Performability in Wireless Sensors Network

By

Fadi Eid

A thesis submitted
in partial fulfillment of the requirements for the degree of
Master of Science
Computer Information System

Faculty of Natural and Applied Sciences
Department of Computer Science
Notre Dame University – Louaize
Zouk Mosbeh, Lebanon

Spring 2005

Performability in Wireless Sensors Network

By Fadi Eid

Committee Members:

H Maalouf

Dr. Hoda Maalouf: Assistant Professor of Computer Science and Chairperson.

Advisor.

Marie Khair

Dr. Marie Khair: Associate Professor of Computer Science.

Member of Committee.

Hikmat Farhat

Dr. Hikmat Farhat: Assistant Professor of Computer Science.

Member of Committee.

Jajou

Dr. Amer Jajou: Assistant Professor of Mathematics.

Member of Committee.

Date of thesis defense: **June 22, 2005**

Acknowledgments

Many people contributed to the success of this thesis, but I will permit myself to mention only a few

First and foremost, I would like to express my sincere thanks to Dr. Hoda Maalouf for kindly accepting to be my supervisor. Her academic and technical guidance and support were most valuable assets throughout the making of this research.
Many Thanks go to all the faculty members of our department.
I want to express my deepest gratitude to my parents for their continuous support. My special thanks go to my wife Lamis, whose love, patience and encouragement remain an essential part of this work.

Performability in Wireless Sensors Network

ABSTRACT

This thesis tackles Performability issues in Wireless sensors networks. Performability is a mission-specific measure of system effectiveness that seeks to combine the traditional reliability and performance measures of a system. Wireless Sensor networks consist of a huge number of small sensor nodes, which communicate wirelessly. These sensor nodes can be spread out in hard accessible areas by what new applications fields can be pointed out [13].

This thesis aims to provide wireless communication architecture for Petroleum installations, such as off shore platforms and on shore processing units, which are characterized by high density of temperature and pressure sensors [24]. There are many challenges related to this work, such as time synchronization, radio resource management and routing.

In this thesis we design a routing protocol for wireless sensors networks, namely Smart Routing in Wireless Sensors Networks (SR-WSN). We utilize NS2 networks simulation tool to evaluate the performability of the SR-WSN. In particular, these simulations study the impact of nodes failure on the total delay and power consumption.

Finally, we build up a Markovian Reward Model (MRM) to compare the performability between SR-WSN and a typical broadcast system. The found results prove that our system outperforms broadcast system.

Table of Contents

List of Acronyms

ADV	Advertisement for Data
AP	Access Point
BSS	Basic Service Set
CONSER	Collaborative Simulation for Education and Research
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DARPA	Defense Advanced Research Projects Agency
DSN	Distributed Sensor Networks
ESS	Extended Service Set
FIFO	First in First out
ICSI	Center for Internet Research
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
LAN	Local Area Networks
LEACH	Low-Energy Adaptive Clustering Hierarchy
MANET	Mobile Adhoc Networks
MEMS	Micro Electromechanical System
MRM	Markov Reward Model
NIC	Network Interface Card
NS2	Network Simulator 2
P2P	Peer-to-Peer
PAN	Personal Area Networks
PDA	Personal Digital Assistants
QoS	Quality of Service
REQ	Request for Data
RSM	Route Switch Module
RTP	Real-time Transport Protocol
SAMAN	Simulation Augmented by measurement and Analysis for Networks
SensIT	Sensor Information Technology
SPIN	Sensor Protocols for Information via Negotiation
TASS	Tactical Automated Security System
TCP	Transmission Control Protocol
UDP	User Datagram
WSN	Wireless Sensor Networks

List of Tables and Figures

Chapter 1

Introduction and problem definition

1.1 INTRODUCTION

Networked micro sensors are the most important technologies for the 21st century. Cheap, smart devices with multiple onboard sensors networked through wireless links and deployed in large numbers, and in several environment. Networked micro sensors provide the technology for a broad spectrum of systems, generating new capabilities for reconnaissance and surveillance as well as other tactical applications.

Networked micro sensors belong to the general family of sensor networks that use multiple distributed sensors to collect information on entities of interest. Table 1 summarizes the range of possible attributes in general sensor networks.

Current and potential applications of sensor networks include: military sensing, physical security, air traffic control, traffic surveillance, video surveillance, industrial and manufacturing automation, distributed robotics, environment monitoring, and building and structures monitoring.

The sensors in these applications may be small or large, and the networks may be wired or wireless. However, ubiquitous wireless networks of micro sensors probably offer the most potential in changing the world of sensing [3].

Table 1: Attributes of Sensor Networks

Sensors:	***Size:*** small (e.g. micro-electro mechanical systems (MEMS)), large (e.g., radars, satellites) ***Type:*** passive (e.g., acoustic, video, magnetic), active (e.g., radar) Composition or mix: homogeneous, heterogeneous Spatial coverage: dense, sparse Deployment: fixed and planned (e.g., factory, networks), ad hoc (e.g. air dropped) ***Dynamics:*** stationary (e.g., seismic sensors), mobile (e.g., robot)
Sensing entities of interest:	***Extent:*** distributed (e.g., environmental monitoring). Localized (e.g. target tracking) ***Mobility:*** Static, Dynamic ***Nature:*** Cooperative (e.g., air traffic control), non-cooperative (e.g. military targets)
Operating environment:	Factory floor, battlefield
Communication:	***Networking:*** Wired, Wireless Bandwidth: high, low
Processing architecture:	Centralized (all data sent to central site), distributed (located at sensor or other sites), hybrid
Energy Availability:	Constrained (e.g. in small sensors), unconstrained (e.g., in large sensors)

1.2 SENSOR NETWORKS EVOLUTION

The development of sensor networks requires technologies from three different research areas: sensing, communication, and computing (including hardware, software, and algorithms). Combined and separate advancements in each of these areas have stimulated research in sensor networks. Examples of early sensor networks include the radar networks used in air traffic control. The national power grid, with its many sensors, can be viewed as one large sensor network. These systems were developed with specialized computers and communication capabilities, and before the term “sensor networks” came into vogue.

1.2.1. Early Research on Military Sensor Networks

Modern research on sensor networks started around 1980 with the Distributed Sensor Networks (DSN) program at the Defense Advanced Research Projects Agency (DARPA).

For demonstration, MIT Lincoln Laboratory developed the real-time test bed for acoustic tracking of low-flying aircraft [19]. The sensors were acoustic arrays (nine micro-phones arranged in three concentric triangles with the largest being 6 m across). A PDP11/34 computer and an array processor processed the acoustic signals. The nodal computer (for target tracking) consists of three MC68000 processors with 256-kB memory and 512-kB shared memory, and a custom operating system. Communication was by Ethernet and microwave radio.

1.2.2. Sensor Network Research in the 21st Century

Recent advances in computing and communication have caused a significant shift in sensor network research and brought it closer to achieving the original vision. Small and inexpensive sensors based upon micro electromechanical system (MEMS) [22] technology, wireless networking, and inexpensive low-power processors allow the deployment of wireless ad hoc networks for various applications. The Defense Advanced Research Projects Agency (DARPA) started a research program on sensor networks to leverage the latest technological advances.

The recently concluded DARPA Sensor Information Technology (SensIT) program [4] pursued two key research and development thrusts. First, it developed new networking techniques. In the battlefield context, these sensor devices or nodes should be ready for rapid deployment, in an ad hoc fashion, and in highly dynamic environments. Today's networking techniques, developed for voice and data and relying on a fixed infrastructure, will not suffice for battlefield use. Thus, the program developed new networking techniques suitable for highly dynamic ad hoc environments. The second thrust was networked information processing, i.e., how to extract useful, reliable, and timely information from the deployed sensor network. This implies leveraging the distributed computing environment created by these sensors for signal and information processing in the network, and for dynamic and interactive querying and tasking the sensor network.

SensIT generated new capabilities relative to today's sensors. Current systems such as the Tactical Automated Security System (TASS) [14] for perimeter security are dedicated rather than programmable. They use technologies based on transmit-only nodes and a long-range detection paradigm.

SensIT networks have new capabilities. The networks are interactive and programmable with dynamic tasking and querying. A multitasking feature in the system allows multiple simultaneous users. Finally, since detection ranges are much shorter in a sensor system, the software and algorithms can exploit the proximity of devices to threats in order to drastically improve the accuracy of detection and tracking. The software and the overall system design supports low latency, energy-efficient operation, built-in autonomy and survivability, and low probability of detection of operation. As a result, a network of SensIT nodes can support detection, identification, and tracking of threats, as well as targeting and communication, both within the network and to outside the network, such as an overhead asset.

1.3. TECHNOLOGY TRENDS

Current sensor networks can exploit technologies not available 20 years ago and perform functions that were not even dreamed of at that time. Sensors, processors, and communication devices are all getting much smaller and cheaper. Commercial companies such as Ember, Crossbow, and Sensoria are now building and deploying small sensor nodes and systems. These companies provide a vision of how our daily lives will be enhanced through a network of small, embedded sensor nodes. In addition to products from these companies, commercial off-the-shelf personal digital assistants (PDAs) using Palm or Pocket PC operating systems contain significant computing power in a small package. These can easily be upgraded to become processing nodes in a sensor network. Some of these devices even have built-in sensing capabilities, such as cameras. These powerful processors can be hooked to MEMS devices and machines along with extensive databases and communication platforms to bring about a new era of technologically sophisticated sensor nets.

Wireless networks based upon IEEE 802.11 standards can now provide bandwidth approaching those of wired networks. At the same time, the IEEE has noticed the low expense and high capabilities that sensor networks offer. The organization has defined the IEEE 802.15 standard for personal area networks (PANs), with “personal networks” defined to have a radius of 5 to 10 m. Networks of short-range sensors are the ideal technology to be employed in PANs. The IEEE encouragement of the development of technologies and algorithms for such short ranges ensures continued development of low-cost sensor nets [11]. Further more, increases in chip capacity and processor production capabilities have reduced the energy per bit requirement for both computing and communication. Sensing, computing, and communications can now be performed on a single chip, further reducing the cost and allowing deployment in ever-larger numbers.

Our petroleum system will profit from the MEMS technology, which will produce sensors that are even more capable and versatile. MEMS sensors can sense and communicate and are tiny enough to fit inside a cubic millimeter. A Smart Dust optical mote uses MEMS to aim sub millimeter-sized mirrors for communications. Smart Dust sensors can be deployed using a 3 10 mm "wavelet" shaped like a maple tree seed and dropped to float to the ground. A wireless network of these ubiquitous, low-cost, disposable micro sensors can provide close-in Sensing capabilities in many novel applications.

Table 2. Comparison between different generations of sensor nodes.

	Yesterday(1980 –1990's	**Today(2000 - 2005)**	**Tommorrow(2010)**
Manufacturer	Custom contractor's, e.g, for TRSS	Commercial: Crossbow Technology, Inc. Sensoria Corp., Ember Corp	Dust, Inc. and others to be formed
Size	Large shoe box and up	Pack of cards to small shoe box	Dust particle
Weight	Kilograms	Grams	Negligible
Node architecture	Separate sensing, processing and Communication	Integrated sensing, processing and communication	Integrated sensing, processing and communication
Topology	Point-to-point, star	Client server, peer to peer	Peer to peer
Power supply lifetime	Large batteries; hours, days and longer	AA batteries; days to weeks	Solar; months to year
Deployment	Vehicule-placed or air-drop single sensors	Hand-emplaced	Embedded, "sprinkled" left-behind

1.4 PROBLEM DEFINITION

Wireless sensors networks are a natural candidate for Petroleum Systems, since they enable the usage of thousand pressure and temperature sensors and save the high node density which entails substantial cabling cost, both in terms of weight, installation effort and maintenance.

Wireless Sensor networks for Petroleum Systems pose a number of unique technical challenges due to the following factors:

- **Sensors Density:** A very large numbers of sensors are used in petroleum systems;
- **Sensors Lifetime:** sensors are placed in potentially explosive areas, which put severe limits on Peak Power Consumption;
- **Unattended operation:** In most cases, once deployed, sensor networks have no human intervention. Hence the nodes themselves are responsible for reconfiguration in case of any changes;
- **Unfettered:** the device is assumed to be battery powered, giving stringent average power consumption requirements. The sensor nodes are not connected to any energy source. There is only a finite source of energy, which must be optimally used for processing and communication. An interesting fact is that communication dominates processing in energy consumption. In order to make optimal use of energy, communication should be minimized as much as possible;
- **Dynamic changes:** It is required that a sensor network system be adaptable to changing connectivity (for e.g., due to addition of more nodes, failure of nodes etc.). Unlike traditional networks, where the focus is on maximizing channel throughput or minimizing node

deployment, the major consideration in a sensor network is to extend the system lifetime as well as the system robustness [18].

