Spring 2003

Detecting malfunction in wireless sensor networks

Pablito C. Lake
New Jersey Institute of Technology

ABSTRACT

DETECTING MALFUNCTION IN WIRELESS SENSOR NETWORKS

by
Pablito C. Lake

The objective of this thesis is to detect malfunctioning sensors in wireless sensor networks. The ability to detect abnormality is critical to the security of any sensor network. However, the ability to detect a faulty wireless sensor is not trivial. Controlled repeatable experiments are difficult in wireless channels. A Redhat Linux 7.0 Wireless Emulation Dynamic Switch software was used to solve this problem.

Six nodes were configured with a node acting as a base station. The nodes were all part of a cell. This means that every node could communicate with all other nodes. A client-server program simulated the background traffic. Another program simulated a faulty node. A node was isolated as the faulty node while all other nodes were good. The experiment ran for several hours and the data was captured with tcpdump. The data was analyzed to conclusions based on a statistical comparison of good node versus bad node.

The statistical delay on the good node was an average of 0.69 ms while the standard deviation was 0.49. This was much better than the delay on the bad node that was 0.225192 s with a standard deviation of 0.89. This huge difference in the delay indicated that the faulty node was detected statistically. A threshold value of 1 ms was chosen. The good node was within this value about 98 % of the time. The bad node on the other hand was far out of this range and was definitely detected. The channel utilization data provided the same conclusion.

DETECTING MALFUNCTION IN WIRELESS SENSOR NETWORKS

by
Pablito C. Lake

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Engineering

Department of Electrical and Computer Engineering

May 2003

APPROVAL PAGE

DETECTING MALFUNCTION IN WIRELESS SENSOR NETWORKS

Pablito C. Lake

Dr. Constantine Manikopoulos, Thesis Advisor — Date
Associate Professor, Department Electrical and Computer Engineering, NJIT

Dr. George Antoniou, Committee Member — Date
Professor, Department of Computer Science, Montclair State University

Dr. Bin He, Committee Member — Date
Senior scientist, XPRT Solutions Inc.

BIOGRAPHICAL SKETCH

Author:	Pablito C. Lake
Degree:	Master of Science
Date:	May 2003

Undergraduate and Graduate Education:

- Master of Science in Computer Engineering,
  New Jersey Institute of Technology, Newark, NJ 2003
- Bachelor of Science in Computer Engineering
  New Jersey Institute of Technology, Newark, NJ 2002
- Associate of Science in Engineering,
  Essex County College, Newark, NJ, 1999

Major:	Computer Engineering

To my supportive wife, Valerie Lake and mother, Orfelina Lake

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to Dr. Constantine Manikopoulos, who mentored and supervised me with constant support and encouragement. His guidance, patience and understanding provided the roadmap towards the completion of this thesis. I also give special thanks to Dr. Sotirios Ziavras who provided invaluable assistance throughout my entire undergraduate and graduate student life.

Special fellow graduate students in the Network Intrusion Detection Laboratory, Mr. Olufemi Tairu, Mr. Sachin Aora, Mr. Jun Li and Ms. Ling Li, are deserving of recognition for their support.

TABLE OF CONTENTS

Chapter Page

1 INTRODUCTION 1

1.1 Objective 1

1.2 Background Information 2

2 INTRUSION DETECTION SYSTEM (IDS) 6

2.1 Types and Analysis of IDS 6

2.2 IDS Constraints in Wired IDS versus Wireless Mediums 22

2.3 Summary 24

3 SENSOR NETWORKS 25

3.1 Types of Sensor Networks 25

3.2 Sensor Networks IDS constraints and Proposed Solutions 42

3.3 Neighborhood Watch IDS 44

3.4 Summary 46

4 IMPLEMENTATION 48

4.1 Dynamic Switch Emulation Tool 48

4.1.1 Architecture and Leaky Bucket Model 48

4.1.2 Dynamic Switch Constraints 53

4.2 Simulation Results and Analysis 53

4.3 Summary 56

5 CONCLUSIONS 58

REFERENCES 60

LIST OF TABLES

Table **Page**

2.1 Comparison of a Smartdust sensor with a workstation....................................23

LIST OF FIGURES

Figure **Page**

4.1.1.1 (a) Dynamic Switch Topology 49

4.1.1.1(b) Equivalent Wireless Network 49

4.1.1.2 Two-state Markov Chain for Packet Drop Process 51

4.1.1.3 Leaky Bucket Token Model 52

4.2.1 Wireless Test Bed 53

4.2.2 Statistical Delay on Good Node 54

4.2.3 Statistical Delay on Faulty Node 55

4.2.4 Channel Utilization on Good Node 56

4.2.5 Channel Utilization on Faulty Node 56

CHAPTER 1

INTRODUCTION

1.1 Objective

The objective of this thesis is to detect malfunctioning sensors in wireless sensor networks. At this point in the discussion it serves well to define the term "malfunction". According to the Oxford Dictionary the term malfunction means "failure to function normally". Therefore, there are many situations that determine whether a sensor is malfunctioning. The following bullets give some examples of a malfunctioning sensor:

- Sensor is hacked into by an unauthorized person
- Sensor reporting erroneous data because of hardware or software problem
- Sensor reporting erroneous data due to extreme environmental conditions
- Sensor exhausting its energy supply
- Sensor physically damaged by an individual

These instances cover a wide variety of situations responsible for the abnormal behavior of sensors. The ability to detect the above circumstances is critical to the security of any sensor network. Although this is true, the ability to detect a faulty wireless sensor is not trivial since the possibilities for abnormality are endless and the wireless medium's performance is difficult to predict and or control.

The wireless channels properties include low bandwidth, high delay, high bit-error rates, power and distance tradeoffs, and high packet drop rates. The erratic behavior of these characteristics makes it difficult to conduct controlled research and experimentation towards predicting, with reasonable accuracy, the malfunctioning of sensors in a wireless sensor network. Thus, this thesis takes advantage of a Linux software switch to study the malfunctioning of sensors in a simulated wireless environment where some parameters are controlled.

1.2 Background Information

So, what is a sensor? A sensor is a device that communicates a physical change in the environment such as radiation or temperature. Sensors are by no means new. However, they are creating a buzz commercially and in research communities. This increased interest spurred out of two major technological revolutions. The first was the connection of sensors to computer systems and the second was the emergence of small, inexpensive and highly reliable micro electronic and mechanical systems (MEMS) [8]. The idea of the dynamic topology of wireless ad-hoc networks is also a driving force behind the great deal of commercial and research interest. "Furthermore, integration of inexpensive, power efficient and reliable sensors in nodes of wireless ad-hoc networks, with significant computational and communication resources, opens new research and engineering vistas" [8].

A typical sensor, like the TinyOS Sensor Mote, is equipped with a 4 MHz microprocessor with 512 bytes of RAM and 8 KB of code space, a 917 MHz RFM radio running at 10 kbps, and 32 KB of EEPROM [6]. The radio hardware has a single channel with half duplex communication. An AA battery pack powers this device. The sensor's effective lifespan is proportional to its power supply. The energy consumed while transmitting and receiving bits of information and computing and processing data depletes the power supply. The average amount of energy needed to send or receive information a bit of is about 4000 nanojoules. These characteristics of a typical senor show that a sensor is quite limited in energy, radio range and memory. Such limited resources are attributed to the low cost of sensors. Wireless sensor networks, a special class of ad-hoc networks, efficiently use high node density with keen attention to energy consumption [9]. A sensor in a high-density wireless sensor network operates for about five to ten years.

The limitations on a wireless sensor network include enough battery power to supply all sensor nodes, sufficient memory to store cryptographic keys, and means for communicating with outside networks. Sensor applications have limited local exchange and data processing that fall into three categories:

1. Node to base station communication, e.g. sensor readings
2. Base station to node communication, e.g. specific requests
3. Base station to all nodes e.g. queries or reprogramming of the entire network.

The security goal is to address these communication patterns. Generally, the sensor networks may be deployed in non-trusted locations. The integrity of the each node can be realized through dedicated secure micro controllers. However, individual sensors still cannot be trusted! Wireless communication is fundamentally untrustworthy! Any adversary can eavesdrop on the traffic, and inject new messages or replay and change old messages because of the traffic's broadcast nature. Hence, one cannot place any trust assumptions on the communication infrastructure, except that messages are delivered to the destination with non-zero probability.

The security of sensors is categorized into four main requirements – (1) Data Confidentiality, (2) Data Authentication, (3) Data Integrity and (4) Data Freshness [10].

Data Confidentiality is concerned with data leaking from a sensor to an unintended recipient. **Data Authentication** allows the recipient to verify that the data did indeed come from the claimed sender [10]. **Data Integrity** ensures the receiver that the data was not tampered with on its route. **Data Freshness** means that the data is recently sent. The above requirements of a sensor make an effort to provide various levels of security to a wireless sensor network. They are taken into consideration when one is designing an Intrusion Detection System (IDS). The methods of dealing with the four requirements for sensor security are explained in the following paragraphs.

Data Confidentiality – A sensor network should not leak sensor readings to the neighboring networks. In many applications the nodes communicate highly sensitive data. The standard solution to keep sensitive data secret is to encrypt the data with a secret key that only the intended receivers possess, hence achieving confidentiality. If necessary, one can use initially set up secure channels between nodes and base stations to bootstrap other secure channels given the observed communication patterns.

Data Authentication – Message authentication is of paramount importance for many applications in sensor networks. Within the building sensor network, authentication is necessary for many administrative tasks (e.g. network reprogramming or controlling sensor node duty cycle). At the same time, an adversary can easily inject messages, so the receiver needs to make sure that the data used in any decision-making process originated from the correct source. Informally, data authentication allows the receiver to verify that the claimed sender really sent the data. In the two-party communication case, data authentication can be achieved through a purely symmetric mechanism: The sender and the receiver share a secret key to compute a message authentication code (MAC) of all communicated data. When a message with a correct MAC arrives, the receiver knows that the sender must have sent it. This style of authentication cannot be applied to a broadcast setting without placing much stronger trust assumptions on the network nodes. It is insecure for one sender to send authentic data to mutually non-trusted receivers using a symmetric MAC: Any one of the receivers knows the MAC key, and hence could impersonate the sender and forge messages to other receivers. Hence, one needs an asymmetric mechanism to achieve authenticated broadcast.

Data Integrity – In communication, data integrity ensures the receiver that an adversary does not alter the received data in transit [10]. This can be achieved through data authentication, which is a stronger property. Therefore, data integrity is a subset of data authentication.

Data Freshness – Given that all sensor networks stream some forms of time varying measurements, it is not enough to guarantee confidentiality and authentication; one also must ensure each message is *fres*h. Informally, data freshness implies that the data is recent, and it ensures that no adversary replayed old messages. There are two types of freshness: weak freshness, which provides partial message ordering, but carries no delay information, and strong freshness, which provides a total order on a request-response pair, and allows for delay estimation. Weak freshness is required by sensor measurements, while strong freshness is useful for time synchronization within the network.

