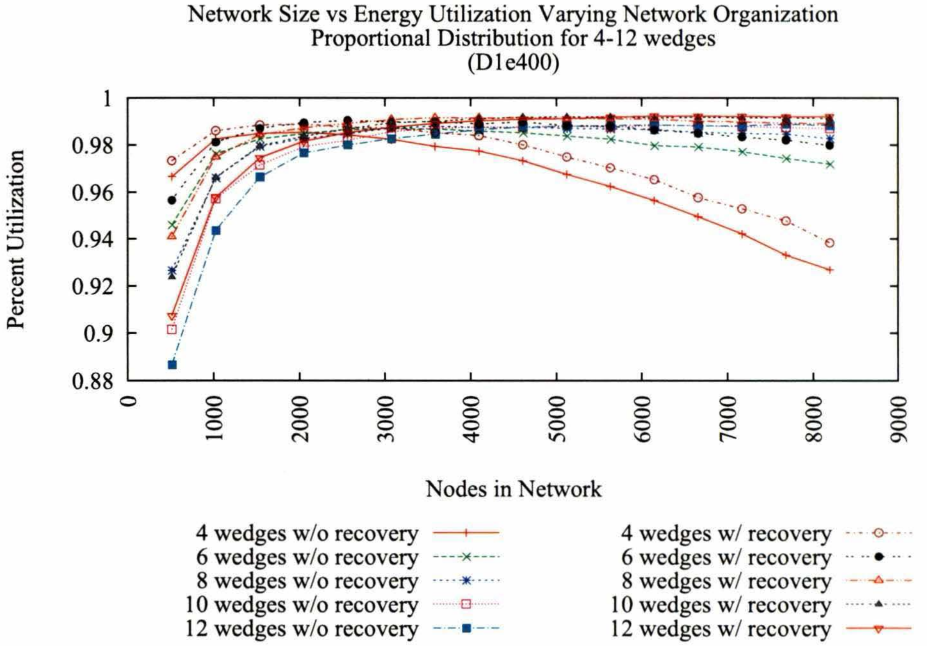


FIG. 48: Network energy utilization for proportional node distributed networks with an initial energy of 400 units.

VITA

Kevin M. Somervill
Department of Electrical and Computer Engineering
Old Dominion University
Norfolk, VA 23529

Education

Master of Science, Computer Engineering, Old Dominion University, Norfolk, VA, December, 2009.
Bachelor of Science, Computer Engineering, Christopher Newport University, Newport News, VA, May, 2000.

Publications

Sana Rezgui, Gary Swift, Kevin Somervill, Jeff George, Carl Carmichael, and George Allen, "Complex Upset Mitigation Applied to a Re-Configurable Embedded Processor", *IEEE Transactions on Nuclear Science*, vol. 52, issue 6, part 1, Dec. 2005, pp. 2468–2474.

Presentations

Kevin Somervill, "Lunar Applications in Reconfigurable Computing", MAPLD 2008, Annapolis, Maryland, Sept. 15-18, 2008.
Kevin Somervill, Robert F. Hodson, John Williams, and Robert Jones, "Reconfigurable Processing Module", MAPLD 2005, Washington DC, Sept. 7-9, 2005.

Professional Development

NASA Langley Research Center, Hampton, VA, Electronics Engineer, A.S.T, January 2003.
Hewlett Packard Co., Roseville, CA, Digital Design Engineer, June 2000.
Thomas Jefferson National Laboratory, Newport News, VA, Research Assistant, December 1998.

Typeset using LaTeX.