The routing problem for sensor networks differs substantially from that of traditional ad-hoc wireless networks because sensor networks typically involve many resources constrained;

- **Response time**: requirement for this type of installation is the packet delivery performance as well as the routing protocols algorithm for this collection of sensors.

In light of the above, this thesis seeks to design a wireless sensors networks protocol that achieves fault tolerance in the presence of individual node failure while minimizing energy consumption.

Since the limited wireless channel bandwidth must be shared among all the sensors in our network, the proposed routing protocol for these types of networks should be able to perform local collaboration to reduce bandwidth requirements.

1.5 SYSTEM REQUIREMENT

This thesis assumes that the underlying layers are PHY/MAC defined by IEEE 802.11 in the 2.4 GHz frequency band. 802.11 specifications use the Ethernet protocol and Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) for path sharing [8]. (Details on CSMA/CA protocol are presented in chapter 2).

1.6 RESEARCH OBJECTIVES

This thesis uses wireless sensors networks to monitor ambient conditions such as temperature and pressure in oil drillers platforms and Petroleum systems. In this environment, Wireless sensor network technology poses its unique design challenges: The need for an effective routing protocol that consumes the least amount of power. This issue will be explained in more details next.

1.6.1 Routing protocols

Conventional wireless networks routing protocols have several limitations when being used in sensor networks due to the energy-constrained nature of these networks.

These protocols essentially follow the flooding technique in which a node stores the data item it receives and then sends copies of the data item to all its neighbors. There are two main insufficiencies to this approach [25].

- **Implosion**: If a node is a common neighbor to nodes holding the same data item, then it will get multiple copies of the same data item. Therefore, the protocol wastes resources sending the data item and receiving it;
- **Resource management**: In conventional flooding, nodes are not resource-aware. They continue with their activities regardless of the energy available to them at a given time.

Also, when it comes to unicast routing protocols, established information paths may fail at any given time in WSN. This may be due to the harsh, time-varying, wireless channel conditions, or in some situations due to nodes failure-crash that can cause the network topology itself to change rapidly. A greater concern in large-scale static sensor networks is the temporary or lasting failure of intermediate nodes.

These node failures can be due to the inherent unreliability of inexpensive components or due to battery drainage. Hence it is important for information routing algorithms in this space to provide tolerance to such failures in an energy-efficient manner

One of the main objectives of this research is to design a routing protocol for sensor networks that is able to overcome these deficiencies and to look at newer ways for routing protocol by conserving energy and increasing the lifetime of the network.

1.6.2 Energy Efficiency:

Energy consumption is the most important factor to determine the life of a sensor network. Usually sensor nodes are driven by battery and have very low energy resources. This makes energy optimization more complicated in sensor networks since it involved not only reduction of energy consumption but also prolonging the lifetime of the network as much as possible [21].

Developing design methodologies and architectures that help in energy aware design of sensor networks can reduce the power consumed by the sensor nodes. The lifetime of a sensor network can be increased significantly if the application layer and the network protocols are designed to be energy aware. Power management in radios is very important because radio communication consumes a lot of energy during operation of the system. Another aspect of sensor nodes is that a sensor node also acts as a router and the majority of the packets, which the sensors receive, are forwarded.

Traffic can also be distributed in such a way as to maximize the life of the network. A path should not be used continuously to forward packets regardless of how much energy is saved because this depletes the energy of the nodes on this path and there is a breach in the connectivity of the network. It is better that the load of the traffic be distributed more uniformly throughout the network.

A second objective of this thesis is to ensure that SR-WSN protocol does not waste a lot of energy and to distribute the network load among different possible paths.

1.7 APPROACH

The main aim of this thesis is the communication efficiency and fault-tolerance in a wireless sensors network. Since each sensor must act as a router with limited energy resource because the battery recharge or replacement is impractical, a network with a reliable routing protocol and energy-aware design is crucial to achieve the desired efficiency, effectiveness and lifetime performance. Good response time with high reliability and low energy consumption are essential to build a successful WSN for a petroleum system.

A fundamental assumption in a petroleum system is that all nodes are "equal", hence, any node can be used to forward packets between arbitrary sources and destinations. In this environment, we investigate a routing strategy with focus on solution that scale well to large number of sensors and can handle a percentage of nodes failure reaching the 50 percent.

Event driven simulations using NS2 are built in chapter 3 to analyze the delay and the power consumption in our proposed routing algorithm. Also Markovian Reward Models are built in chapter 4 to calculate the performability of our system.

Previous work in the field, which acted as catalyst for the conduct of this thesis will be presented in the chapter that follows.

Chapter 2

Background and Motivation

2.1 INTRODUCTION

Wireless sensor networks (WSN) are dense wireless networks of small, low-cost sensors. They consist of large numbers of inexpensive energy-constrained devices, and expect to find a wide range of applications.

Wireless sensor networks (WSN) differ substantially from traditional ad-hoc wireless networks, and they are characterized by the following criteria:

- The typical mode of communication in a WSN is from multiple data sources to a data sink;
- The data being collected by multiple sensors is based on common phenomena. Therefore it is likely to be some redundancy in the data being communicated by the various sources in sensor networks;
- In most envisioned scenarios the sensors are not mobile (though the sensed phenomena may be), so the nature of the dynamics in WSN and mobile ad-hoc networks is different.

In summary, WSN have severe energy constraints, redundant low-rate data, and many-to-one flows. For these reasons the end-to-end routing protocols that have been proposed for Mobile Adhoc Networks (MANET) in recent years are not suitable for wireless sensor networks [2]. Alternative approaches are required.

This thesis focuses on finding a reliable routing Protocol that can be used in wireless sensor networks. In fact, we aim to build a robust, long live and low latency network. Since we assume that all nodes locations are fixed for the duration of their lifetime, mobility is not a constraint in our system. While In chapter 3 we explain in details our proposed routing algorithm, in what follows we describe the existing and previous work in the field.

2.2 ROUTING ALGORITHM

2.2.1 Low-Energy Adaptive Clustering Hierarchy Protocol

Low-Energy Adaptive Clustering Hierarchy (LEACH) [1], a clustering-based protocol that utilizes randomized rotation of local cluster base stations (cluster-heads) to evenly distribute the energy load among the sensors in the network.

LEACH assumes every node can directly reach a base station by transmitting with sufficiently high power. However, one hop transmission directly to a base station can be a high power operation and is especially inefficient considering the amount of redundancy typically found in sensor networks.

LEACH organizes nodes into clusters with one node from each cluster serving as a cluster-head. Nodes first send sensor readings to their cluster-head, and the cluster-head aggregates or compresses the data from all its “children” for transmission to a base station. If cluster-head selection is static, those unlucky nodes chosen as cluster-heads would quickly run out of energy and die.

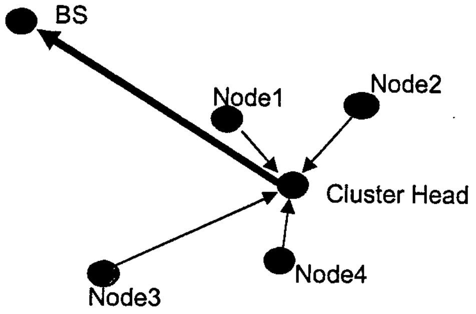


Figure 2.1: LEACH Protocol

LEACH does not address possible network partitioning that occur during random cluster generation in case of no cluster head is elected within range of a given node. A routing problem can also happen by multiple cluster head changing. Since it is unrealistic to assume that all our nodes are capable of long-range communication, LEACH cannot be implemented in our WSN for Petroleum system.

2.2.2 Sensor Protocol for Information via Negotiation

Sensor Protocol for Information via Negotiation (SPIN) [23], are protocols that aim at disseminating information among all the sensor nodes by using information descriptors for negotiation prior to transmission of the data. These information descriptors are called meta-data and are used to eliminate the transmission of redundant data in the network. In SPIN, each sensor node also has its own resource manager that keeps track of the amount of energy that the particular node has.

SPIN is based on controlled flooding where traditional flooding problems like implosion, overlap, and resource blindness is handled.

The Protocol starts when a node receives data then it disseminates by sending out an ADV (Advertisement for Data) message to its neighbor, if the adjacent node does not possess the data it responds by sending an REQ (request for Data) message then the node sends the missing Data. SPIN has access to the current energy level of the node and adapts the protocol it is running based on how much energy is remaining.

The SPIN family of protocols uses the following three messages for communication [16]:

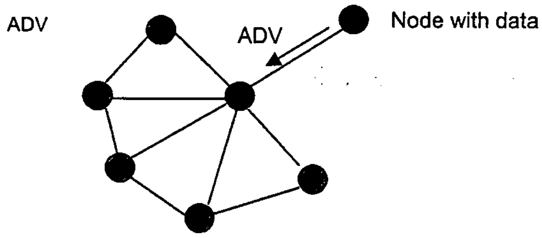
- **ADV** : When a SPIN node has some new data, it sends an ADV message to its neighbors containing meta-data (data descriptor);
- **REQ** : When a SPIN node wished to receive some data, it sends an REQ message;
- **DATA** : These are actual data messages with a meta-data header.

SPIN Protocol is more efficient than the standard flooding and gossiping protocols but it has also it's own limitation, some of these are:

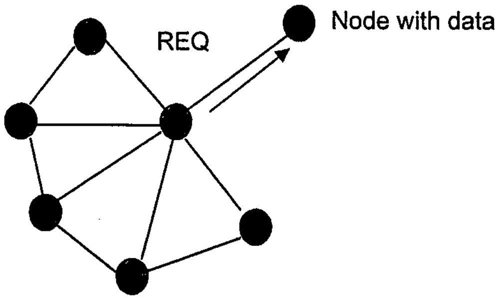
- Nodes are always active (idle nodes still consumes energy when listening to "Adv" messages);
- Network lifetime has not been studied as a performance metric;
- High degree nodes may consume more energy, reducing lifetime of the node.

As a result of these problems SPIN could not be considered as a possible solution for our system.

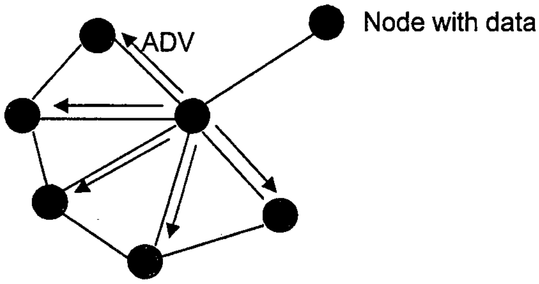
Spin Algorithm



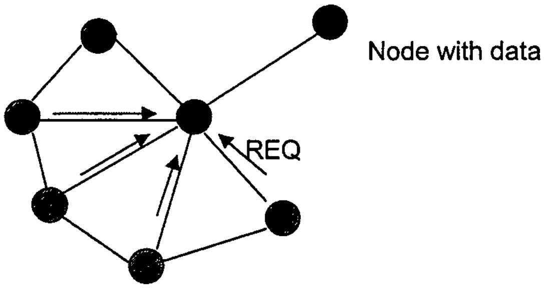
The Node with data advertises to all its neighbors



The neighbor requests for data and data is sent to it



The node which received data sends advertisements to all its neighbors



Now the neighboring nodes who received the advertisements request for data.

Figure 2.2. SPIN Protocol

2.2.3 Direct diffusion (gradient routing protocol)

Directed diffusion [18] is a data-centric routing communication paradigm for drawing information out of a sensor network.

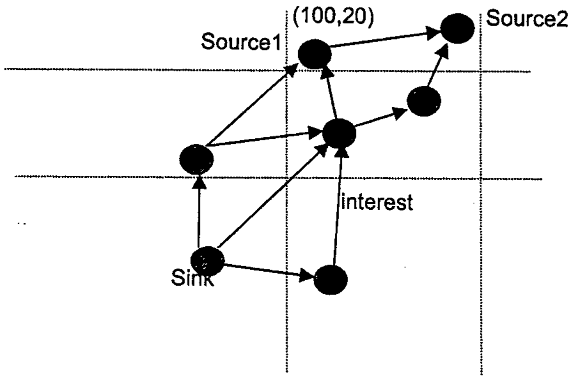
A sensing task is propagated throughout the network for named data by a node and data, which matches this interest, is then sent towards this node. One important feature of the data diffusion paradigm is that the propagation of data and its aggregation at intermediate nodes on the way to the request-originating node are determined by the messages that are exchanged between neighboring nodes within some distance (localized interactions).