CHAPTER 2

INTRUSION DECTECTION SYSTEM (IDS)

2.1 Types and Analysis of IDS

Intrusion Detection Systems (IDSs) are software or hardware systems that automate the process of monitoring the events occurring in a computer system or network, analyzing them for signs of security problems [1]. As network attacks have increased in number and severity over the past few years, Intrusion Detection Systems have become a necessary addition to the security infrastructure of most organizations. This Chapter is intended as a primer in intrusion detection, developed for those who need to understand what security goals intrusion detection mechanisms serve, how to select and configure intrusion detection systems for their specific system and network environments, how to manage the output of intrusion detection systems, and how to integrate intrusion detection functions with the rest of the organizational security infrastructure. There are many publicly available information sources for the reader who requires specialized or more detailed advice on specific intrusion detection issues.

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of *intrusions,* defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or network. Intrusions are caused by attackers accessing the systems from the Internet, authorized users of the systems who attempt to gain additional privileges for which they are not authorized, and authorized users who misuse the privileges given to them. Intrusion Detection Systems (IDSs) are software or hardware that automate this monitoring and analysis process.

Intrusion detection allows organizations to protect their systems from the threats that come with increasing network connectivity and reliance on information systems. Given the level and nature of modern network security threats, the question for security professionals should not be whether to use intrusion detection, but which intrusion detection features and capabilities to use. IDSs have gained acceptance as a necessary addition to every organization's security infrastructure. Despite the documented contributions intrusion detection technologies make to system security, in many organizations one must still justify the acquisition of IDSs. There are several compelling reasons to acquire and use IDSs [1]:

- To prevent problem behaviors by increasing the perceived risk of discovery and punishment for those who would attack or otherwise abuse the system,
- To detect attacks and other security violations that are not prevented by other security measures
- To detect and deal with the preambles to attacks (commonly experienced as network probes and other "doorknob rattling" activities)
- To document the existing threat to an organization
- To act as quality control for security design and administration, especially of large and complex enterprises
- To provide useful information about intrusions that do take place, allowing improved diagnosis, recovery, and correction of causative factors

A fundamental goal of computer security management is to affect the behavior of individual users in a way that protects information systems from security problems. Intrusion Detection Systems help organizations accomplish this goal by increasing the perceived risk of discovery and punishment of attackers. This serves as a significant deterrent to those who would violate security policy.

Attackers, using widely publicized techniques, can gain unauthorized access to many, if not most systems, especially those connected to public networks. This often happens when known vulnerabilities in the systems are not corrected. Although vendors and administrators are encouraged to address vulnerabilities lest they enable attacks, there are many situations in which this is not possible [1]:

- In many legacy systems, the operating systems cannot be patched or updated.
- Even in systems in which patches can be applied, administrators sometimes have neither sufficient time nor resource to track and install all the necessary patches. This is a common problem, especially in environments that include a large number of hosts or a wide range of different hardware or software environments.
- Users can have compelling operational requirements for network services and protocols that are known to be vulnerable to attack.
- Both users and administrators make errors in configuring and using systems.
- In configuring system access control mechanisms to reflect an organization's procedural computer use policy, discrepancies almost always occur. These disparities allow legitimate users to perform actions that are ill advised or that overstep their authorization.

In an ideal world, commercial software vendors would minimize vulnerabilities in their products, and user organizations would correct all reported vulnerabilities quickly and reliably. However, in the real world, this seldom happens thanks to our reliance on commercial software where new flaws and vulnerabilities are discovered on a daily basis. Given this state of affairs, intrusion detection can represent an excellent approach to protecting a system. An IDS can detect when an attacker has penetrated a system by exploiting an uncorrected or uncorrectable flaw. Furthermore, it can serve an important function in system protection, by bringing the fact that the system has been attacked to the attention of the administrators who can contain and recover any damage that results. This is far preferable to simply ignoring network security threats where one allows the attackers continued access to systems and the information on them.

When adversaries attack a system, they typically do so in predictable stages. The first stage of an attack is usually probing or examining a system or network, searching for an optimal point of entry. In systems with no IDS, the attacker is free to thoroughly examine the system with no risk of discovery or retribution. Given this unfettered access, a determined attacker will eventually find a vulnerability in such a network and exploit it to gain entry to various systems. The same network with an IDS monitoring its operations presents a much more formidable challenge to that attacker. Although the attacker may probe the network for weaknesses, the IDS will observe the probes, will identify them as suspicious, may actively block the attacker's access to the target system, and will alert security personnel who can then take appropriate actions to block subsequent access by the attacker. Even the presence of a reaction to the attacker's probing of the network will elevate the level of risk the attacker perceives, discouraging further attempts to target the network.

When you are drawing up a budget for network security, it often helps to substantiate claims that the network is likely to be attacked or is even currently under attack. Furthermore, understanding the frequency and characteristics of attacks allows one to understand what security measures are appropriate to protect the network against those attacks. IDSs verify, itemize, and characterize the threat from both outside and inside an organization's network by assisting in making sound decisions regarding the allocation of computer security resources. Using IDSs in this manner is important, as many people mistakenly deny that anyone (outsider or insider) would be interested in breaking into their networks. Furthermore, the information that IDSs give you regarding the source and nature of attacks allows one to make decisions regarding security strategy driven by demonstrated need, not guesswork or folklore.

When IDSs run over a period of time, patterns of system usage and detected problems can become apparent. These can highlight flaws in the design and management of security for the system, in a fashion that supports security management correcting those deficiencies before they cause an incident. Even when IDSs are not able to block attacks, they can still collect relevant, detailed, and trustworthy information about the attack that supports incident handling and recovery efforts. Furthermore, this information can, under certain circumstances, enable and support criminal or civil legal remedies. Ultimately, such information can identify problem areas in the organization's security configuration or policy.

There are several types of IDSs available today, characterized by different monitoring and analysis approaches [1]. Each approach has distinct advantages and disadvantages. Furthermore, all approaches can be described in terms of a generic process model for IDSs. Many IDSs can be described in terms of three fundamental functional components:

- **Information Sources** – the different sources of event information used to determine whether an intrusion has taken place. These sources can be drawn from different levels of the system, with network, host, and application monitoring most common.
- **Analysis** – the part of intrusion detection systems that actually organizes and makes sense of the events derived from the information sources, deciding when those events indicate that intrusions are occurring or have already taken place. The most common analysis approaches are *misuse detection* and *anomaly detectio*n.
- **Response** – the set of actions that the system takes once it detects intrusions. These are typically grouped into active and passive measures, with active measures involving some automated intervention on the part of the system, and passive measures involving reporting IDS findings to humans, who are then expected to take action based on those reports.

There are several design approaches used in Intrusion Detection. These drive the features provided by a specific IDS and determine the detection capabilities for that system. For those who must evaluate different IDS candidates for a given system environment, these approaches can help them determine what goals are best addressed by each IDS. The architecture of an IDS refers to how the functional components of the IDS are arranged with respect to each other. The primary architectural components are the Host, the system on which the IDS software runs, and the Target, the system that the IDS is monitoring for problems.

In early days of IDSs, most IDSs ran on the systems they protected. This was due to the fact that most systems were mainframe systems, and the cost of computers made a separate IDS system a costly extravagance. This presented a problem from a security point of view, as any attacker that successfully attacked the target system could simply disable the IDS as an integral portion of the attack.

With the advent of workstations and personal computers, most IDS architects moved towards running the IDS control and analysis systems on a separate system, hence separating the IDS host and target systems. This improved the security of the IDS as this made it much easier to hide the existence of the IDS from attackers. Although there are many goals associated with security mechanisms in general, there are two overarching goals usually stated for intrusion detection systems – Accountability and Response.

Accountability is the capability to link a given activity or event back to the party responsible for initiating it [1]. This is essential in cases where one wishes to bring criminal charges against an attacker. The goal statement associated with accountability is: *"I can deal with security attacks that occur on my systems as long as I know who did it and where to find them."* Accountability is difficult in TCP/IP networks, where the protocols allow attackers to forge the identity of source addresses or other source identifiers. It is also extremely difficult to enforce accountability in any system that employs weak identification and authentication mechanisms.

Response is the capability to recognize a given activity or event as an attack and then taking action to block or otherwise affect its ultimate goal [1]. The goal statement associated with response is *"I don't care who attacks my system as long as I can recognize that the attack is taking place and block it."* Note that the requirements of detection are quite different for response than for accountability. Control Strategy describes how the elements of an IDS are controlled, and furthermore, how the input and output of the IDS are managed. They are outlined below:

Centralized – Under centralized control strategies, all monitoring, detection and reporting are controlled directly from a central location

Interval-Based (Batch Mode) – In interval-based IDSs, the information that flows from monitoring points to analysis engines is not continuous. In effect, the information is handled in a fashion similar to "store and forward" communications schemes. Many early host-based IDSs used this timing scheme, as they relied on operating system audit trails, which were generated as files. Interval-based IDSs are precluded from performing active responses.

Real-Time (Continuous) – Real-time IDSs operate on continuous information feeds from information sources. This is the predominant timing scheme for network-based IDSs, which gather information from network traffic streams. Here, the term "real-time" is used as it is used in process control situations. This means that detection performed by a "real-time" IDS yields results quickly enough to allow the IDS to take action that affects the progress of the detected attack.

Information Sources – The most common way to classify IDSs is to group them by information source. Some IDSs analyze network packets, captured from network backbones or segments, to find attackers. Other IDSs analyze information sources generated by the operating system or application software for signs of intrusion.

The majority of commercial intrusion detection systems are **Network-based**. These IDSs detect attacks by capturing and analyzing network packets [1]. Listening on a network segment or switch, one network-based IDS can monitor the network traffic affecting multiple hosts that are connected to the network segment, thereby protecting those hosts. Network-based IDSs often consist of a set of single-purpose sensors or hosts placed at various points in a network. These units monitor network traffic, performing local analysis of that traffic and reporting attacks to a central management console. As the sensors are limited to running the IDS, they can be more easily secured against attack. Many of these sensors are designed to run in "stealth" mode, in order to make it more difficult for an attacker to determine their presence and location. The advantages of Network-Based IDSs are as follows:

- A few well-placed network-based IDSs can monitor a large network.
- The deployment of network-based IDSs has little impact upon an existing network. Network-based IDSs are usually passive devices that listen on a network wire without interfering with the normal operation of a network. Thus, it is usually easy to retrofit a network to include network-based IDSs with minimal effort.
- Network-based IDSs can be made very secure against attack and even made invisible to many attackers.

The disadvantages of Network-Based IDSs are [1]:

- Network-based IDSs may have difficulty processing all packets in a large or busy network and, therefore, may fail to recognize an attack launched during periods of high traffic. Some vendors are attempting to solve this problem by implementing IDSs completely in hardware, which is much faster. The need to analyze packets quickly also forces vendors to both detect fewer attacks and also detect attacks with as little computing resource as possible that can reduce detection effectiveness.
- Many of the advantages of network-based IDSs don't apply to more modern switch-based networks. Switches subdivide networks into many small segments (usually one fast Ethernet wire per host) and provide dedicated links between hosts serviced by the same switch. Most switches do not provide universal monitoring ports and this limits the monitoring range of a network-based IDS sensor to a single host. Even when switches provide such monitoring ports, often the single port cannot mirror all traffic traversing the switch.
- Network-based IDSs cannot analyze encrypted information. This problem is increasing as more organizations (and attackers) use virtual private networks.
- Most network-based IDSs cannot tell whether or not an attack was successful; they can only discern that an attack was initiated. This means that after a network-based IDS detects an attack, administrators must manually investigate each attacked host to determine whether it was indeed penetrated.
- Some network-based IDSs have problems dealing with network-based attacks that involve fragmenting packets. These malformed packets cause the IDSs to become unstable and crash.

Host-based IDSs operate on information collected from within an individual computer system. (Note that application-based IDSs are actually a subset of host-based IDSs.) This vantage point allows host-based IDSs to analyze activities with great reliability and precision, determining exactly which processes and users are involved in a particular attack on the operating system. Furthermore, unlike network-based IDSs, host-based IDSs can "see" the outcome of an attempted attack, as they can directly access and monitor the data files and system processes usually targeted by attacks. Host-based IDSs normally utilize information sources of two types, operating system audit trails, and system logs. Operating system (OS) audit trails are usually generated at the innermost (kernel) level of the operating system, and are therefore more detailed and better protected than system logs. However, system logs are much less obtuse and much smaller than audit trails, and are furthermore far easier to comprehend. Some host-based IDSs are designed to support a centralized IDS management and reporting infrastructure that can allow a single management console to track many hosts. Others generate messages in formats that are compatible with network management systems. The advantages are [1]:

- Host-based IDSs, with their ability to monitor events local to a host, can detect attacks that cannot be seen by a network-based IDS.
- Host-based IDSs can often operate in an environment in which network traffic is encrypted, when the host-based information sources are generated before data is encrypted and/or after the data is decrypted at the destination host
- Host-based IDSs are unaffected by switched networks.

- When Host-based IDSs operate on OS audit trails, they can help detect Trojan Horse or other attacks that involve software integrity breaches. These appear as inconsistencies in process execution.

The disadvantages are [1]:

- Host-based IDSs are harder to manage, as information must be configured and managed for every host monitored.
- Since at least the information sources (and sometimes part of the analysis engines) for host-based IDSs reside on the host targeted by attacks, the IDS may be attacked and disabled as part of the attack.
- Host-based IDSs are not well suited for detecting network scans or other such surveillance that targets an entire network, because the IDS only sees those network packets received by its host.
- Host-based IDSs can be disabled by certain denial-of-service attacks.
- When host-based IDSs use operating system audit trails as an information source, the amount of information can be immense, requiring additional local storage on the system.
- Host-based IDSs use the computing resources of the hosts they are monitoring, therefore inflicting a performance cost on the monitored systems.

Application-based IDSs are a special subset of host-based IDSs that analyze the events transpiring within a software application [1]. The most common information sources used by application-based IDSs are the application's transaction log files. The ability to interface with the application directly, with significant domain or application-specific knowledge included in the analysis engine, allows application-based IDSs to detect suspicious behavior due to authorized users exceeding their authorization. This is because such problems are more likely to appear in the interaction between the user, the data, and the application.

The advantages are [1]:

- Application-based IDSs can monitor the interaction between user and application, which often allows them to trace unauthorized activity to individual users.
- Application-based IDSs can often work in encrypted environments, since they interface with the application at transaction endpoints, where information is presented to users in unencrypted form.

The disadvantages are [1]:

- Application-based IDSs may be more vulnerable than host-based IDSs to attacks as the applications logs are not as well protected as the operating system audit trails used for host-based IDSs.
- As Application-based IDSs often monitor events at the user level of abstraction, they usually cannot detect Trojan Horse or other such software tampering attacks. Therefore, it is advisable to use an Application-based IDS in combination with Host-based and/or Network-based IDSs.

There are two primary approaches to analyzing events to detect attacks: **misuse detection and anomaly detection** [1]. Misuse detection, in which the analysis targets something known to be "bad", is the technique used by most commercial systems. Anomaly detection, in which the analysis looks for abnormal patterns of activity, has been, and continues to be, the subject of a great deal of research. Anomaly detection is used in limited form by a number of IDSs. There are strengths and weaknesses associated with each approach, and it appears that the most effective IDSs use mostly misuse detection methods with a smattering of anomaly detection components.

Misuse detectors analyze system activity, looking for events or sets of events that match a predefined pattern of events that describe a known attack [1]. As the patterns corresponding to known attacks are called *signature*s, misuse detection is sometimes called "signature-based detection." The most common form of misuse detection used in commercial products specifies each pattern of events corresponding to an attack as a separate signature. However, there are more sophisticated approaches to doing misuse detection (called "state-based" analysis techniques) that can leverage a single signature to detect groups of attacks.

The advantages are [1]:

- Misuse detectors are very effective at detecting attacks without generating an overwhelming number of false alarms.
- Misuse detectors can quickly and reliably diagnose the use of a specific attack tool or technique. This can help security managers prioritize corrective measures.
- Misuse detectors can allow system managers, regardless of their level of security expertise, to track security problems on their systems, initiating incident handling procedures.

The disadvantages are [1]:

- Misuse detectors can only detect those attacks they know about – therefore they must be constantly updated with signatures of new attacks.

- Many misuse detectors are designed to use tightly defined signatures that prevent them from detecting variants of common attacks. State-based misuse detectors can overcome this limitation, but are not commonly used in commercial IDSs.

Anomaly detectors identify abnormal unusual behavior (anomalies) on a host or network [1]. They function on the assumption that attacks are different from "normal" (legitimate) activity and can therefore be detected by systems that identify these differences. Anomaly detectors construct profiles representing normal behavior of users, hosts, or network connections. These profiles are constructed from historical data collected over a period of normal operation. The detectors then collect event data and use a variety of measures to determine when monitored activity deviates from the norm. The measures and techniques used in anomaly detection include [1]:

- Threshold detection, in which certain attributes of user and system behavior are expressed in terms of counts, with some level established as permissible. Such behavior attributes can include the number of files accessed by a user in a given period of time, the number of failed attempts to login to the system, the amount of CPU utilized by a process, etc. This level can be static or heuristic (i.e., designed to change with actual values observed over time).
- Statistical measures, both parametric, where the distribution of the profiled attributes is assumed to fit a particular pattern, and non-parametric, where the distribution of the profiled attributes is "learned" from a set of historical values, observed over time.

- Rule-based measures, which are similar to non-parametric statistical measures in that observed data defines acceptable usage patterns, but differs in that those patterns are specified as rules, not numeric quantities.
- Other measures include neural networks, genetic algorithms, and immune system models.

Only the first two measures are used in current commercial IDSs. Unfortunately, anomaly detectors and the IDSs based on them often produce a large number of false alarms, as normal patterns of user and system behavior can vary wildly. Despite this shortcoming, researchers assert that anomaly-based IDSs are able to detect new attack forms, unlike signature-based IDSs that rely on matching patterns of past attacks. Furthermore, some forms of anomaly detection produce output that can in turn be used as information sources for misuse detectors. For example, a threshold-based anomaly detector can generate a figure representing the "normal" number of files accessed by a particular user; the misuse detector can use this figure as part of a detection signature that says "if the number of files accessed by this user exceeds this "normal" figure by ten percent, trigger an alarm." Although some commercial IDSs include limited forms of anomaly detection, few, if any, rely solely on this technology. The anomaly detection that exists in commercial systems usually revolves around detecting network or port scanning. However, anomaly detection remains an active intrusion detection research area and may play a greater part in future IDSs.

The advantages are [1]:

- IDSs based on anomaly detection detect unusual behavior and thus have the ability to detect symptoms of attacks without specific knowledge of details.

- Anomaly detectors can produce information that can in turn be used to define signatures for misuse detectors.

The disadvantages are [1]:

- Anomaly detection approaches usually produce a large number of false alarms due to the unpredictable behaviors of users and networks.
- Anomaly detection approaches often require extensive "training sets" of system event records in order to characterize normal behavior patterns.

Once IDSs have obtained event information and analyzed it to find symptoms of attacks, they generate responses. Some of these responses involve reporting results and findings to a pre-specified location. Others involve more active automated responses. Though researchers are tempted to underrate the importance of good response functions in IDSs, they are actually very important. Commercial IDSs support a wide range of response options, often categorized as active responses, passive responses, or some mixture of the two. This thesis does not consider any response system after detecting a compromised node in a wireless sensor network. However, they are paramount in all Intrusion Detection Systems!

2.2 IDS Constraints in Wired versus Wireless Mediums

The above explanation on the ingredients of a full-fledged IDS is suited for both wired and wireless networks. However, as mentioned in Chapter 1, the wireless channels are quite erratic. The channel parameters are extremely dynamic and unpredictable. Therefore the approaches used to study wireless sensor network security are often a subset of the above explanation due to the unpredictable channel and the resource-constrained sensors. Therefore, it suffices to say the main contributor to the difference between an IDS on a wired network and a wireless network is the medium. The medium dictates the hardware architecture, routing protocol, security algorithm, energy limitation, bandwidth utilization and signal-to-noise ratio (SNR). From Table 2.1 one can clearly conclude that the small sensor node can only accommodate a small IDS system while the workstation can accommodate a large and complex IDS.

Table 2.1 Comparison of a Smartdust Sensor with a Workstation

Parameters	**Smartdust Sensor**	**Typical Workstation**
CPU	8 bit, 4 MHz	32 bit, 2 GHz
Storage Flash Memory	8 KB instruction flash	64 KB
RAM	512 bytes	256 MB
EEPROM/ROM	512 bytes	20 GB hard drive
Communication	916 MHz radio	100 Mbps Ethernet link
Bandwidth	10 Kbps	100 Mbps Ethernet link
Operating System	TinyOS	Linux, NT
OS code space	3500 bytes	~ 64 MB
Available code space	4500 bytes	~ 128 MB
Routing Protocol	LEACH	RIP
Energy Limitation	AA battery 3.5 V	AC power supply 120 V
SNR	Bad	Not a factor
Security Algorithm	µTESLA	RSA

LEACH – Low Energy Adaptive Clustering Hierarchy RIP – Routing Information Protocol
µTESLA – Micro Timed Efficient Streaming Loss-tolerant Authentication
RSA – Public-key algorithm developed by Ron Rivest, Adi Shamir and Leonard Adleman

2.3 Summary

The type of IDS used in this thesis research was a Network-based IDS. The analysis employed was the Anomaly Detection. The data was studied using a combination of the Threshold and Statistical detection analysis methods. In a wireless sensor network, the sensor node capabilities are very inferior to those of a workstation in an Ethernet network. The unpredictable wireless channel with limited bandwidth and energy were simulated using a Linux software switch. However, the sensor nodes were simulated with workstations. Thus, all of the sensor's constraints listed in Table 2.1 were not considered. The main constraints considered included bandwidth, packets dropped, packets transmitted and packets received. These parameters can be used to indirectly calculate the energy associated with the captured data in the future.