Nodes receiving the same interest from multiple neighboring nodes may propagate events along the corresponding multiple links. Interests initially specify a low rate of data flow, but once a base station starts receiving events it will reinforce one (or more) neighbor in order to request higher data rate events. This process proceeds recursively until it reaches the nodes generating the events, causing them to generate events at a higher data rate. Alternatively, paths may be negatively reinforced as well. There is a multipath variant of directed diffusion [4] as well. After the primary dataflow is established using positive reinforcements, alternate routes are recursively established with maximal disjointedness by attempting to reinforce neighbors not on the primary path.

The Gradient setup phase is expensive (maintenance is needed) and it requires a cost value for every potential destination in the network (scalability).

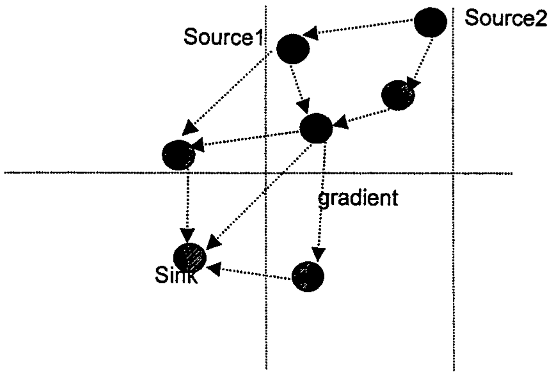
The cost to each sink much be discovered by flooding (energy consumption). Moreover the best paths might be used too often and the nodes within the range of base station may die quickly thus it is not energy aware. Finally, directed diffusion is not scalable as soon as the network enlarges and it is costly in terms of battery life.

Direct diffusion Algorithm



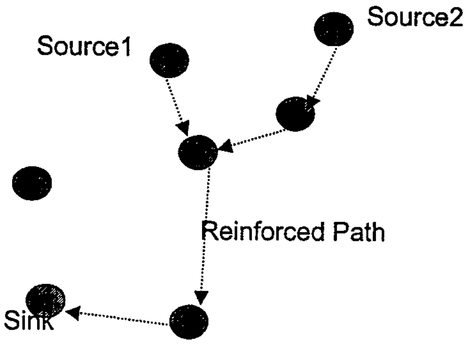
Get me info of all vehicles moving in (100,20)

Gradients



Gradients Established

Reinforced Path



Data transferred via reinforced path

Figure 2.3. Direct diffusion Protocol

2.3 NETWORK SIMULATOR 2 (NS2)

NS2 [9][10] is an open-source simulation tool that runs on different platforms. It is a discreet event simulator targeted at networking research and provides substantial support for simulation of routing, multicast protocols and IP protocols, such as UDP, TCP, RTP and SRM over wired and wireless (local and satellite) networks. It has many advantages that make it a useful tool, such as support for multiple protocols and the capability of graphically detailing network traffic. Additionally, NS2 supports several algorithms in routing and queuing. LAN routing and broadcasts are part of routing algorithms. Queuing algorithms include fair queuing, deficit round-robin and FIFO.

NS2 started as a variant of the REAL network simulator in 1989. REAL is a network simulator originally intended for studying the dynamic behavior of flow and congestion control schemes in packet-switched data networks. Currently NS2 development by VINT group is supported through Defense Advanced Research Projects Agency (DARPA) with Simulation Augmented by measurement and Analysis for Networks (SAMAN).

SAMAN and through NSF with Collaborative Simulation for Education and Research (CONSER), both in collaboration with other researchers including Center for Internet Research (ICSI). NS2 is available on several platforms such as FreeBSD, Linux, SunOS and Solaris. NS2 also builds and runs under Windows.

2.4 CARRIER SENSE MULTIPLE ACCESS/COLLISION AVOIDANCE

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is a protocol for carrier transmission in 802.11 networks [12]. Unlike CSMA/CD (Carrier Sense Multiple Access with Collision Detect), which deals with transmissions after a collision has occurred, CSMA/CA acts to prevent collisions before they happen.

In CSMA/CA, as soon as a node receives a packet that is to be sent, it checks to ensure that the channel is clear (no other node is transmitting at the time). If the channel is clear, then the packet is sent. If the channel is not clear, the node waits for a randomly chosen period of time, and then checks again to see if the channel is clear. This period of time is called the backoff factor, and is counted down by a backoff counter. If the channel is clear when the backoff counter reaches zero, the node transmits the packet. If the channel is not clear when the backoff counter reaches zero, the backoff factor is set again, and the process is repeated.

2.5 ARCHITECTURE OF 802.11

The IEEE 802.11 [24] standard permits devices to establish either peer-to-peer (P2P) networks or networks based on fixed access points (AP) with which mobile nodes can communicate. Hence, the standard defines two basic network topologies: the infrastructure network and the ad hoc network.

The infrastructure network is meant to extend the range of the wired LAN to wireless cells. A laptop or other mobile device may move from cell to cell (from AP to AP) while maintaining access to the resources of the LAN. A cell is the area covered by an AP and is called a "basic service set" (BSS). The collection of all cells of an infrastructure network is called an extended service set (ESS). This first topology is useful for providing wireless coverage of building or campus areas. By deploying multiple APs with overlapping coverage areas, organizations can achieve broad network coverage. WLAN technology can be used to replace wired LANs totally and to extend LAN infrastructure. A WLAN environment has wireless client stations that use radio modems to communicate to an AP.

The client stations are generally equipped with a wireless network interface card (NIC) that consists of the radio transceiver and the logic to interact with the client machine and software. An AP comprises essentially a radio transceiver on one side and a bridge to the wired backbone on the other. The AP, a stationary device that is part of the wired infrastructure, is analogous to a cell-site (base station) in cellular communications. All communications between the client stations and between clients and the wired network go through the AP. The basic topology of a WLAN is depicted in Figure 2.4.

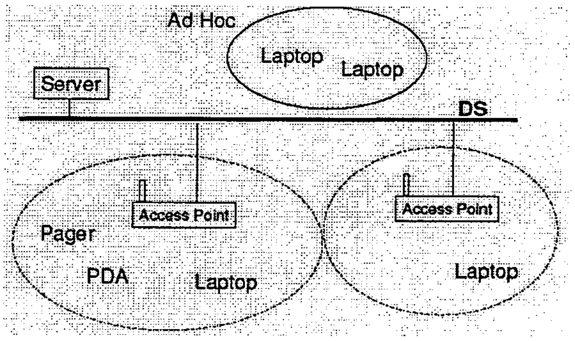


Figure 2.4. : Fundamental Wireless LAN topology

2.6 PERFORMABILITY CONCEPT

Performability defines the combined performance and reliability of the system. The main problem in defining the performability formula is the different scales of the performance and reliability factors. When evaluating a system it is necessary to describe what the actual system is and does with respect to what the system is specified to be and do. The performance of a system depends on how its resources are altered by faults. Therefore, performance and dependability are the key elements in the evaluation process. Performance of a system may be described as the quality of proper output (quality could be instruction execution rate) that the system provides, where proper output means that the output satisfies the user specifications.

Dependability on the other hand takes into account measures of reliability and availability. Reliability is defined, as the probability that the output remains proper during a particular observation period while availability refers to the fraction of time the output remains proper during the same period.

Performability evaluation is especially important in degradable (i.e., the quality of output decreases but the output remains proper) systems, those which continue to operate failure-free in the presence of fault-caused errors by lowering quality of service. In degradable systems, performance and dependability are especially interconnected. In the case of a degradable three microprocessor system, the performance of the system at best when all three processors are running correctly. One processor may break down, causing the other two to process 50% more work. This causes a decrease in performance of the overall system because of the failure of the one microprocessor. This failure would be predicted by quantifying the dependability of the system. Thus, the performability of a degradable system predicts performance based on dependability.

Performability was first applied in the evaluation of highly reliable aircraft control computers used of the United States National Aeronautics and Space Administration (NASA). The idea was to design a computer system that abandoned less important work if a loss of resources occurred. This allowed ensuring that the important work would be performed for the duration of the aircraft's flight. The need for a highly accurate evaluation of the flight computers combined with the degradable nature of the system gave vital importance to the concept of performability as a form of evaluation.

The most common solution method for performability is based on reward models. This model associates reward rates with state occupancies. The reward rate can be thought of as the work accomplished in that specific state. By combining the model of a stochastic (random) process for a given system with the reward rates for that system, a reward model results.

The total reward accumulated over a given time period is the performance of the system. Performability then results by combining this performance with a Markov process representing the dependability of the system (A Markov process is one whose future state does not depend on its past states, but only on the current state).

If a given system fails quite frequently, and generally runs at a lower performance level its performability system Probability Distribution Function (PDF) would tend to look more like in figure 2.5. Otherwise when the system runs at a high performance with seldom failure, its performability system PDF would look more like in figure 2.6 below

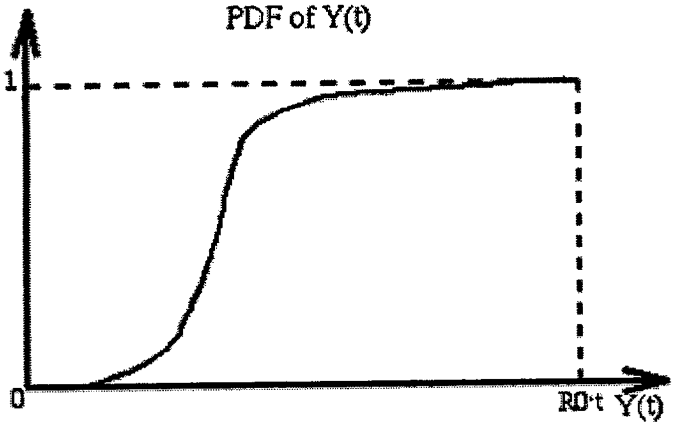


Figure 2.5 Low performability system PDF

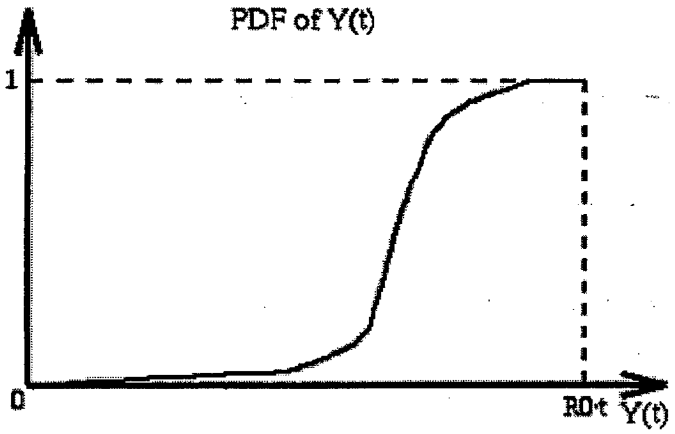


Figure 2.6 High performability system PDF

It becomes clear that in the latter case, there is less probability of only accomplishing up to a certain level of work, (i.e. there is a greater probability of accomplishing a higher level of work). Thus the latter case demonstrated a higher performability.

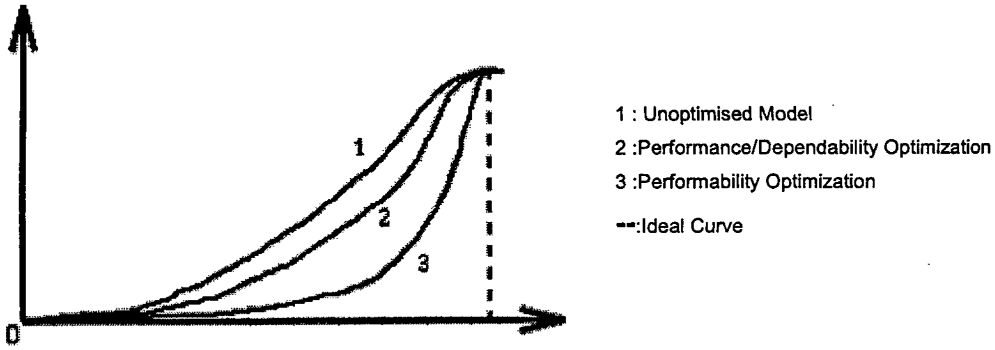


Figure 2.7 Comparative PDF's

An ideal performability case is a failure free system that operates at maximum capacity all the time. In this case, there is zero probability of accomplishing anything less than the maximum amount of work, and

100% probability of doing that. This distribution is described by the impulse shape shown in figure 2.7. This is the ideal shape that any optimization technique strives to attain. Figure 2.7 above illustrates how the different evaluation techniques might compare to each other.
In summary, performability provides an excellent mean finding the most efficient use of resources that ensure mission completion (considering time factors) [8].

Finally, performability became more widely used in its original disciplines of computing and communication systems, complex tools were developed in order to design the evaluation models. As these tools develop, still they are starting to be applied to an increasingly wide variety of systems, well beyond the original scope of computing and communication. These systems include flexible manufacturing systems, vehicle-highway systems, and economic systems among many others. Performability is expected to increase exponentially in its importance as an evaluation method into the twenty-first century, especially as the tools used to implement these evaluations become better developed [15].

2.7 Research Motivation

Despite the plethora of studies evaluating the performability of Computer and Communication Systems, little is known about the performability of Wireless Sensor Networks (WSN), although nodes failure could often occur in such a network. Therefore it is a challenging opportunity to analyze the performability of such a system and to evaluate its average delay and energy consumption.

Petroleum systems typically implement Wired Sensor Networks. Wireless Sensor Networks for petroleum systems is still in development phase. In particular, routing algorithms has not been analyzed for these particular systems. That is why we have decided to tackle this problem in this research work.

With the conclusion of this chapter, emphases will be shifted to describe our proposed routing protocol for WSN's (SR-WSN) and present our major simulation results using NS2 in the chapter that follows.

Chapter 3

Proposed Routing Algorithm for WSN:

Smart Routing in WSN (SR-WSN)

3.1 INTRODUCTION

This chapter is divided into four different sections. The next section gives an overall description of the system under analyses.

While in section 3.3 we presented an analysis of the communication requirements in our research, in section 3.4 we describe the algorithm of a new routing protocol designed for Wireless Sensor Networks, which takes into consideration the different performance constraints of the system. This includes the delay minimization of the delivered packets, the lifetime of the sensor nodes and the reliability constraints. These considerations are illustrated and evaluated via simulation and presented in section 3.5 of this chapter.

3.2 OVERALL SYSTEM DESCRIPTION

The assumption of this thesis is based on the research work conducted by Maalouf and Aakvaag [17]. The thesis describes the platform and basis of the communication system next. The system described does not represent an actual platform, but is typical of the installations found in the petroleum activity [17].

3.2.1 Platform topology

As in [17], we assume a communication area of 200 x 200m. This is larger than typical platforms but representative of many on-shore installations. Furthermore, we assume that a total of 1500 sensors need to communicate with a centrally placed controller. We propose to divide our system into N levels where the average density of the sensors nodes is (approximately) the same in all levels.

The depth, *d*, of each level is limited by the radio transmission range *R* of a given sensor ($d \leq R$). Based on the expected coverage of IEEE 802.11 in the 2 Mbps band we consider N = 5, d = 20 m. and R = 30 m.

3.2.2 Communication requirements

The data transfer is unidirectional from the sensors to the controllers. Each sensor generates approximately 20 bytes of data every τ seconds which is relayed by the intermediary sensors to the central controller. A sensor node in our system has several tasks: Sensing the environment, processing the information, generating and forwarding data, and relaying traffic as an intermediate node in the multi-hop network.

Power saving is achieved in [17] by putting nodes to sleep for a large percentage of its operational time, only waking them up into active mode for brief periods [7]. This issue was not tackled in our research and could be part of any future work in the field.

To keep the system design simple, the present thesis assumes a beaconless communication system with acknowledged transmission. It is important to note here, that this assumption is different than the "unacknowledgment" assumption in [17]. This is mainly due to the fact that we are concerned here about the reliability of the system, while reliability was not an issue of concern in [17].

Sensing processes are done at level N. All level N sensors will start transmitting their data using the CSMA-CA MAC scheme. Data will then aggregate towards the controller, with level n transmitting to level n-1, this chain of transmission processes will start periodically every τ seconds.

3.3. ANALYSIS

3.3.1 Radio Resource Management

In order to reduce the complexity of the system we will assume that sensors are all working on the same frequency. By considering a depth *d* of 20 m. between two levels and a radio range R of 30 m, thus every sensor will access at least one sensor on the next level. Also, at a given time, a given sensor can intervene with two neighboring sensors on the same level (situated at its right and its left).

3.3.2 Sensor Distribution

Assuming that the numbers of sensors in the different level is approximately proportional to the areas of these levels, one finds that we need approximately 60 sensors, 180 sensors, 300 sensors, 420 sensors and 540 sensors for level 1,2 ,3,4 and 5 respectively [17]. (See figure 3.1 and figure 3.2 below)

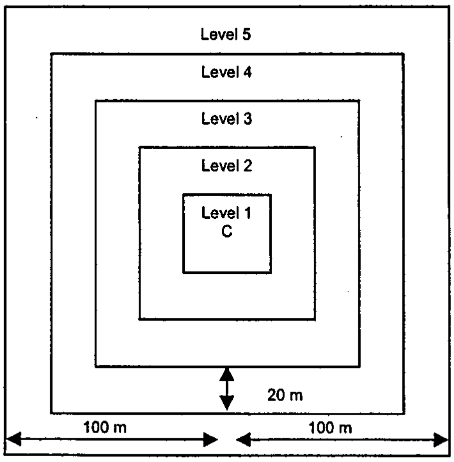


Figure 3.1. Platform Topology

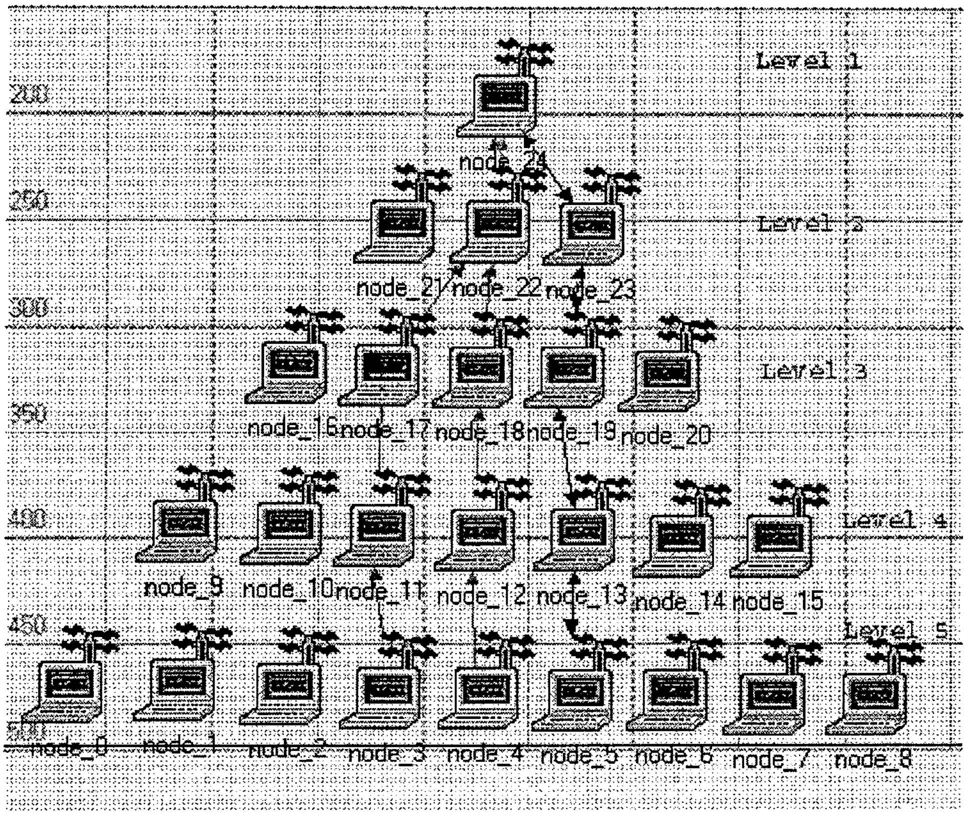


Figure 3.2. Allocation of sensors

3.4 TRAFFIC LOAD ANALYSIS

The sensors at level 5 will not receive any messages since they are at the bottom level and will send only messages to the level above them. In order to explain our algorithm, we shall describe next a possible scenario of events:

Let us assume that *node 5* (in figure 3.5 above) at level 5 sends a packet "p" to the above level, *node 4* and *node 6 (at level 5)* will get a copy of packet "p" (as they are within the coverage range), the following possible situations exist:

Suppose that *node 13* is the next sensor (within the radio range) at level 4 which receives packet "p", *node 13* will send back an acknowledgment to the source node. Thus *node 5* will stop sending packet "p", and will not send any trigger to any other node on the same level. In this case all level 5 nodes in the coverage range which received packet "p" will discard this packet after a short interval of time.

Node 13 at level 4 will now be the current source node and all rules applied for *node 5* applies now for this node.

Now assume that n*ode 13* retransmit the packet "p" to the level above it. All nodes within the coverage range, should receive "p". This includes the two neighbors of node 13 on the same level i.e *node 14* and *node 12.* Now suppose it should reach *node 19* at level 3 if *node 19* is still alife.

If *node 13* does not receive any acknowledgment from *node 19*, or any other node on level 3, then *node 13* will send a trigger "REQ" to *node 14* (on its right). This trigger will push *node 14* to retransmit packet "p" to the level above it. Now if acknowledgment is sent back to node 14 this acknowledgment will be forwarded to *node 13*. If no "ack" is received by *node 14* (nor *13*), then *node 13* will trigger its other neighbor on the left i.e *node 12*, by sending another "REQ" message to *node 12*. Again, *node 12* will retransmit packet "p" and so on.

Node 12 applies the same algorithm till packet "p" reach the sink sensor (*node 24*).

3.5 COMPUTER SIMULATION

In what follows we will describe the main Simulation results using NS2 simulator, we will assume in these simulations that we have 39 sensors nodes distributed to 5 different levels as described previously in paragraph 3.3.2 (see figure 3.1 and figure 3.2 above).

3.5.1 Delay Analysis

- First, we will assume that one sensor node is generating packets to destination. In figure 3.3 below we notice that the delay increases when the node failure increases in the system. For instance, if no node failure occurs, the average delay of 1000 packets is about 48 ms, while if the node failure percentage is 39% (15 nodes out of 39) then the delay in the system increases to 128 ms;
- Second, we increase the number of generating source nodes from 1 to 3 nodes. As we can see in figure 3.4 below, the average delay increases about 44 % in comparison with a system consisting of one generating source node. This is obvious since the packets coming from different sources will affect each other and increase the total average delay in the system.