CHAPTER 3

SENSOR NETWORKS

3.1 Types of Sensor Networks

During the past two decades, there has been an unprecedented growth in the number of products and services, which utilize information gained by monitoring and measuring using different types of sensors. Sensors monitor and quantify parameters under investigation. A sensor responds to an input quantity by generating a functionally related output usually in the form of an electrical or optical signal.

The development of sensors to meet the need is referred to as sensor technology. This technology is used to configure sensor networks of different shapes and sizes. The type of network depends heavily on the type of application running on the sensor nodes. Sensing principles include, but are not limited to mechanical, chemical, thermal, electrical, chromatographic, magnetic, biological, fluidic, optical, ultrasonic and mass sensing.

Sensors may be exposed to hostile environments. They may be incorporated in mobile robotic systems, or integral to manufacturing systems. Their environment may include high temperatures, high vibration, high noise, or corrosive chemicals. In biological systems, the sensors themselves must not adversely affect the system or organism. Sensor networks are comprised of hundreds to thousands of nodes, where each node is a sensor. These sensor nodes co-operate to carry out some task. These are used to guide as well as control data collection and aggregation. These nodes may be organized in clusters such that a locally occurring event can be detected by most of, if not all, the nodes in a cluster. Each node will have sufficient processing power to make a decision, and it will be able to broadcast this decision to the other nodes in the cluster. One node may act as the cluster master, and it may also contain a longer-range radio using a protocol such as IEEE 802.11 or Bluetooth [2].

Two ways to classify sensor networks are whether or not the nodes are individually addressable, and whether the data in the network is aggregated. The following paragraphs give various sensor network applications. These widely varying sensor applications provide an in-depth insight into the different types of IDSs needed to provide security. Each application is bolded followed by a description.

Bunker Mapping: This can be used in the military field when trying to find out about underground facility that has been constructed [2]. The geometry of the facility is unknown, in terms of size, depth, and shape. Vehicles enter and exit the facility on a fairly regular basis. In this case, the requirement is to determine the geometry of the underground facility.

A small passive measurement device (Mote) is attached to a vehicle before it enters the facility. It later downloads the sensor data later. This is used to reconstruct part of the internal structure of facility. With multiple data sets, a comprehensive map of the internals of the facility is constructed. Either radio frequency (RF) or line-of-sight optical communication to some local retransmitter downloads the sensor data. The Mote may be placed manually, which is considered the best chance for hiding it, and guaranteeing any alignment that may be necessary. On the other hand it would be most risky. It may also be delivered in a ballistic way, by firing the Mote from a gun of some kind.

The following combination of sensors, a three-axis accelerometer, three-axis gyro, and three-axis magnetometer together with a microprocessor, bi-directional RF communication, and power supply can be built on a Mote in a volume of less than one cubic inch. This is probably small enough to be used under some circumstances. As for the power requirements, even with existing off-the-shelf components, it would give a lifetime of days to months depending on duty cycle. The sensor nodes could be augmented with a variety of other sensors, such as an image sensor. In addition, integration of the image compression circuitry with the imager would make for a small, lower power system.

Dynamically Placed Intrusion Sensor Networks: Military units clearing urban terrain must clear a building, but cannot afford to leave people behind to ensure that it stays cleared [2]. The goal is to notify the force if anyone enters the cleared portion of the building after they have left. Soldiers would a dispenser, possibly attaching it to their weapons, filled with sensor nodes that could be shot or emplaced quickly by hand on a wall, stairwell, or doorway. The sensors, using some combination of acoustic, infrared, visual, or vibration signals would pass information about intruders to the appropriate person.

The scenario of sensing would use the soldier's voice message as he put the sensor node. This would be thee verbal message that would be relayed to the soldiers when that sensor detect an intruder. This could be done today with off the shelf components in the cubic inch size range. To be militarily useful it would certainly require substantial modification.

Distributed Surveillance Sensor Network (DSSN): This network program is used to investigate the applicability of small, inexpensive undersea vehicles to surveillance applications and submarine connectivity [2]. It is based on the concept of a fleet of autonomous undersea vehicles, which gather surveillance data and communicate acoustically. Each occasionally docks at an underwater station to dump its data, recharge its batteries, receive any new mission instructions and perhaps remain dormant until its next deployment. The docking station is self powered and is not connected to shore or ship by communications cable. The massive quantity of accumulated data is retrieved at a later time by means of a remotely controlled vehicle guided to the docking station by means of a fiber optic micro-cable (FOMC). The FOMC is the high-bandwidth channel by which the data is recovered and instructions are downloaded to be disseminated to the surveillance fleet.

Autonomous Ocean Sampling Network (AOSN): AOSN is a distributed, highly mobile, adaptive sensor network composed of a mix of autonomous underwater vehicles (AUV's), which exhibit complementary capabilities [2]. It is developed for oceanographic characterization. The architecture is very general; hence the mix of AUV's and their payloads can be optimized for specific mission scenarios, making the concept both highly flexible and very powerful. The AOSN concept is predicated upon the assumption that the current geometric growth in signal processing power continues into the future.

Besides its increasing capabilities and driving costs down, this trend ultimately permits a single hardware device to support multiple applications. For example, digital signal processing (DSP) chips are used to compensate for multipath propagation in the current generation of acoustic modems developed for AOSN. As DSP's become more capable they will be able to support higher reliable data transfer rates. More importantly, as increased processor speed becomes commercially available enough signal-processing capability will eventually exist to permit the extraction of information from the multipath signals themselves (which are currently only discriminated against). This capability configures the AUV communications network into huge multi-static active sonar capable of detecting and localizing anomalies within the volume of seawater supporting the acoustic propagation paths. In time the same basic hardware, which was originally employed for data communications, can simultaneously detect mines and submarines in the water volume --- with only an upgrade in the silicon! This is a striking, but realistic, example of the effectiveness of selecting a system's architecture to take maximum advantage of expected technological evolution.

Digital Traffic Pulse Sensor Network: The Traffic Pulse network is the foundation for all of Mobility Technologies applications [2]. This network uses a process of data collection, data processing, and data distribution to generate the most unique traffic information in the industry. Digital Traffic Pulse collects data through a sensor network, processes and stores the data in a data center, and distributes that data through a wide range of applications. In these applications the sensor network is the one that comprises the core of Digital Traffic Pulse that combines new and unique digital traffic gathering technology with proven non-digital traffic gathering techniques.

Installed along major highways, the digital sensor network gathers lane-by-lane data on travel speeds, lane occupancy, and vehicle counts. These basic data elements make it possible to calculate average speeds and travel times. The data is then transmitted to the data center for reformatting. The network continuously monitors roadway conditions and provides updates to the data center in real time.

In each city, Mobility Technologies maintains a Traffic Operations Center that collects and reports on real-time event, construction, and incident data. This information supplements the data collected from the sensors. Each center produces the information through a wide range of methods: video, aircraft, mobile units, and monitoring of emergency and maintenance services frequencies.

A digital sensor system is installed in the public right-of-way to gather lane-by-lane travel speeds, lane occupancy and vehicle counts that produce point-to-point travel times. The digital data is supplemented with real-time event, construction, and incident reports that are produced using traditional traffic gathering methods. The data sources include video images, aircraft, mobile Units, police and emergency frequencies, other information, public agency information, construction alerts, port information, weather information and transit information.

The traffic data is combined and processed into real-time information for immediate distribution. The data is also archived for eventual use in historical and predictive analyses. The digital traffic information is combined with its traditional incident and event information and transmitted to the National Transportation Data Center (NTDC), the heart of the Traffic Pulse system. The data is then converted into real-time information for immediate distribution and also stored for historical and predictive analysis.

The data is sent to the NTDC every 60 seconds. The data is instantly processed and made immediately available customers via the Internet. This information is also stored in an Archive Database where it is combined with the geo-located traditional incident and event information, all highly valuable for data mining purposes. The NTDC is the commercial grade data center where real-time traffic information is processed and historical traffic information are archived into a database. Located in Philadelphia, Pennsylvania, this multi-tiered architecture provides a highly scalable and flexible platform on which products and services are based.

Using the Internet, superior traffic information is provided to the public via its consumer website at www.traffic.com. Eventually, commuters will be able to create custom commuter profiles and request that their traffic information is sent to them upon demand. Commuters access the superior, real-time traffic reports via radio and television stations across the country.

The intelligent transportation systems provide real-time and archived traffic information to public agencies for the purposes of traffic management, planning, and reporting. The process of developing in-vehicle traffic solutions for both the personal commuter (Telematics) and commercial fleet customers (Enterprise Solutions) are in progress. Commuters will be able to receive real-time, customized traffic updates through their cell phones, PDAs, pagers, and other wireless devices in the very near future.

Wireless Sensor Networks for Habitat Monitoring: This deployment of sensors construct a network that consists of many nodes on a monitored landscape, streaming useful live data onto a computer [2]. This application-driven design exercise serves to identify important areas of further work in data sampling, communications, network re-tasking, and health monitoring.

Habitat and environmental monitoring represent a class of sensor network applications with enormous potential benefits for scientific communities and society as a whole. Supplying natural spaces with numerous networked micro-sensors can enable long-term data collection at scales and resolutions that are difficult, if not impossible, to obtain otherwise. The intimate connection with its immediate physical environment allows each sensor to provide localized measurements and detailed information that is hard to obtain through traditional instrumentation. The integration of local processing and storage allows sensor nodes to perform complex filtering and triggering functions, as well as to apply application-specific or sensor-specific data compression algorithms.

The ability to communicate not only allows information and control to be communicated across the network of nodes, but nodes to cooperate in performing more complex tasks, like statistical sampling, data aggregation, and system health and status monitoring. Increased power efficiency gives applications flexibility in resolving fundamental design tradeoffs, e.g., between sampling rates and battery lifetimes. Low-power radios with well-designed protocol stacks allow generalized communications among network nodes, rather than point-to-point telemetry. The computing and networking capabilities allow sensor networks to be reprogrammed or retasked after deployment in the field. Nodes have the ability to adapt their operation over time in response to changes in the environment, the condition of the sensor network itself, or the scientific endeavor. Architecture that interconnects the core system components ranging from very localized collections of sensor nodes to the area of study to the wide-area where data is ultimately analyzed.