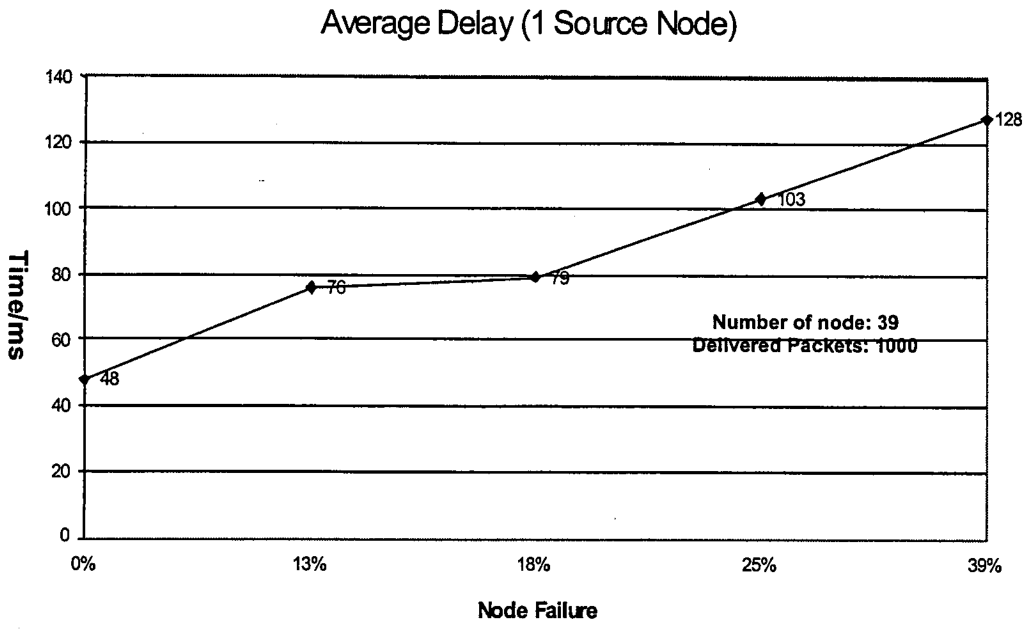


Figure 3.3 Avg. Delay (1 Source)

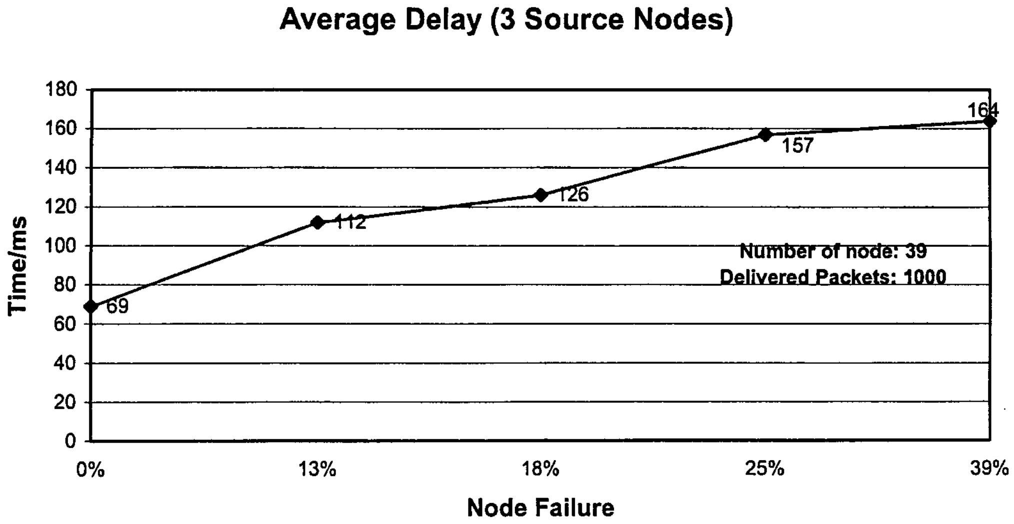


Figure 3.4 Avg. Delay (3 Sources)

3.5.2 Energy Analysis

This paragraph considers one source node and then increases the number of generating nodes to three. In figure 3.5 below we suppose that one node is generating packets to destination. We notice that the Average Energy consumption increases when the node failure increases in the system: if we have no node failure, the average energy consumption for a 1000 packets is 80.6 joules, while if the node failure percentage is 39 % (15 nodes out of 39) then the average energy consumption increases to 143.4 joules. This is due to the different routes taken when a given node makes several trials to reach the destination. On the other hand, in figure 3.6 the number of generating nodes increases from one to three. We notice here that the average energy consumption increases about 170 % in comparison to a system consisting of one source node. This is understandable for the following reasons:

- The traffic generated has increased 3 fold;
- The number of collision has increased;
- The need to try several routes has increased.

The main result yielded from figure 3.7 and 3.8 is that the energy consumption in the middle layers is much higher then that of the bottom or the top layer. This is mainly due to the functionality of these layers: sending, receiving and forwarding packets; hence, the middle layers are more prone to failure then peripheral sensors.

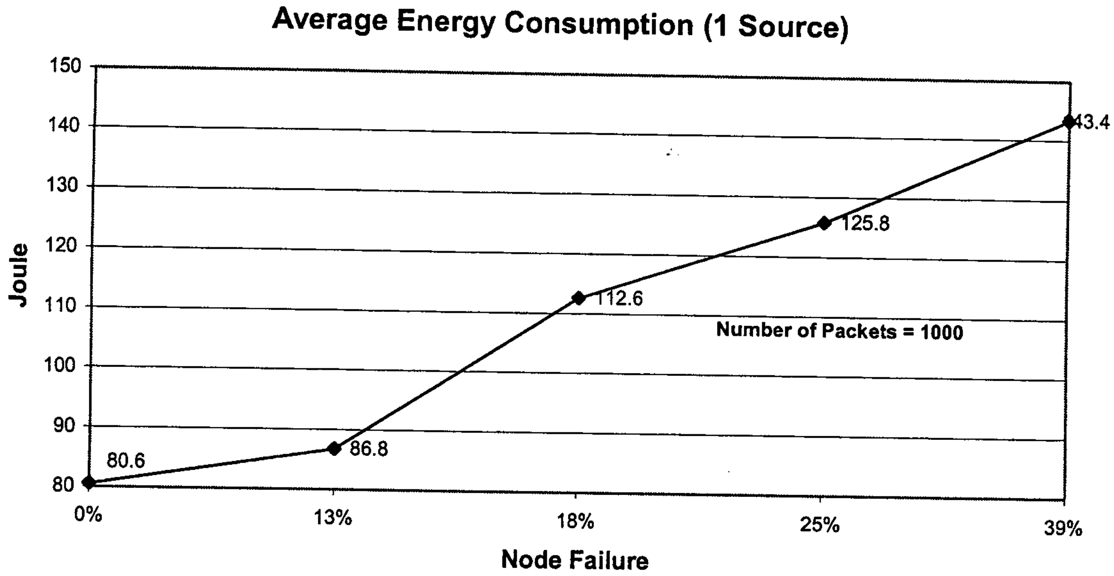


Figure 3.5 Avg.Energy (1 Source)

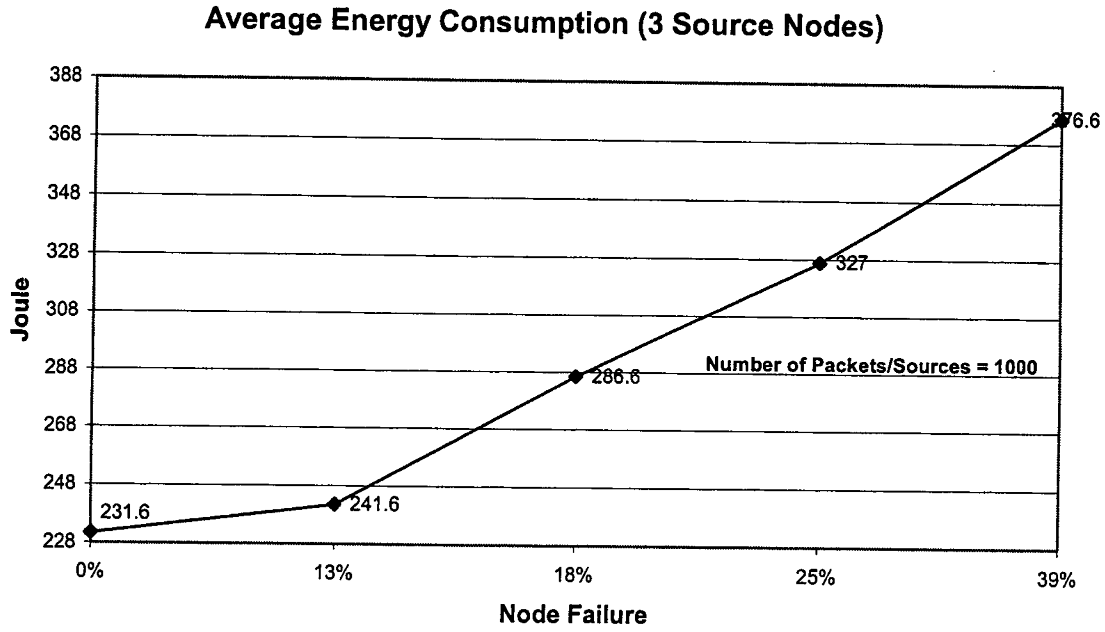


Figure 3.6 Avg.Energy (3 Sources)

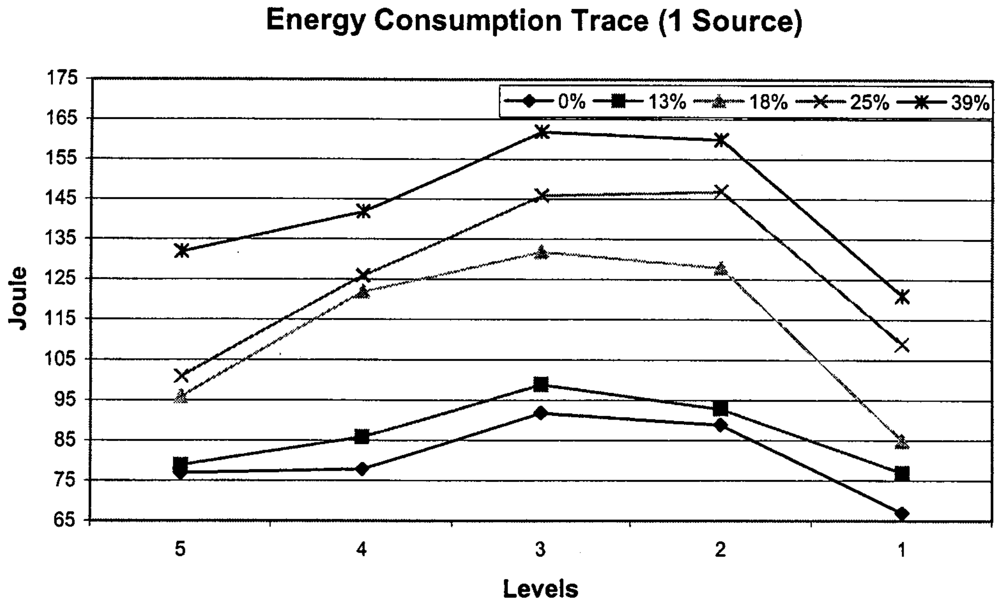


Figure 3.7 Energy Trace (1 Source)

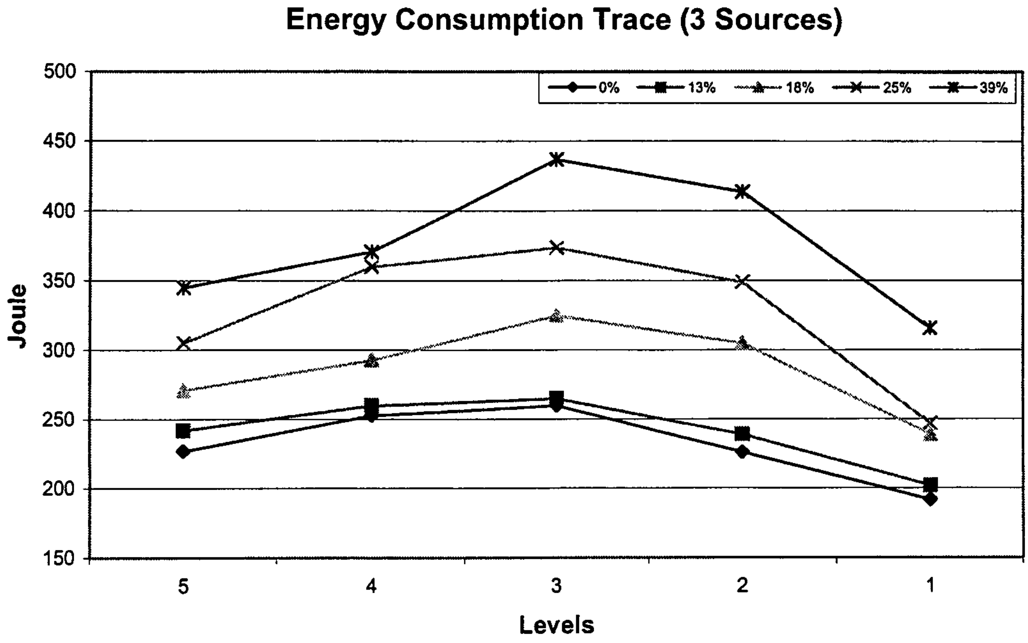


Figure 3.8 Energy Trace (3 Sources)

The performance analysis of a system with node failures cannot be complete and thorough if delay analysis and energy consumption are the only parameters studied, especially that with such systems many packets are lost and never reach the destination. That is why we decided to study the performability of the system, which combines performance and reliability issues together. This will be the main topic of chapter 4.

Chapter 4

Performability

4.1 INTRODUCTION

Performability is a composite measure of a system's performance and its dependability. Performance is similar to "quality of service (QoS), provided the system is correct" [13]. Furthermore Performance modeling involves representing the probabilistic nature of user demands and predicting the system capacity to perform, under the assumption that the system structure remains constant. Dependability is an all-encompassing definition for reliability and availability. Dependability modeling deals with the representation of changes in the structure of the system being modeled, which are generally due to faults, and how such changes affect the availability of the system. Performability modeling, then considers the effect of structural changes and their impact on the overall performance of the system [23].

4.2 PERFORMABILITY MODELING

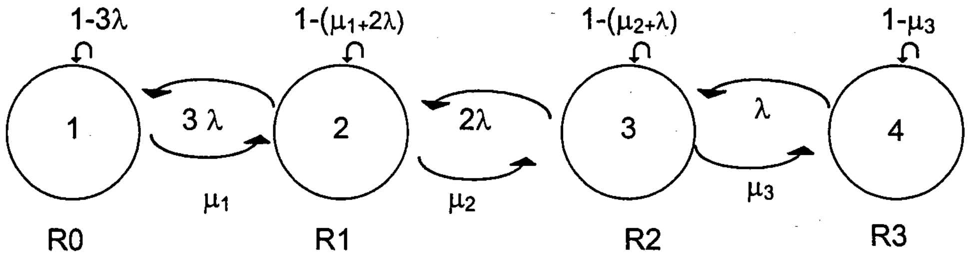
4.2.1 Overview

The most commonly used performability model today is the Markov Reward Model (MRM). The MRM is built up of two distinct models: the behavior model and the reward structure [13]. The behavior model, describes the possible behavior of the system. A degradable system, depending on the faults that occur, can be in different states at different times.

Each different state, representing in our case a different path from the source node to the sink, has a certain performance level associated with it. The amount of performance achievable has a certain reward related to it. This reward rate quantifies the ability of the system to perform. If the system goes to a state with a higher reward, a higher performance level is reached. The set of these rewards, associated to the individual states, make up the reward structure.

In Figure 4.1 below we have an MRM model with four states. The arrows in the state transition diagram describe the possible transitions between each state. The system will only spend a certain amount of time in each state, called the holding time. The holding times in each state are typically exponentially distributed, therefore we can associate to each a probability of changing state, over time. These are the labels on the transition arrows, λ and μ. λ refers to a state switch due to a failure caused by faults occurring in the system, μ to one caused by repairs to the system. From these, and the state transition diagram, we can build up Q, the Generator matrix shown in figure 4.1. Each transition rate from state i to state j is denoted by the term q (i,j) in the matrix.

State Transition Diagram:



Reward Structure

Generator Matrix:

$$Q=\begin{bmatrix} 1-3\lambda & \mu_1 & 0 & 0 \\ 3\lambda & 1-(\mu_1+2\lambda) & \mu_2 & 0 \\ 0 & 2\lambda & 1-(\mu_2+\lambda) & \mu_3 \\ 0 & 0 & \lambda & 1-\mu_3 \end{bmatrix}$$

$$R=\begin{bmatrix} R0 \\ R1 \\ R2 \\ R3 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 1 \\ 0 \end{bmatrix}$$

Figure 4.1 The Markov reward Model

4.2.2 Simulation Results

Concerning figure 4.2, we can see that there are four states describing the system we are modeling. These are:

- State number 1: When a given packet P is sent successfully from the first trial;
- State number 2: When the first trial fails (due to sensor failure) and the second trial succeeds. (i.e. after we trigger the right neighbor);
- State number 3: When the first and second trial fails and the third trial succeeds. (i.e. after we trigger the left neighbor);
- State number 4: All previous trials fail.

The following rewards are assigned for the different states, where the highest reward rate (R0) is associated to state 1, and the lowest (R3) to state 4:

R0: performance level = 3.0 (This is the best case where we have the lowest delay and energy consumption);
R1: performance level = 2.0;
R2: performance level = 1.0;
R3: performance level = 0.0 (This is the worst case where packet P is lost).

In what follows we pick up randomly one packet P from the thousand packets sent and we trace its path from source to destination.

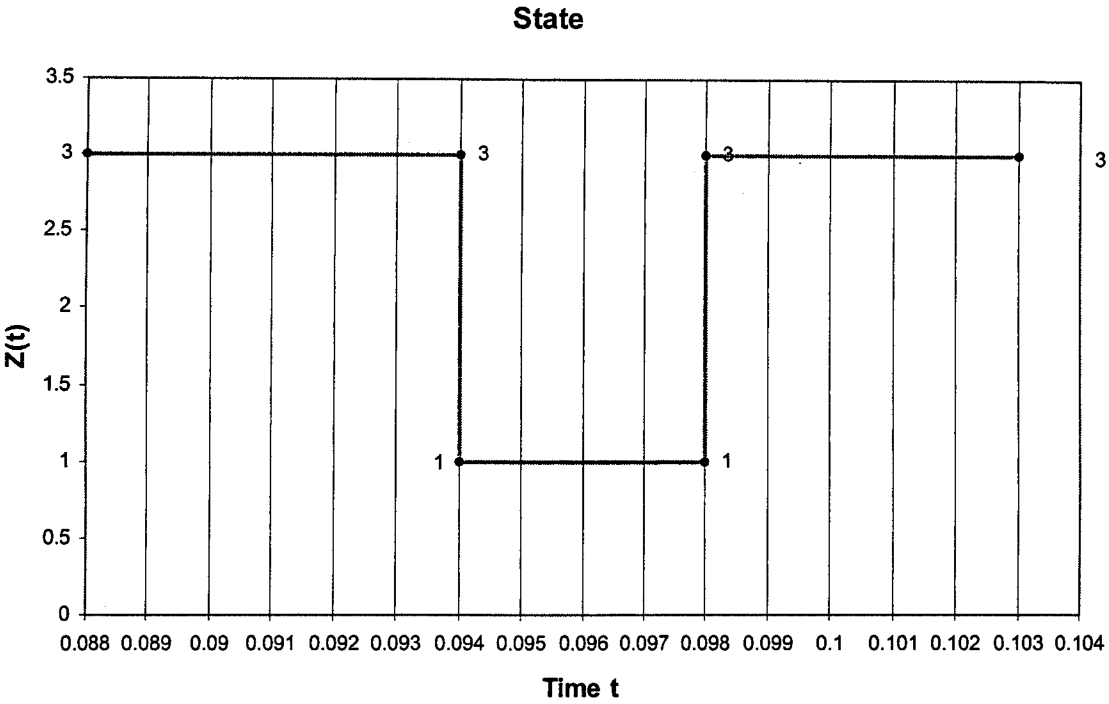
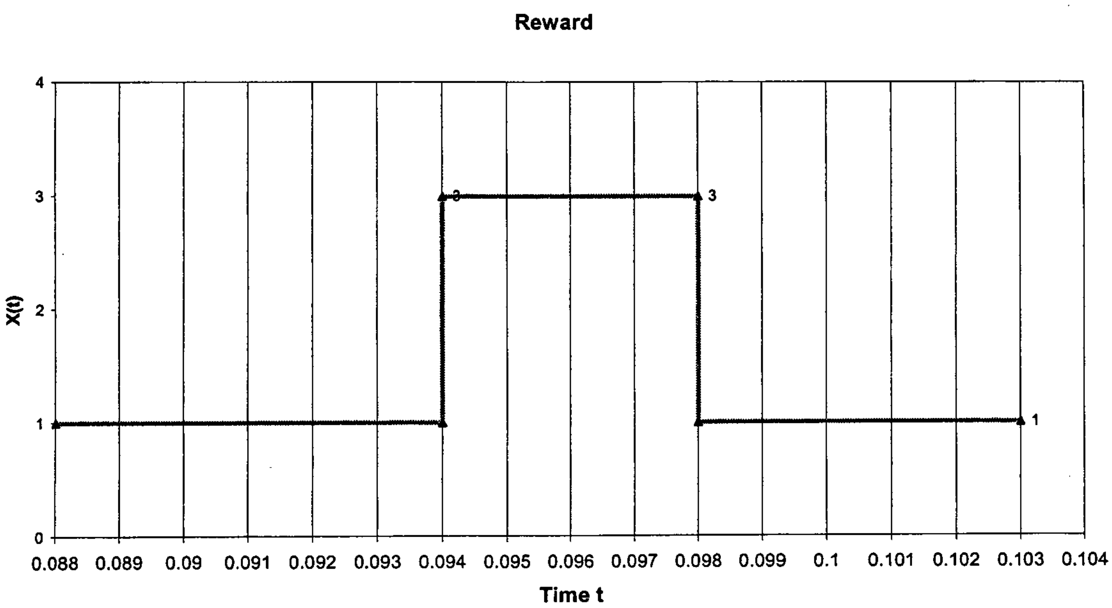


Figure 4.2. State evolution of a given Packet



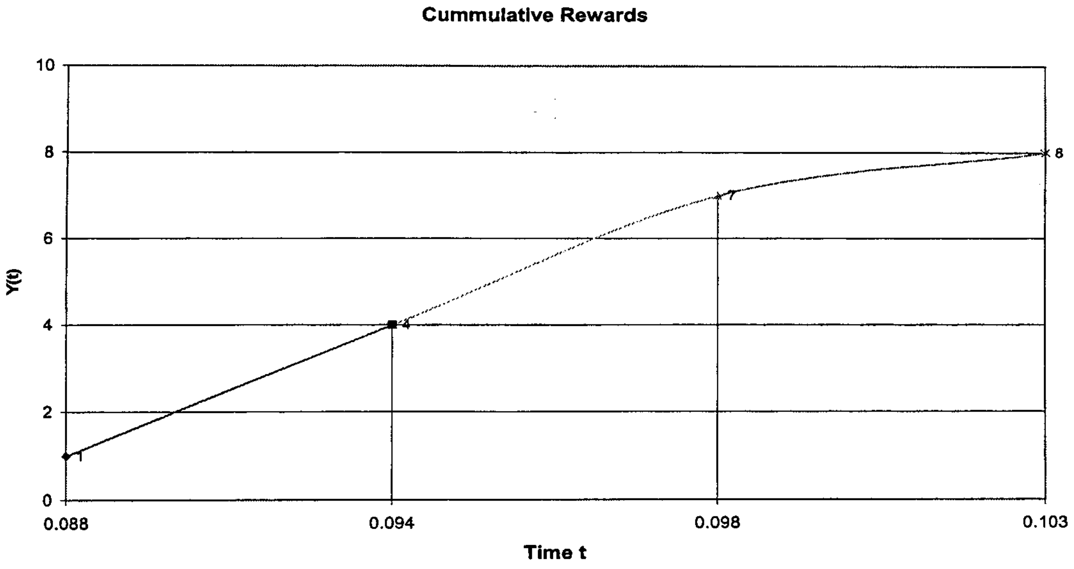


Figure 4.4. Cumulative Rewards of a given Packet

Using simulation we build the following drafts:

Z (t) traces the trajectory of a given packet from source to destination. It will be described in terms of state. In fact, in Figure 4.2, Z (t) represents the evolution (path) of the system in time. It is the state of the system at any time t. In this case the path it describes goes from state $3 \Rightarrow 1 \Rightarrow 1 \Rightarrow 3$. We notice that the lengths of the horizontal lines in Z (t) are different. This is because the holding time in each state is random.

On the other hand X(t) in figure 4.3 defines the reward rate of the system at time t, in this case the rate goes from $1 \Rightarrow 3 \Rightarrow 3 \Rightarrow 1$. This follows directly from the state changes shown in figure 4.2 of Z(t).

The plot y(t) shown in figure 4.4 is deduced from the previous two graphs. It is the accumulated reward until time t, which is the area under the X(t) curve. The higher the reward rate in X(t), the steeper the slope of the Y(t) curve, which implies less delay time and less hops to reach the destination.

By calculating the accumulated reward curves y(t) for 200 packets, we managed to determine the performability of the system. In fact, the solution to the performability model is found by evaluating the Probability Distribution Function (PDF) of accumulated reward y(x,t) as can be seen in figure 4.5 below.