Sensor networks represent a significant advance over traditional invasive methods of monitoring. Sensors can be deployed prior to the onset of the breeding season or other sensitive period (in the case of animals) or while plants are dormant or the ground is frozen (in the case of botanical studies). Sensors can be deployed on small islets where it would be unsafe or unwise to repeatedly attempt field studies. The results of wireless sensor-based monitoring efforts can be compared with previous studies that have traditionally ignored or discounted disturbance effects. Finally, sensor network deployment may represent a substantially more economical method for conducting long-term studies than traditional personnel-rich methods.

The sensor networks must be accessible via the Internet. An essential aspect of habitat monitoring applications is the ability to support remote interactions. Field station needs resources to host Internet connectivity and database systems. A second tier of wireless networking provides connectivity to multiple patches of sensor networks deployed at each of the areas of interest. Three to four patches of 100 static (not mobile) nodes are sufficient to start.

Sensor networks that run for 9 months from non-rechargeable power sources would be required. Although ecological studies span multiple field seasons, individual field seasons typically vary from 9 to 12 months. Every level of the network must operate with bounded energy supplies. Although renewable energy, for example solar power, may be available at some locations, disconnected operation remains a possibility.

The remoteness of the field sites requires the ability to monitor and manage sensor networks over the Internet. The goal is zero on-site presence for maintenance and administration during the field season, except for installation and removal of nodes since Habitat monitoring infrastructure should not disrupt the natural processes or behaviors under study. Local interactions are required during initial deployment, during maintenance tasks, as well as during on-site visits. It is critical that sensor networks exhibit stable, predictable, and repeatable behavior whenever possible. An unpredictable system is difficult to debug and maintain. Archiving sensor readings for off-line data mining and analysis are essential. The reliable off loading of sensor logs to databases in the wired, powered infrastructure is an essential capability. The timely delivery of fresh sensor data is a key. Nodal data summaries and periodic health-and-status monitoring requires timely delivery.

The architecture of the sensor networks is a tiered architecture. Lowest level consists of the sensor nodes that perform general purpose computing and networking in addition to application–specific sensing. Autonomous sensor nodes provide the lowest level of the sensing application. These small, battery-powered devices are placed in areas of interest. Each sensor node collects environmental data primarily about its immediate surroundings. The sensors can often be built using small and inexpensive individual sensors. The gateway is responsible for transmitting sensor data from the sensor patch through a local transit network to the remote base station that provides WAN (wide area network) connectivity and data logging.

The base station connects to database replicas across the Internet. The data is displayed to scientists through a user interface. Mobile devices may interact with any of the networks, whether they are used in the field or across the world connected to a database replica. They can directly communicate with the sensor patch, provide the user with a fresh set of readings about the environment and monitors the network. While they will typically not take custody of any data, it allows the user to interactively control the network parameters by adjusting the sampling rates, power management parameters and other network parameters.

Compared with traditional data logging systems, networked sensors offer two major advantages. They can be retasked in the field and they can easily communicate with the rest of the system. This happens via communication between individual sensors and coordination with one another. The sensors will typically form a multi-hop network by forwarding each other's messages, which vastly extends connectivity options.

Bringing direct wide area connectivity to each sensor path is not feasible because of the cost of equipment. It requires too much power and the installation of all required equipment is quite intrusive to the habitat. Instead, the wide area connectivity is brought to a base station where adequate power and housing for the equipment is provided. Communication between the base station and the sensor patch will be via a wireless local area network. Wireless networks are particularly advantageous since often each habitat involves monitoring several particularly interesting areas, each with its own dedicated sensor patch.

The sensors must be chosen carefully to ensure high interchangeability and high accuracy. Each sensor has less than 3% variation when interchanged with others of the same model [2]. The accuracy of each sensor is within 3% of the actual value. The sensors can be deployed in the field quicker since little or no calibration is needed prior to deployment. Another key aspect of choosing a sensor is its startup time. The start up time is the time a sensor must be powered before its reading stabilizes. Sensors with long start up times require current for a longer period of time, resulting in higher power consumption. Minimizing start-up time yields more power per day to perform other tasks, such as routing and communication.

Many habitat-monitoring applications need to run for nine months - the length of a single field season. Mica runs on a pair of AA batteries, with a typical capacity of 2.5 ampere-hours (Ah). However one can neither use every drop of energy in the batteries nor are the batteries manufactured with identical capacities from manufacturer to manufacturer [2]. A conservative estimate will be made that the batteries will be able to supply 2200 mAh at 3 volts. Assuming the system will operate uniformly over the deployment period, each node has 8.148 mAh per day available for use. The application chooses how to allocate this energy budget between sleep modes, sensing, local calculations and communications. It is noted that since different nodes in the network have different functions, they also may have very different power requirements. In any network, there will be some set of power-limited nodes; when these nodes exhaust their supplies, the network is disconnected and inoperable. Power is to be budgeted with respect to the energy bottleneck of the network. The baseline lifetime of the node is determined by the current draw in the Sleep State. Minimizing power in sleep mode involves turning off the sensors, the radio, and putting the processor into a deep sleep mode.

Power efficient communication paradigms for habitat monitoring must include a set of routing algorithms, media access algorithms, and managed hardware access. The routing algorithms must be tailored for efficient network communication while maintaining connectivity when required to source or relay packets. A simple routing solution for low duty cycle sensor networks is simply broadcasting data to a gateway during scheduled communication periods. This method is only communicated in one direction and there is no dependency on surrounding nodes for relaying packets in a multi-hop manner. The sensor nodes in burrows are transmitting only with a low duty cycle - they sample about once per second. Accordingly, a multi-hop-scheduled protocol must be used to collect, aggregate, and communicate data.

Dual Sensor can provide more information and earlier warning for Diabetics [2]. It monitors glucose and insulin levels simultaneously, allowing better management of the disease. This tiny sensor can be implanted in the human's body. The first device of its kind to measure the glucose-insulin ratio, it could help predict changes leading to high or low blood sugar levels. The sensor detects excess glucose while the pump acts as an artificial pancreas, the organ that produces insulin in the body. Another device known as an insulin detector estimates the amount of the hormone in the blood. Combining two types of sensors in such a small device is a complicated task, because Glucose is measured in the milli-molar range, while insulin is tracked on even smaller nano-molar levels. Employing methods similar to those used to create existing implants; researchers designed a needle-shaped sensor that can be implanted in the body where it won't cause problems.

The next step is testing the device in animals. Researchers hope to halve the size of the sensor, reducing it to the thickness of 26-gauge wire (about the size of a hypodermic needle).

It consist of the following parts [2]:

- Satellite Monitor: A microprocessor-based device incorporating a configurable screen display for continuous data; icon-based keys for selecting data and printout; intuitive operator interface and built-in printer. The sensor is calibrated in the calibration unit.
- Patient Data Module (PDM): Detachable Patient Data Module enables stored patient data to be transferred with the patient and retains historical patient data over 24-hour period. It also provides the interface between the sensor and the monitor.
- Artificial Retina: Currently, smart sensor chips, each with 100 micro-sensors, have been built for an ex-vivo testing system of a retina. The smart sensor has two components (1) an integrated circuit (IC) and (2) an array of sensors. The integrated circuit is a multiplexing chip, operating at 40KHz, with on chip switches and pads to support a 10x10 grid of connections. The circuit has both transmitting and receiving capabilities. Each connection has an aluminum probe surface where the micro-machined sensor is bonded. This is accomplished by using a technique called backside bonding, which places an adhesive on the chip and allows the sensors to be bonded to the chip, with each sensor located on a probe surface. Before the bonding is done, the entire IC, except the probe areas, is coated with a biologically inert substance.

Chemical Vapor Sensor System: Artificial neural networks (ANNs) are used in a wide variety of data processing applications where real-time data analysis and information extraction is required [2]. One advantage of the neural network approach is that most of the intense computation takes place during the training process. Once the ANN is trained for a particular task, operation is relatively fast and unknown samples can be rapidly identified in the field. ANNs have been used in a wide variety of applications related to manufacturing. These applications include process control, quality control, industrial inspection, optimization, and modeling. There are many real-time (rapid response) and remote sensing applications that require an inexpensive, compact, and automated system for identifying an object (e.g., target, chemical, and isotope). Such a system can be built by combining a sensor array with an ANN.

The quantity and complexity of the data collected by sensor arrays can make conventional analysis of data difficult. ANNs, which have been used to analyze complex data and for pattern recognition, could be a better choice for sensor data analysis. A common approach in sensor analysis is to build an array of sensors, where each sensor in the array is designed to respond to a specific physical quantity. Here, the number of sensors must be at least as great as the number of physical quantities being monitored. When an ANN is combined with a sensor array, the number of detectable physical quantities is generally greater than the number of sensors.

A sensor array is composed of several sensing elements, where each element measures a different property of the sensed sample [2]. Each object (e.g., target, chemical, isotope) presented to the sensor array produces a signature or pattern characteristic of the object. By presenting many different objects to the sensor array, a database of signatures can be built up. From this database, training sets and test sets are generated. These sets are collections of labeled patterns (signatures) representative of the desired identification mapping. The training sets are used to configure the ANNs. The goal of this training is to learn an association between the sensor array patterns and the labels representing the data.

When a chemical sensor array is combined with an automated data analysis system (such as an ANN) to identify vapors, it is often referred to as an artificial nose. Several researchers have developed artificial noses that incorporate ANNs for use in applications including monitoring food and beverage odors, automated flavor control, analyzing fuel mixtures, and quantifying individual components in gas mixtures. The prototyped ANN was constructed as a multi-layer feed forward network and was trained with the back propagation of error algorithm by using a training set from the sensor database.

During operation, the sensor array "smells" a vapor, the sensor signals are digitized and fed into a computer, and the ANN (implemented in software) then identifies the chemical. This identification time is limited only by the response of the chemical sensors, but the complete process can be completed with in a few seconds.

Optical Sensor System: This prototype system employs an array of optical sensors and identifies the composition of chemical dyes in solution [2]. Light is passed through the dye solution and into an array of seven optical sensors. Each optical sensor consists of a silicon detector covered by a narrow bandpass interference filter and is sensitive to a specific wavelength of light in the visible and near-infrared spectrum. The output of each sensor provides an input to the ANN. By examining the absorption of the liquid at different wavelengths, the ANN is able to identify and quantify the dyes. Initial tests with this system are in their infancy.

Sensor excitation and sensor electronics power requirements are intimately related to the thermal noise in the sensor itself. In regards to power consumption in communication systems, there are so many variables that come into play in evaluating performance of these systems. However, the fundamental limits are again related to thermal noise. A receiver with a noise bandwidth B, temperature T, and K sensors, the thermal noise power from the antenna is KTB.

High-Energy Shaker Monitoring: The equipment generates displacements and accelerations that are capable of shaking sand from huge castings and feeding railroad-car size loads of coal and lumber across conveyors at a specific rate [2]. As might be expected, vibratory equipment can induce a small portion of the high-energy load forces back through its own structural support members. These resonant frequencies with relatively small amplitudes can produce some vibration within the structure's steel members, foundations, control panels, and office buildings.

The frequencies and displacements are distributed over the structural machine members often in complex patterns and can vibrate in characteristic phase relationships that either add together to increase resonance and the chance of failure, or cancel to reduce the resonance. As many as eight different members or machines vibrating at once can produce excessively high destructive internal forces and audible noise. This has to be minimized or avoid the frequencies and forces both during the machine design phase and after the machines are installed in the field.

Unwanted forces and frequencies generated in vibrating equipment are most typically a result of weak support structures, insufficient foundations, and poor soil conditions under the foundation, city water mains, or just random resonance. Therefore, designing methods to provide sufficient structural damping while maintaining the rigidity necessary to sustain the shock and vibration is a challenge. In order to help determine the machine geometry as well as the balancing and damping, the initial design undergoes 3D modeling and the machine runs real-time testing with multiple-channel data acquisition equipment.

3.2 Sensor Networks IDS Constraints and Proposed Solutions

It is critical that sensor networks exhibit stable, predictable, and repeatable behavior whenever possible. An unpredictable system is difficult to keep secure, debug and maintain. However, even in a well-designed predictable network, the security constraints on sensor networks are still a major topic of discussion. It is difficult to make a sensor network secure! The proposed solutions include traffic encryption key, key cryptography, threshold cryptography, certificate repository, watchdog and pathrater and reverse metempsychosis [4]. These techniques are all under research in attempts to provide security to sensor networks. The problem is, however, there is no "one size fits all" security measure to formulate and IDS for sensor networks. Sensor networks are too specific in nature. For example, a habitat sensor network monitoring animals needs an IDS that supports mobility whilst another network monitoring radiation in the atmosphere needs a static IDS.

In key cryptography, sensors can only communicate with other sensors that share the same key [10]. This is important since the sensors communicate by using the broadcast primitive. Those sensors that possess the same key "hear" the same broadcasted message. The other wireless nodes will receive such signal as noise. This authenticates the data by avoiding information leakage. However, sensors are limited by their low energy supply and small memory. Therefore, another technique embodies this phenomenon paying attention to these resource-constrained sensors – traffic encryption key.

The resource-constrained sensors can only use symmetric cryptography. Therefore a cluster of these sensors should share a traffic encryption key. The disadvantage is that once a node malfunctions, forward secrecy is broken. Therefore it is imperative that sensors are tamper-resistant [4]. Thus traffic encryption key is insufficient as an IDS.

Zhou proposed the principle of threshold cryptography [4]. A management server not only has to store its own key pair, but also the public keys of requirement exerted on the servers which must potentially be specialized nodes in the network, and the overhead in signing and verifying routing message both in terms of computation and of communication.

Hubaux expanded Zhou's philosophy by insisting that each node must keep its own certificate repository. "These repositories store the public certificates the node themselves issue, and a selected set of certificates issued by the others. The performance is defined by the probability that any node can obtain and verify the public key of any other user, using only the local certificate repositories of the two users. The dilemma is: too many certificates in a sensor node would easily exceed their capacity, yet too few might greatly impact the performance of the entire network" [4].

The watchdog and pathrater technique uses a watchdog in promiscuous mode that constantly monitors its neighbors forwarding activity [4]. The pathrater uses the reports of the watchdog to rate the transmission reliability of all routes to a particular destination node. This technique sound very flawless but it uses far too much of the sensor's resources. It uses a huge amount of energy, thus making it impractical.

Stajano proposed reverse metempsychosis as a way of protecting the keys embedded in the nodes [4]. The node is put to rest when it is not in use. Then it wakes up with another set of imprinted keys once it regains normal functionality. When a node fails to detect the presence of at least n nodes for some time, it would refuse to work or reset its keys. This method of protecting the keys is energy efficient. However, it does not address a recovery policy after a compromised node refuses to work. How will one recover after a compromised node refuses to work? The next section will address this question in attempt to describe a policy proposed by Dr. Constantine Manikopoulos and Ling Li – "the Neighborhood Watch" concept.

3.3 Neighborhood Watch IDS

It has already been established that the wireless environment is very unpredictable. This makes it very difficult to construct an IDS to protect these wireless sensor networks. Mobile Ad-Hoc Networks (MANETs) provides security engineers with two major challenges [7]:

- There is no node playing a central role.
- It is very difficult to handle a compromised node.

A wired network has a server acting as a central node. In this instance, central services like naming services, certification authorities and network-based intrusion detection provide security. These are impractical to secure MANETs. Each MANET node operates while not trusting any of its peers [7]. This gives each node the flexibility of deciding whether it should communicate with any of its peers. For example, if node A believes that node B is under attack, node A can choose to not communicate with node B simply because it already does not trust node B. This complicates and even inhibits security services because no one can tell if node A or B is acting appropriately. The solution to this problem is a Neighborhood Watch technology.

Two nodes are neighbors if they can communicate with each other. The Neighborhood Watch IDS works in conjunction with a host IDS running on each sensor node. Therefore, each node can detect an attack based on its own embedded IDS. For example, an escalation of privilege may hijack a node that detects it. The problem then is how will this information be processed to alert the other nodes of the attacked node? How will the entire network respond to such a change? The Neighborhood Watch attempts to answer these question be providing a recovery policy from such security breaches.

The following example explains how this works. As mentioned before, all nodes broadcast messages and the nodes in range with the appropriate key receive the messages. Thus in the Neighborhood Watch technology a status update message, IDS_Status, is broadcast periodically or triggered by an event [7]. The nodes within communicating range of each other will receive and respond to this message. Now suppose node A is compromised by means of a security breach. Then all neighboring nodes to node A will process this same data leading to identically valid evaluation of the security status of node A. Suppose there are five nodes, including node A, in the neighborhood of node A. All five nodes will cast an independent vote on the status of node A. Therefore, even if node A lies about its status, the majority voting will reveal the truth of the security breach occurring at node A.

Once node A is revealed as being compromised, the other nodes will make a decision to delete node A from the network. This is achieved by broadcasting a Delete (x) message. The network topology and thus its routing will be updated due to the deletion of node A. Thus by locally "watching" one's "neighborhood" each node can make a decision of the status of each other's security status. This is referred to as Local Collaborative Groups (LCGs). Therefore, the Neighborhood Watch IDS works as a complementary tool to an existing IDS. It detects and deletes a compromised node thereby recovering the network while avoiding this security breach from spreading throughout the network.

3.4 Summary

There are many types of sensor networks. Bunker Mapping, Dynamically Placed Intrusion Sensor Networks, Distributed Surveillance Sensor Network (DSSN), Autonomous Ocean Sampling Network (AOSN), Digital Traffic Pulse Sensor Network, Wireless Sensor Networks for Habitat Monitoring and Chemical Vapor Sensor System are various examples of sensor networks. The applications monitor events on land, in the sea, in the air, in plants and in animals. The actual network employed depends on the type of applications needed to monitor the physical quantity being measured. The security method varies with the type of network.

There is no "one size fits all" security measure that will protect all the networks described. However, once one understands the network and the measure of security needed, one can employ a strategy comprising of one or more of the security method proposed throughout this Chapter. Techniques like traffic encryption key, key cryptography, threshold cryptography, certificate repository, watchdog and pathrater, and reverse metempsychosis all have different shortfalls. The Neighborhood Watch IDS can be used in conjunction with the other proposed solutions to arrive at the level of security needed. All these techniques, however, are still in research and will take time to develop into their own!

CHAPTER 4

IMPLEMENTATION

4.1 Dynamic Switch Emulation Tool

Wireless mobile ad hoc networks differ from wired networks in that their topologies are highly dynamic and their links can have a relatively high error bit rate. These properties make it difficult to conduct controlled repeatable experiments for security engineers to study and research wireless ad hoc networks [5]. The Dynamic Switch Emulation Tool is a software that connects multiple unaltered hosts according to a controllable dynamic topology with a controllable bit error rate on the links. This switch emulates a wireless mobile network using standard Ethernet physical connections. This allows the study of security in a controlled environment!

4.1.1 Architecture and Leaky Bucket Model

The architecture of the Dynamic Switch hinges on the recompilation of Redhat Linux 7.0 kernel 2.2.16 [5]. The Dynamic Switch kernel modifies the communication protocol to that of a wireless nature while using its wired counterpart. A wired network is easy and inexpensive to construct in a laboratory. A MANET, on the other hand, is time consuming and expensive to set up. Therefore, the Switch provides the inexpensiveness of a wired network with wireless properties. The low bit error rate and high data rate in a wired network does not match the MANET environment. The Switch emulates higher bit error rates and low data rate using the wired connections. Traditional wired switches use the medium access control (MAC) and Internet protocol (IP) addressing information to forward packets. However, the Dynamic Switch uses just the interface card to forward packets from one node to another. It is sometimes referred to as the IP/MAC killer!

The basic concept of the Dynamic Switch is to use a star topology to connect multiple nodes with the Switch acting as the central hub [5]. The mobile nodes can run any software provided it has the appropriate interface card. The Switch runs on a standard PC. It has multiple network interface cards (NICs) and or multiple ports interface cards. Each connection to an interface card and or port represents a node on the wireless network. The Dynamic Switch can switch between any set of connected hosts based on a local switching connectivity table. It is transparent to all the nodes on the network at or above the MAC layer. All incoming frames are switched based solely on the connectivity table and the interface card. The switch adds no address information to the MAC frame or IP datagram.