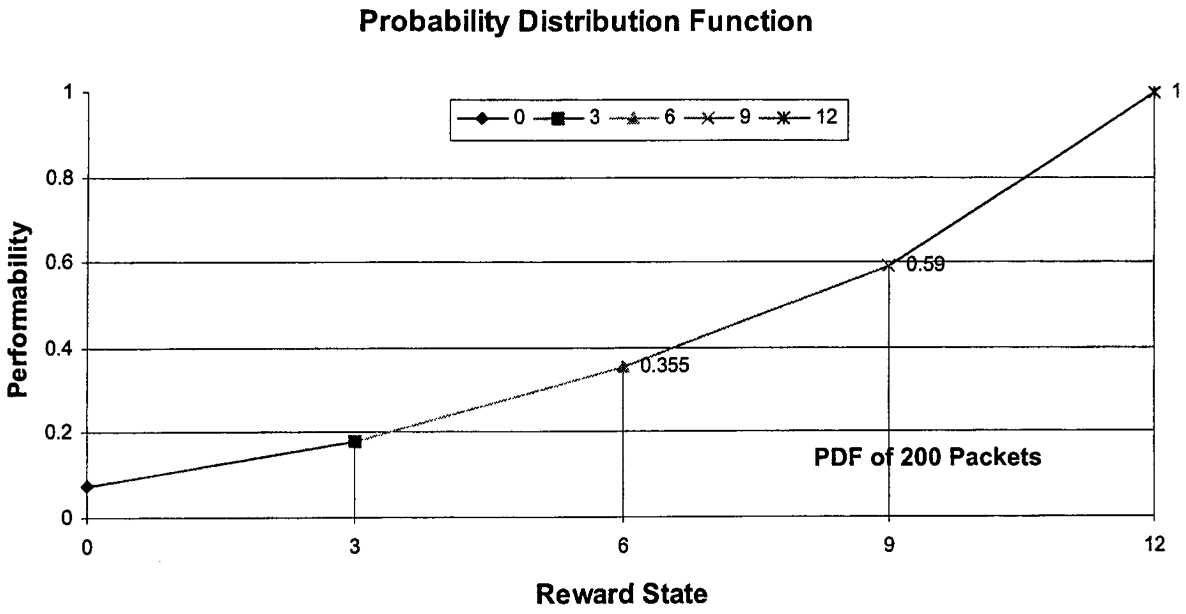


Figure 4.5. The Probability Distribution Function of (Y(t)

The Distribution of accumulated reward by time t evaluated at x is denoted by: Y (x,t) = Prob [Y(t) ≤ x]

Informally performability can be defined as the probability that the system does a certain amount of useful work over a mission time t. Hence, the solution to the performability model is deduced by evaluating the PDF of accumulated reward y (x,t).

The PDF in figure 4.5 illustrates the average system probability distribution. It shows how the probabilities of accomplishing the work are distributed. We should note here that a system which operates for the majority of time at fully operational status will have a higher density of probabilities nearer to the maximum accomplishment level (y(t)=12) in our case, the furthest right of the PDF, which represents the greatest possible accumulated reward).

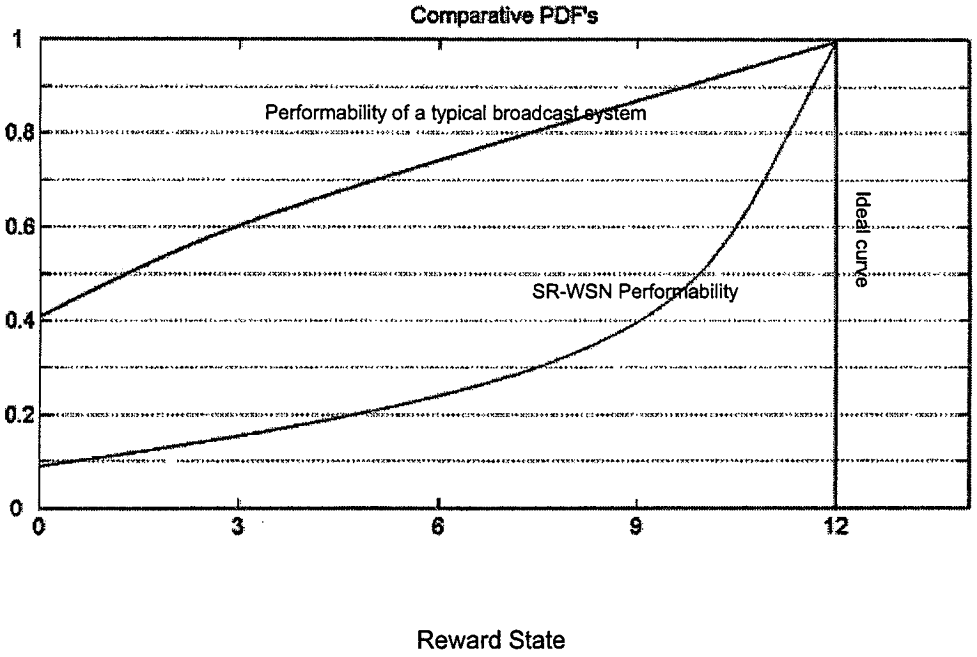


Figure 4.6. Comparative PDF's

In figure 4.6 above we compare the performability of SR-WSN protocol to that of a typical broadcast system, also in figure 4.6 we show the ideal performability curve.

The ideal case is a failure free system that operates at maximum capacity all the time. In this case, there is zero probability of accomplishing anything less than the maximum amount of work, and 100% probability of doing that. The impulse shape shown in figure 4.6 describes this distribution, this is the ideal shape that optimization techniques attempt to attain. Figure 4.6 illustrates how the different evaluation techniques might compare to each other. As can be seen, performability provides the best means to find the most efficient use of resources that ensure mission completion (considering time factors).

It is clear from figure 4.6 above that SR-WSN outperforms a broadcast system. Let us take an example:

$\text{prob}(y(t) \leq 9) = 0.4 \Rightarrow \text{prob}(y(t) > 9) = 0.6$ (SR-WSN)

while we have

$\text{prob}(y(t) \leq 9) = 0.9 \Rightarrow \text{prob}(y(t) > 9) = 0.1$ (broadcast System).

Since y(t) is the accumulated reward at time t this proves that SR-WSN outperforms any broadcast system usually used in wireless networks.

Chapter 5

Conclusion & Future Work

5.1 INTRODUCTION

This chapter summarizes the main results of the study and provides recommendations for future research in the area.

5.2 RESULTS OF CHAPTER 3

In this chapter, we have presented a smart, dependable routing mechanism for Petroleum installations (SR-WSN) between sensor nodes and a base station in a wireless sensor network environment. SR-WSN tolerates failure of random individual nodes in the network or a small part of the network by dynamically discovering new routes when nodes fail. It also takes into consideration the energy consumption constraints of the sensors. We have simulated SR-WSN using Network Simulator (NS2). We calculated the delay and the energy consumption in such systems, as a function of the percentage of node failures.

The main contributions are

i. Proposing a routing algorithm for Wireless Sensor Networks (SR-WSN);
ii. Building a simulation model using Network Simulator (NS2).

5.3 RESULTS OF CHAPTER 4

This research did not compare the delay between SR-WSN and a typical broadcast system. This is mainly because packets are lost. The lost packets cannot be included in the analysis. That is why it was decided to study performability, which includes performance and reliability together.

In fact, chapter 4 compares the performability of SR-WSN and a classical broadcast protocol by the means of a Markovian Reward Model.

The main contribution is

i. Generating a Markovian Reward Model for our system and showed that our system works better then broadcast algorithm (fig. 4.6)

5.4 PROPOSED FUTURE WORK

Wireless sensor network usually put the sensors to sleep for a large portion of their operational time in order to save energy. This issue was not analyzed in this thesis. We believe that analyzing the impact of ON/OFF pattern of sensors on the Markovian Reward Model and the performability in particular could be possible directions for future work.

BIBLIOGRAPHY

[1] A. Chandrakasan, W. R. Heinzelman, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," Proc. HICSS,Vol. 8, pp. 8020, 2000.

[2] Bhaskar Krishnamachari, Deborah Estrin, Stephen Wicker "Modelling Data-Centric Routing in Wireless SensorNetworks" IEEE INFOCOM 2002.

[3] CHEE-YEE CHONG, MEMBER, IEEE AND SRIKANTA P. KUMAR, SENIOR MEMBER, IEEE "10 emerging technologies that will change the world," Technol.Rev., vol. 106, no. 1, pp. 33–49, Feb. 2003.

[4] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networks (MobiCOM 2000), August 2000, Boston, Massachusetts.

[5] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, F. Silva, "Directed Diffusion for Wireless Sensor Networking", IEEE/ACM Trans. Networking, vol. 11, no. 1, pp. 2-16, Feb. 2003.

[6] De Souza e Silva,Gail(1992)-'Performability analysis of computer systems: from model specification to solution',Perf.Eval.14,157-196.

[7] E.H. Callaway, Jr. "Wireless sensor Networks, Architectures and Protocols", AUERBACH 2003.

[8] Frederick Abou Jawad & Eric Johnsen, " Performability: the vital evaluation method for degradable systems and its most commonly used modelling method, Markov Reward Modelling".

[9] Group of Network Simulator (NS2), http://www.isi.edu/nsnam/ns/.

[10] G. Chung & M. Claypool "Worcester Polytechnic Institute", http://nile.wpi.edu/NS/.

[11] IEEE 802.15 Working Group for WPAN. [Online] Available: http:// grouper.ieee.org/groups/802/15/.

[12] IEEE Standards Association; http://standards.ieee.org/.

[13] Jan Blumenthal, Matthias Handy, Frank Golatowski, Marc Haase, Dirk Timmermann, "Wireless Sensor Networks - New Challenges in Software Engineering", August 2003, University of Rostock.

[14] J. Corella, "Tactical automated security system (TASS): Air force expeditionary security," presented at the SPIE Conf. Unattended.

[15] J. F. Meyer(1992)-'Performability: a retrospective and some pointers to the future', Performance Evaluation 14 (1992) 139-156.

[16] J. Kulik , W. Heinzelman , and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," Wireless Networks, March 2002.

[17] Maalouf & Aakvaag (Working paper), Synchronization and Performance Issues in High Density Wireless Sensors Networks.

[18] N. Bulusu, D. Estrin, L. Girod and J. Heidemann, "Scalable Coordination for wireless sensor networks: Self-Configuring Localization Systems," In Proceedings of the Sixth International Symposium on Communication Theory and Applications (ISCTA 2001), Ambleside, Lake District,UK, July 2001.

[19] R. T. Lacoss, "Distributed mixed sensor aircraft tracking," presented at the Amer. Control Conf., Minneapolis, MN, 1987.

[20] S. Kumar and D. Shepherd, "SensIT: Sensor information technology for the warfighter," in Proc. 4th Int. Conf. on Information Fusion, 2001, pp. TuC1-3–TuC1-9.

[21] V. Raghunathan, C. Schurgers, Park.S, and M.B. Srivastava, "Energy-aware wireless microsensor networks," IEEE Signal Processing Magazine, Volume: 19 Issue: 2, March 2002.

[22] J.W. Gardner, V. K Varadan, and O. O. Awadelkarim, Microsensors, MEMS and Smart Devices. New York: Wiley, 2001.

[23] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in the Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99), Seattle, WA, August 1999.

[24] William A. Arbaugh, Narendar Shankar "Your 802.11 Wireless Network Has no Clothes" Y.C. Justing Wan. 2001.

[25] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," In Proceedings of the fifth annual ACM/IEEE international conference on Mobile computing and networking, August 1999.