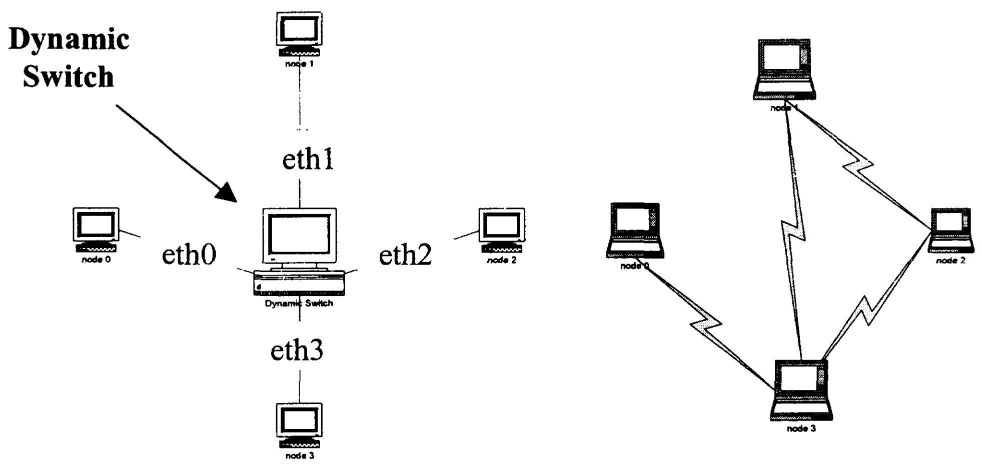


Figure 4.1.1.1 (a) Dynamic Switch Topology **(b)** Equivalent Wireless Network

The topology of the wired star network in Figure 4.1.1.1(a) shows four interfaces connected to the Dynamic Switch. They are denoted eth0, eth1, eth2 and eth3. These interfaces correspond to the nodes in the wireless equivalent network in Figure 4.1.1.1(b). This network is created using a simple script. The script to construct such a wireless topology is:

```
./process 0 3 up
./process 1 2 up
./process 1 3 up
./process 2 3 up
```

The first line indicates that the link between interface 0 and interface 3 is up. Thus node 0 and node 3 are within communication range. The rest of the script establishes the links displayed in Figure 4.1.1.1(b). The Dynamic Switch in the equivalent wireless network is transparent. Thus the nodes have no knowledge of its existence and can only "see" those nodes within communication range. The routing protocol used in the Dynamic Switch is the Optimized Link State Routing Protocol (OLSR). This is a proactive protocol. This means that the nodes on the wireless network periodically transmit "hello" messages and link state changes.

The packet drop rate is high in wireless communication. This is implemented on the Switch by controlling the pack drop rate on the links. A Gilbert model, a two-state discrete-time Markov model, was employed to simulate dropping packets. "In the two-state Markov model a channel can be in one of two possible states, "good" or "bad". The state transition diagram is shown in Figure 4.1.1.2. The probability of dropping a packet, that is, the probability of a packet error, is different in each state. P_G is the probability of dropping a packet while in the good state and P_B, $P_B > P_G$, is the probability of dropping a packet while in the bad state. Given a present state, a channel may transfer to the other state or stay in the present state with certain probabilities. P_1 and P_2 are transition probabilities of staying in the good and bad state respectively" [5].

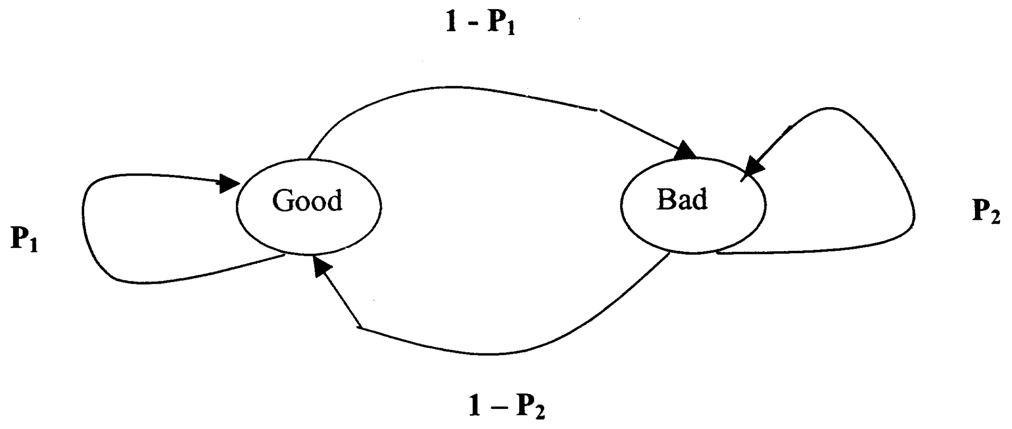


Figure 4.1.1.2 Two-state Markov Chain for Packet Drop Process

One of the limited resources in wireless communication is bandwidth. The Dynamic Switch implements the Leaky Bucket Token Model to emulate constrained capacity on the links. In this model there are three controllable parameters. The first is the size of the bucket, B tokens. The second is the token arrival rate, r tokens per second. The third is the allowable transmission rate, μ bytes per token. These parameters contribute to the maximum allowable data rate C by the following formula: $C = \mu \times r$ bytes per second implemented in the Leaky Bucket Token Model [5].

The Leaky Bucket Token Model forwards a packet only if a token is available in the token buffer. All packets must have a token to be transmitted. A packet is dropped if there is no token available. The tokens enter the bucket at the constant rate r tokens per second. The outbound traffic is controlled by μ that determines the number of bytes per token. The correct combination of these three parameters is very tricky. The selection of r involves a tradeoff between accuracy and overhead [5]. A high token arrival rate r results in transmissions being spread out over a long interval that more closely resembles a low data rate link. However, r must be reduced to the minimum timer interval supported by the operating system of the Switch host to reduce timer interrupt overhead. Figure 4.1.1.3 shows the relationship of r, μ and B and gives a pictorial explanation of the mechanism for forwarding and or dropping packets.

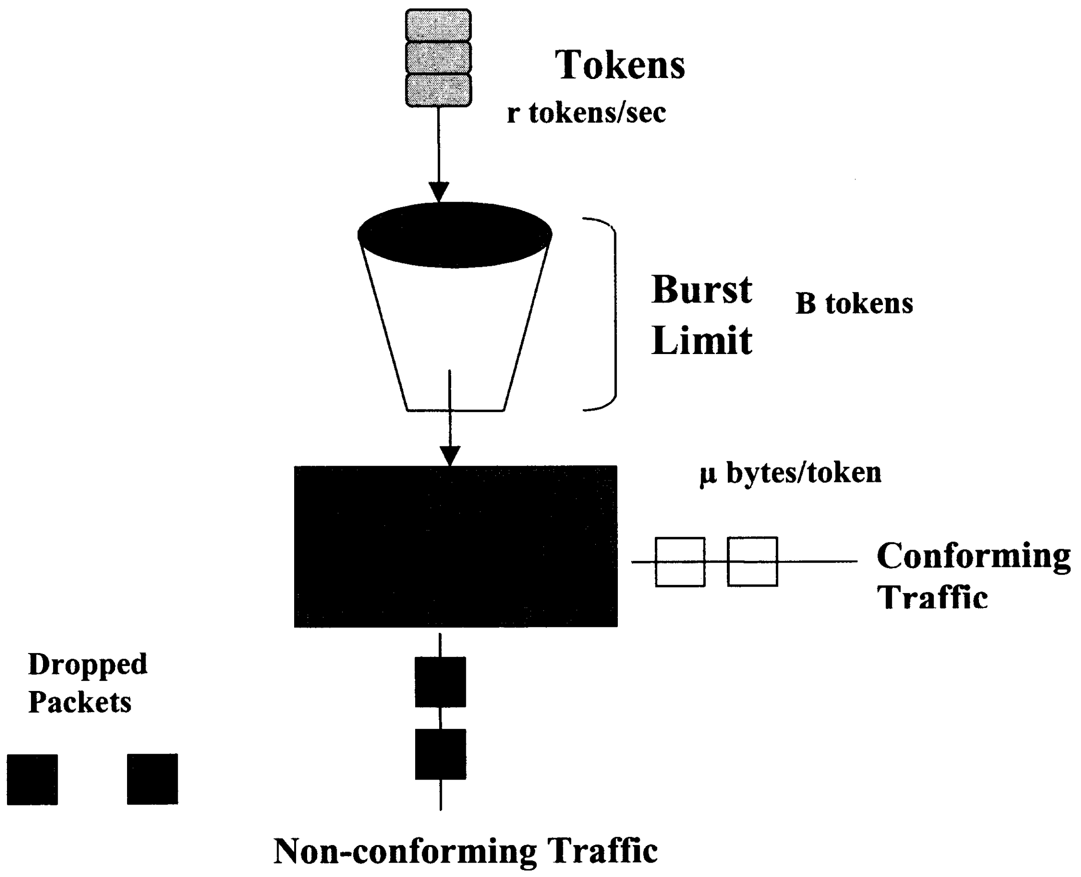
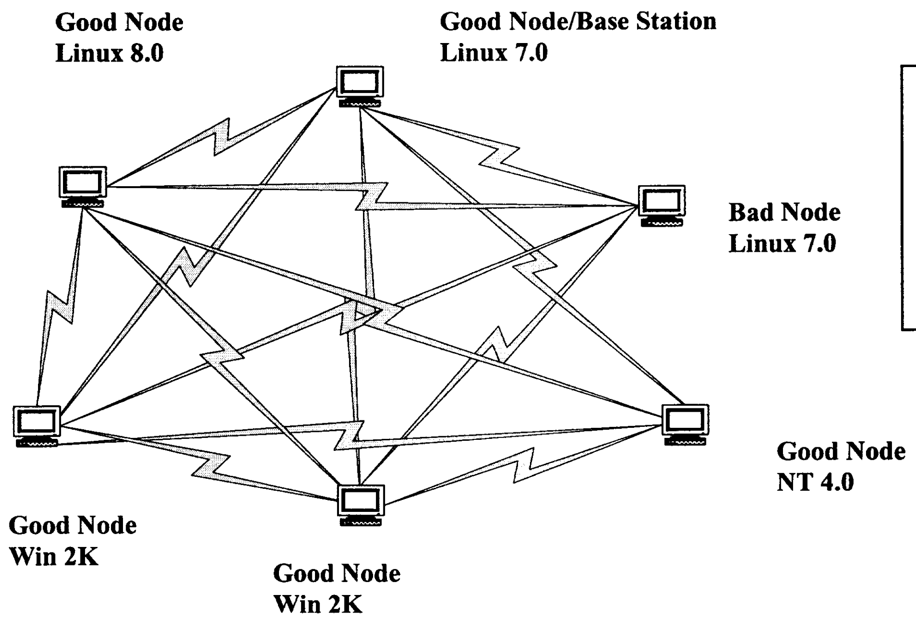


Figure 4.1.1.3 Leaky Bucket Token Model

4.1.2 Dynamic Switch Constraints

As mentioned before the operating system timer interrupt overhead posses a constraint on the Dynamic Switch. Hence, the bandwidth parameter on the Dynamic Switch cannot be accurately determined. However, the maximum channel capacity C = μ x r bytes per second is the best method of correlating the bandwidth utilization. The Switch does not emulate a wireless MAC layer protocol, so absolute delays and throughput in a MANET routing protocol cannot be accurately measured since these metrics, certainly in absolute terms, are sensitive to the performance of the MAC layer. However, using statistics and doing comparisons of various data sets can give one a good approximation of delays in the wireless network.