APPENDIX A

Simulation Code

```
# variables which control the number of nodes and how they're grouped
# (see topology creation code below)
set val(chan)           Channel/WirelessChannel    ;#Channel Type
set val(prop)           Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)          Phy/WirelessPhy            ;# network interface type
set val(nn)            39                         ;# number of node
set val(mac)            Mac/802_11                 ;# MAC type
#set val(mac)           Mac                        ;# MAC type
#set val(mac)          Mac/Simple
set val(ifq)            Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)             LL                         ;# link layer type
set val(ant)            Antenna/OmniAntenna        ;# antenna model
set val(ifqlen)         500                        ;# max packet in ifq
#set val(rp)             DSDV
#set val(rp)             DSR
set val(rp)            AODV
# size of the topography
set val(x)              500
set val(y)              500


set ns_ [new Simulator]

set f [open 1MSThesis_Fadi.tr w]
$ns_ use-newtrace
$ns_ trace-all $f
set nf [open 1MSThesis_Fadi.nam w]
$ns_ namtrace-all-wireless $nf $val(x) $val(y)



# set up topography object
set topo       [new Topography]

$topo load_flatgrid $val(x) $val(y)

#
# Create God
#
create-god $val(nn)
set chan_1_ [new $val(chan)]

$ns_ node-config -adhocRouting $val(rp) \
               -addressingType hierachical \
                -llType $val(ll) \
                -macType $val(mac) \
                -ifqType $val(ifq) \
                -ifqLen $val(ifqlen) \
                -antType $val(ant) \
                -propType $val(prop) \
                -phyType $val(netif) \
                -topoInstance $topo \
                -agentTrace OFF \
                -routerTrace ON \
                -macTrace OFF \
                -movementTrace OFF \
               -energyModel "EnergyModel" \
               -rxPower 10.0 \
               -txPower 10.0 \
               -initialEnergy 1200.0\
                -channel $chan_1_

$ns_ node-config -reset

#source ./1MS_thesis_fadi.scn
```

```
set node_(39) [$ns_ node]
## node_(39) at 467.354950,1298.646240
$node_(39) set X_ 1000.0
$node_(39) set Y_ 900.0
$node_(39) set Z_ 0.0
$node_(39) color "black"
$ns_ initial_node_pos $node_(39) 45.000
set node_(38) [$ns_ node]
## node_(38) at 888.029602,898.167114
$node_(38) set X_ 1800.0
$node_(38) set Y_ 830.0
$node_(38) set Z_ 0.0
$node_(38) color "black"
$ns_ initial_node_pos $node_(38) 45.000
set node_(37) [$ns_ node]
## node_(37) at 779.895203,895.914307
$node_(37) set X_ 1600.0
$node_(37) set Y_ 830.0
$node_(37) set Z_ 0.0
$node_(37) color "black"
$ns_ initial_node_pos $node_(37) 45.000
set node_(36) [$ns_ node]
## node_(36) at 676.266418,895.914307
$node_(36) set X_ 1400.0
$node_(36) set Y_ 830.0
$node_(36) set Z_ 0.0
$node_(36) color "black"
$ns_ initial_node_pos $node_(36) 45.000
set node_(35) [$ns_ node]
## node_(35) at 568.131958,889.155884
$node_(35) set X_ 1100.0
$node_(35) set Y_ 830.0
$node_(35) set Z_ 0.0
$node_(35) color "black"
$ns_ initial_node_pos $node_(35) 45.000
set node_(34) [$ns_ node]
## node_(34) at 459.997620,889.155884
$node_(34) set X_ 900.0
$node_(34) set Y_ 830.0
$node_(34) set Z_ 0.0
$node_(34) color "black"
$ns_ initial_node_pos $node_(34) 45.000
set node_(33) [$ns_ node]
## node_(33) at 360.874420,889.155884
$node_(33) set X_ 700.0
$node_(33) set Y_ 830.0
$node_(33) set Z_ 0.0
$node_(33) color "black"
$ns_ initial_node_pos $node_(33) 45.000
set node_(32) [$ns_ node]
## node_(32) at 254.992798,889.155884
$node_(32) set X_ 500.0
$node_(32) set Y_ 830.0
$node_(32) set Z_ 0.0
$node_(32) color "black"
$ns_ initial_node_pos $node_(32) 45.000
set node_(31) [$ns_ node]
## node_(31) at 151.364014,884.650330
$node_(31) set X_ 300.0
$node_(31) set Y_ 830.0
$node_(31) set Z_ 0.0
$node_(31) color "black"
$ns_ initial_node_pos $node_(31) 45.000
           .
           .
           .
set node_(4) [$ns_ node]
## node_(4) at 680.771973,141.226318
$node_(4) set X_ 720.0
$node_(4) set Y_ 140.0
```

```
$node_(4) set Z_ 0.0
$node_(4) color "black"
$ns_ initial_node_pos $node_(4) 45.000
set node_(3) [$ns_ node]
## node_(3) at 439.722382,141.226318
$node_(3) set X_ 480.0
$node_(3) set Y_ 140.0
$node_(3) set Z_ 0.0
$node_(3) color "black"
$ns_ initial_node_pos $node_(3) 45.000
set node_(2) [$ns_ node]
## node_(2) at 209.936798,141.226318
$node_(2) set X_ 209.0
$node_(2) set Y_ 140.0
$node_(2) set Z_ 0.0
$node_(2) color "black"
$ns_ initial_node_pos $node_(2) 45.000
set node_(1) [$ns_ node]
## node_(1) at -31.112797,134.467926
$node_(1) set X_ -31.0
$node_(1) set Y_ 140.0
$node_(1) set Z_ 0.0
$node_(1) color "black"
$ns_ initial_node_pos $node_(1) 45.000


$ns_ color &node_(12) blue

# Setup traffic flow between nodes

Agent/TCP set packetSize_ 20
set source3 [new Agent/TCP]
set sink3 [new Agent/TCPSink]
$ns_ attach-agent $node_(7) $source3
$ns_ attach-agent $node_(39) $sink3
$sink3 listen
$ns_ connect $source3 $sink3

set tcp [new Agent/TCP]
set sink [new Agent/TCPSink]
$ns_ attach-agent $node_(5) $tcp
$ns_ attach-agent $node_(39) $sink
$sink listen
$ns_ connect $tcp $sink

set source2 [new Agent/TCP]
set sink2 [new Agent/TCPSink]
$ns_ attach-agent $node_(6) $source2
$ns_ attach-agent $node_(39) $sink2
$sink2 listen
$ns_ connect $source2 $sink2

set conn1 [new Application/FTP]
$ftp attach-agent $tcp
$ns_ at 0.01 "$ftp start"
$ns_ at 0.01 "$node_(5) add-mark m1 brown"
$ns_ at 0.01 "$node_(39) add-mark m2 brown"
set conn2 [new Application/FTP]
$conn2 attach-agent $source2
$ns_ at 0.23 "$conn2 start"
$ns_ at 0.23 "$node_(6) add-mark m1 brown"
set conn3 [new Application/FTP]
$conn3 attach-agent $source3
$ns_ at 0.01 "$conn3 start"
$ns_ at 0.01 "$node_(7) add-mark m1 brown"

#$ns_ rtmodel-at 5.0 down $node_(5) $node_(16)
#$ns rtmodel-at 2.0 up $n(1) $n(2)

#$ns_ node-down at 5.0 &node_(16)
#$ns_ rtmodel-at 3.0 down $node_(1)
```

```
#$ns_ at 5.0 "$node_(16) node-down"
#$ns_ at 5.0 "$ns_ trace-annotate \"(at 3.0) node down: 1\""
$ns_ at 5.0 "$node_(1) add-mark m3 orange"
#$ns_ initialEnergy-at 5.0 &node_(16)

for {set i 1} {$i < $val(nn) } {incr i} {
 $ns_ at 5.0 "$node_($i) reset";
}

$ns_ at 20.0 "stop"
$ns_ at 20.01 "puts \"NS EXITING...\" ; $ns_ halt"
proc stop {} {
   global ns_ f nf val
    $ns_ flush-trace
    close $f
    close $nf

#        puts "running nam..."

exec nam 1MSThesis_Fadi.nam &
        exit 0
}

$ns_ run
```

APPENDIX B

Part of the Trace file (one source node)

Hop Source	Hop Destination	Node Energy	Packet ID	Time
Hs 34	Hd 23	Ne 1183.986862	289	3.695174653
Hs 23	Hd 12	Ne 1175.594627	289	3.866661244
Hs 23	Hd 23	Ne 1175.594627	289	3.866661244
Hs 12	Hd 5	Ne 1175.773939	289	3.868999714
Hs 12	Hd 12	Ne 1175.773939	289	3.868999714
Hs 5	Hd 5	Ne 1179.007043	289	3.874292760
Hs 5	Hd 0	Ne 1179.007043	289	3.874292760
Hs 34	Hd 23	Ne 1183.737742	292	3.743359559
Hs 23	Hd 12	Ne 1175.549507	292	3.871783600
Hs 23	Hd 23	Ne 1175.549507	292	3.871783600
Hs 12	Hd 12	Ne 1175.622861	292	3.886202495
Hs 12	Hd 5	Ne 1175.622861	292	3.886202495
Hs 5	Hd 5	Ne 1178.883683	292	3.888700849
Hs 5	Hd 0	Ne 1178.883683	292	3.888700849
Hs 34	Hd 23	Ne 1183.737742	293	3.743359559
Hs 23	Hd 23	Ne 1175.260511	293	3.919223826
Hs 23	Hd 12	Ne 1175.260511	293	3.919223826
Hs 12	Hd 12	Ne 1175.439821	293	3.921702296
Hs 12	Hd 5	Ne 1175.439821	293	3.921702296
Hs 5	Hd 5	Ne 1178.711043	293	3.931713498
Hs 5	Hd 0	Ne 1178.711043	293	3.931713498

⋮

Hs 34	Hd 33	Ne 1140.962881	930	11.270043242
Hs 33	Hd 33	Ne 1158.007793	930	11.286295371
Hs 33	Hd 22	Ne 1158.007793	930	11.286295371
Hs 22	Hd 21	Ne 1123.901961	930	11.352014999
Hs 22	Hd 22	Ne 1123.901961	930	11.352014999
Hs 21	Hd 21	Ne 1165.976753	930	11.378370732
Hs 21	Hd 15	Ne 1165.976753	930	11.378370732
Hs 15	Hd 2	Ne 1186.049428	930	11.380969086
Hs 15	Hd 15	Ne 1186.049428	930	11.380969086
Hs 2	Hd 2	Ne 1187.879343	930	11.390892870
Hs 2	Hd 3	Ne 1187.879343	930	11.390892870
Hs 3	Hd 3	Ne 1151.147171	930	11.414993624
Hs 3	Hd 4	Ne 1151.147171	930	11.414993624
Hs 4	Hd 4	Ne 1132.779657	930	11.422508957
Hs 4	Hd 0	Ne 1132.779657	930	11.422508957

Hs 34	Hd 33	Ne 1140.962881	931	11.270043242
Hs 33	Hd 22	Ne 1157.977415	931	11.290114571
Hs 33	Hd 33	Ne 1157.977415	931	11.290114571
Hs 22	Hd 21	Ne 1123.879401	931	11.354674091
Hs 22	Hd 22	Ne 1123.879401	931	11.354674091
Hs 21	Hd 21	Ne 1165.909073	931	11.386036163
Hs 21	Hd 15	Ne 1165.909073	931	11.386036163
Hs 15	Hd 15	Ne 1185.981748	931	11.388454517
Hs 15	Hd 2	Ne 1185.981748	931	11.388454517
Hs 2	Hd 2	Ne 1187.856783	931	11.393592009
Hs 2	Hd 3	Ne 1187.856783	931	11.393592009
Hs 3	Hd 3	Ne 1151.124611	931	11.417472290
Hs 3	Hd 4	Ne 1151.124611	931	11.417472290
Hs 4	Hd 4	Ne 1132.637737	931	11.438253320
Hs 4	Hd 0	Ne 1132.637737	931	11.438253320
Hs 34	Hd 33	Ne 1140.962881	932	11.270043242
Hs 33	Hd 33	Ne 1157.964853	932	11.2529537[illegible]
Hs 33	Hd 22	Ne 1157.964855	932	11.2529537[illegible]
Hs 34	Hd 33	Ne 1140.962831	943	11.270049242
Hs 33	Hd 33	Ne 1157.932295	933	11.28665[illegible]
Hs 33	Hd 22	Ne 1157.932295	933	11.28665[illegible]
Hs 34	Hd 35	Ne 1139.869268	939	11.498631877
Hs 35	Hd 35	Ne 1139.365930	939	11.501370277
Hs 35	Hd 24	Ne 1139.365930	939	11.501370277
Hs 24	Hd 23	Ne 1122.948004	939	11.503868579
Hs 24	Hd 24	Ne 1122.948004	939	11.503868579
Hs 23	Hd 23	Ne 1114.771383	939	11.509386879
Hs 23	Hd 12	Ne 1114.771383	939	11.509386879
Hs 12	Hd 12	Ne 1115.252880	939	11.512005349
Hs 12	Hd 5	Ne 1115.252880	939	11.512005349
Hs 5	Hd 5	Ne 1128.619580	939	11.515484595
Hs 5	Hd 0	Ne 1128.619580	939	11.515484595