4.1 Simulation Results and Analysis



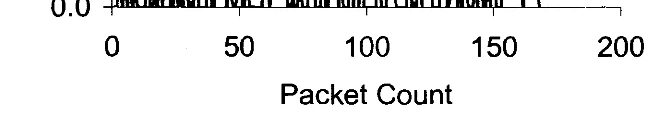
Parameter Values
μ = 50 bytes per token

r = 100 tokens per second

B = 100 tokens

Figure 4.2.1 Wireless Test Bed

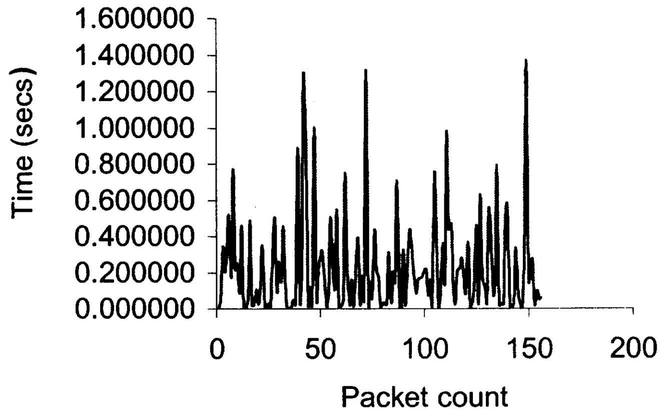
The simulation experimentations attempted to study the behavior of good nodes versus bad ones. The Statistical and Threshold Anomaly Detection technique explained in Chapter 2 was employed. Figure 4.2.1 shows the test bed used to analyze channel bandwidth utilization and delays in the good nodes and bad node. There were six nodes fully connected in a wireless network one was also used a base station. Then a client-server program that simulated self-similar background traffic was installed on each node. All nodes communicated with the base station. Another faulty client-server program simulated the bad node's operation. The assumption here is that a good node behaves different to a bad one. The data was gathered using tcpdump. This data captured statistics on every packet. This information was extracted from the tcpdump data in 20-minute time slots. These time slots were studied independently and then averaged over the entire time. The final results were graphed in Microsoft Excel. The graphical representations are explained below. The overall objective was to detect the faulty node based on the statistical difference and a threshold value.



Average delay = 0.694649 ms
Standard Deviation = 0.49

Figure 4.2.2 Statistical Delay on Good Node

The statistical delay on the good node was an average of 0.69 ms while the standard deviation was 0.49 shown in Figure 4.2.2. This was much better than the delay on the bad node that was 0.225192 s with a standard deviation of 0.89 shown in Figure 4.2.3. This huge difference in the delay indicates that the faulty node was detected statistically. A threshold value of 1 ms was chosen. The good node was within this value about 98 % of the time. The bad node on the other hand was far out of this range and was definitely detected.



Average delay = 0.225192 s
Standard Deviation = 0.89

Figure 4.2.3 Statistical Delay on Faulty Node

The channel utilization on the good node was approximately 2000 bytes per second, shown below in Figure 4.2.4. That for the bad node was about 3750 bytes per second, shown in Figure 4.2.5. This occurred because the faulty node used up more bandwidth than any of the good nodes. It constantly communicated with the base station since its status was constantly queried. Therefore, the bad node hogged the limited bandwidth.

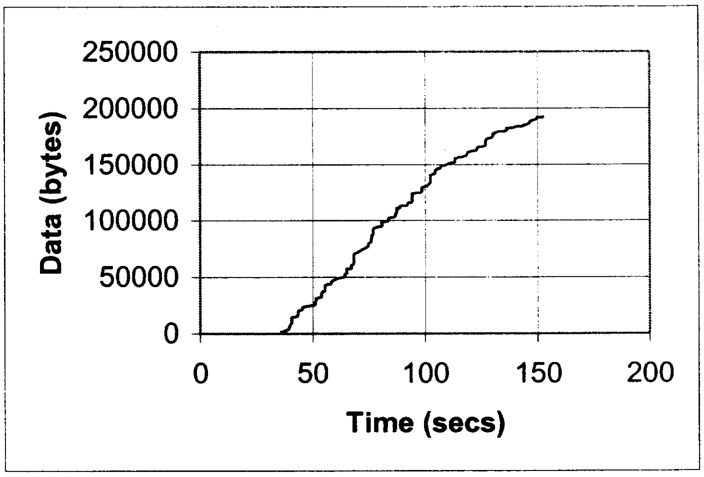


Figure 4.2.4 Channel Utilization on Good Node

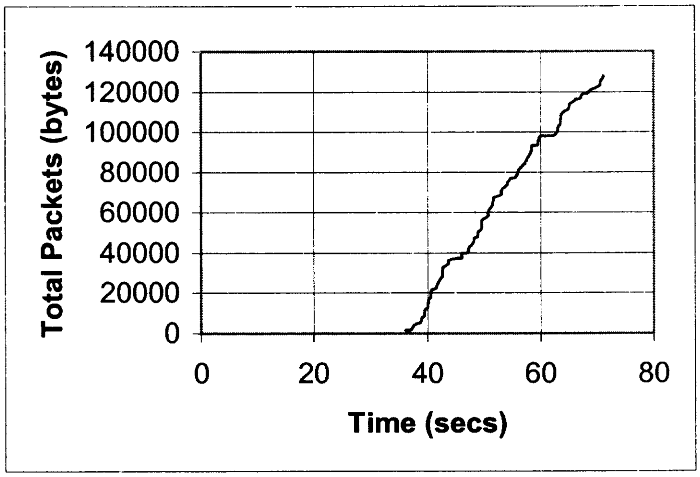


Figure 4.2.5 Channel Utilization on Faulty Node

4.2 Summary

The Dynamic Switch was used to emulate a fully connected wireless network with six nodes and one node acting as a base station. This software uses wired Ethernet links to emulate wireless channels by allowing interfaces to communicate with each other. The background client-server software simulated the background traffic while a faulty client-server program simulated the faulty node. The data was captured using tcpdump on a Linux node. These data were analyzed with Microsoft Excel graphing utility. A statistical study of 20-minute time slots of the data was carried out independently. Then the statistical average for the good nodes were plotted to give the "normal" behavior of good nodes. This was compared to the bad node and the results clearly showed that the faulty node was detected by the Statistical and Threshold Anomaly Detection techniques.

CHAPTER 5

CONCLUSIONS

This thesis never mentioned one of the most important limitations of wireless networks – energy. There is limited energy supply! This was not mentioned because the study only took two parameters into consideration. Energy is definitely a parameter that can be added to the future study of detecting a faulty node in wireless technology. The main reason it was left out is because energy efficient sensors heavily depend on the type of routing protocols being used. The Dynamic Switch only used one routing protocol, OLSR. Therefore, an energy study will be limited to this protocol and will not provide enough information to make plausible conclusions.

Still, the study of energy in wireless sensor networks should not be ignored. According to J. L. Gao [3], "energy consumption is one of the most important metrics for wireless ad hoc sensor networks because it directly relates to the operational lifetime of the network". These studies, however, concentrate on the trade-offs between low-energy routing and self-organizing protocols. The creators of the Dynamic Switch are hoping to add various protocols to the Switch. Once this is completed, one can study energy by taking each packet size and divide it by the bandwidth. This gives the following: (bits)/(bits per second) = seconds. Then take the answer in seconds and multiply it by the battery initial power, giving: (seconds)(joules/second) = joules. Then subtract this energy from the initial energy repeatedly, packet after packet. Eventually all the energy will be used up – this is equivalent to the battery's lifespan.

Even though the energy metric was ignored, the thesis still examined two very important metrics – channel utilization and delay. The wired IDS techniques proved to be inefficient in the wireless sensor network environment. The wireless sensors are very resource constrained and cannot support the security algorithms and protocols used in a wired network. The high bit-error rates, low bandwidth, low transmission rate and high delays make the wireless channels unpredictable. This makes it difficult to study security in actual wireless sensor networks since it is very time-consuming and expensive to set up a test bed. This is true irrespective of the type of wireless sensor network being studied.

The types of wireless sensor networks monitor various physical quantities on land, in the air, at the bottom of the ocean, in plants and in animals. For example, monitoring animals in their natural habitats and monitoring radioactive isotopes in the atmosphere are two types of wireless sensor networks. These sensor nodes run very different applications and are exposed to different atmospheric conditions. Therefore, they demand different approaches for their security. Although some proposals were made, including cryptography, encryption keys and the Neighborhood Watch techniques, there is no "one size fits all" security for wireless sensor networks. Each network is too specific and requires its own attention when implementing an IDS.

The Dynamic Switch solved the problem of setting up a time-consuming and expensive wireless sensor network but it had it own shortfalls. It emulated a wireless channel with a wired one. It provides pseudo-wireless capabilities where absolute delays and operating system timer interrupt overhead were concerns. However, by doing a Statistical and Threshold Detection analysis, this thesis detected the faulty node among good ones. The faulty node had much higher delays and much higher channel utilization. It was definitely detected!

REFERENCES

1. R. Bace and P. Bell, "Intrusion Detection Systems," Infidel, Inc., Scotts Valley, CA and National Institute of Standards and Technology, February 2001.

2. H. Bau, N. F. DeRooij and B. Kloeck, *Mechanical Sensors, Volume 7, Sensors: A Comprehensive Survey*, John Wiley & Sons, USA, December 1993.

3. J. L. Gao, "Analysis of Energy Consumption for Ad Hoc Wireless Sensor Networks Using a Bit-Meter-per-Joule Metric," Jet Propulsion Laboratory, California Institute of Technology, August 2002.

4. Y. W. Law, S. Dulman, S. Estalle and P. Havinga, "Assessing Security-Critical Energy-Efficient Sensor Networks," Department of Computer Science, University of Twente, June 2002.

5. T. Lin, S. F. Midkiff and J. S. Park, "A Dynamic Topology Switch for the Emulation of Wireless Mobile Ad Hoc Networks," Bradley Department of Electrical and Computer Engineering Virginia Polytechnic Institute and State University, January 2002.

6. S. Madden, R. Szewczyk, M. J. Franklin and D. Culler, "Supporting Aggregate Queries Over Ad-Hoc Wireless Sensor Networks," University of California, Berkeley, June 2002.

7. C. Manikopoulos and L. Li, "Architecture of the Mobile Ad-hoc Network Security (MANS) System," CONEX Laboratory, NJWINS Center, ECE Department, NJIT, August 2002.

8. S. Meguerdichian, F. Koushanfar, G. Qu and M. Potkonjak, "Exposure In Wireless Ad-Hoc Sensor Networks," Computer Science Department, University of California, Los Angeles, Electrical Engineering and Computer Science Department, University of California, Berkeley, Electrical and Computer Engineering Department, University of Maryland, March 2002.

9. R. Min and A. Chandrakasan, "Energy-Efficient Communication for Ad-Hoc Wireless Sensor Networks," Massachusetts Institute of Technology, July 2001.

10. A. Perrig, R. Szeczyk, V. Wen, D. Culler and J.D. Tygar, "SPINS: Security Protocols for Sensor Networks," Department of Electrical Engineering and Computer Sciences University of California, Berkeley April 2001.