Abstract

Sensor networks represent new paradigm for reliable environment monitoring and information collection. They hold the promise of revolutionizing sensing in a wide range of application domains because of their reliability, accuracy, flexibility, cost-effectiveness, and ease of deployment. Furthermore, in future smart environments, it is likely that sensor networks will play a key role in sensing, collecting, and disseminating information about the environment.

Each sensor is a tiny device with limited computing power, scarce memory and limited battery power. For wireless micro-sensor networks, physical environment and air medium present error prone medium with low bandwidth. Thus, the protocols for sensor networks should take into account the above mentioned challenges and characteristics . Communication in a sensor network is not an end-to-end communication as that of a traditional network; rather, it is a collective communication operation from the sensors to the observers. Each network is highly specialized and designed for a specific application. Thus, the first contribution of this thesis is to classify sensor networks in terms of factors that affect communication and the implications of these factors on network design. By understanding and isolating these factors, we can come up with classes of interesting sensor network organizations and use them to develop representative scenarios for evaluating sensor network organization alternatives.

To validate this taxonomy, the thesis develops a framework that isolates the different subsystems within a sensor network. Within this framework alternative scenarios can be constructed and different design options at the three design levels (infrastructure, protocol

and application) can be analyzed under uniform assumptions. Using this framework, we present an analysis of the fundamental infrastructure design decisions. The infrastructure of a sensor network represents the type of sensors, their number and their deployment strategy. For two types of reporting models, the thesis studies the effect of the infrastructure on the performance of the network in both networking measures and application specific measures. This analysis demonstrates several important conclusions that were not known previously: (1) it is just as important to determine what sensors report and when they report as it is to transport these samples effectively; (2) a denser network is not always better if the sensors are not carefully managed; and (3) congestion is a problem that can harm network level performance as well as application level performance. As a final contribution, we present an outline of a general solution for handling congestion in sensor networks.

To summarize, the contributions of this thesis are

1. A taxonomy of wireless micro-sensor networks communication models.

2. Development of a frame-work to evaluate protocols for sensor networks.

3. An experimental study of infrastructure tradeoffs for sensor networks.

4. An outline of a general solution for Congestion Manager module for sensor networks.

Acknowledgements

First, I would like to thank my advisor Prof. Nael Abu-Ghazaleh. I would like to thank Nael for all his patient advice and support on matters technical and otherwise. I will be forever greatful to Nael for all the things I have learned from him during the countless hours we spent in his office discussing exciting research ideas. Nael has been a great advisor and friend. Its been a wonderful experience to work with Nael. I wish Nael and Talla good luck.

I would like to thank Prof. Wendi Heinzelman for all her help and advice. Wendi has been extremely supportive and encouraging. I thank her for all the countless hours we spent working on research ideas. Working with Wendi has been a joyful experience. I thank her for always being there for me whenever I needed her help.

I would like to thank Prof. Michael Lewis for all his support and for being there on my thesis committee.I thank him for giving me the opportunity to server as a TA for his two courses and for confidence that he always had in me. Mike has been always very encouraging and helpful.

I would like to thank Prof. Patrick Madden for his advice and help.

I would like to thank my friend Raghunath Parthasarathy. I can always count on his friendship. I thank Niranjan Desai for his support. I would like to thank Sandeep Khanzode for all his support. Sandeep was always there to help me out.

I am indebted to my parents and my bother Sagar, who filled my life with love and have always provided support and right direction. I can always count on my Mamma and Baba. I would like to thank my grandparents Nana, Inni for their love and support. Nana and Inni have been my inspiration all the time. Avimama has been always very supportive and helpful. I would like to thank Dr. Karve family for their help. I want to thank my all maternal relatives for their support and love.

I would like to dedicate this thesis to my grandparents Nana and Inni.

Contents

List of Figures

Chapter 1

Introduction

Advances in VLSI, MEMS and wireless network technologies have placed us at the doorstep of a new era where small wireless devices will provide access to information anywhere anytime as well as actively participate in creating smart environments. One of the applications at the forefront of smart spaces is *sensor networks*: networks that are formed when a set of small untethered sensor devices are deployed in an ad hoc fashion and cooperate on sensing physical phenomena [6, 9, 29]. Sensor networks hold the promise of revolutionizing sensing in a wide range of application domains because of their reliability, accuracy, flexibility, cost-effectiveness, and ease of deployment. They will also "provide one of the missing connections between the Internet and the physical world" [24] and likely play an important role in providing the sensing aspect for smart spaces.

Contrary to traditional data networks where the communicating nodes represent ends of a connection, sensors in a wireless network are not ends in themselves; rather they are instruments for measuring information and relaying it to one or more observers. The nature of the sensor nodes themselves is also different: they are powered by small batteries and have limited computational resources [6]. While sensor networks share some of the operational characteristics of wireless ad hoc networks – they are distributed, self-organizing and cooperate in relaying information – they are significantly different in terms of infrastructure, performance metrics, application, and protocol requirements. Thus, there is a need for protocols to effectively manage these networks to improve their performance and extend their lifetime.

To motivate the challenges in designing sensor networks, consider the following scenarios: sensors are rapidly deployed in a remote inhospitable area for a surveillance application; sensors are used to analyze the motion of a tornado; sensors are deployed in a forest for fire detection; sensors are attached to taxi cabs in a large metropolitan area to study the traffic conditions and plan routes effectively; and smart Kindergarten [40] where sensor networks are deployed to create a developmental problem-solving environment for early childhood education.

Because of these factors, the design of sensor networks is significantly different than that of data networks. Furthermore, with the presence of a large number of potential applications that have widely different characteristics, it is unlikely that a single solution will fit all situations. There is a need for understanding the types of sensor networks and the appropriate sensor network organization for the different types. This thesis addresses this problem.

The thesis first develops taxonomy of sensor networks along axes that are likely to influence communication in sensor networks. Based on this taxonomy, the thesis develops an evaluation environment that isolates the three design layers and allows investigation of tradeoffs under uniform assumptions. The environment also allows modeling of different types of scenarios to allow designers to carry out what-if analysis with different design options. The final contribution of the thesis is analysis of the design tradeoffs at the infrastructure level. The study analyzes the effect of changing the sensor organization and the deployment strategy on the performance of the sensor network in terms of both networking metrics and application-specific metrics. In the process, we make the following observations: (1) it is just as important to determine what sensors report and when they report as it is to transport these samples effectively; (2) a denser network is not always better if the sensors are not carefully managed; and (3) congestion is a problem that can harm network level performance as well as application level performance. As a final contribution, we present an outline of a general solution for handling congestion in sensor networks.

The broad outline of the thesis is as follows: Chapter 2 provides the introduction to the wireless micro-sensor networks and overviews related work. Chapter 3 describes the taxonomy of wireless micro-sensor network models. Chapter 4 describes the evaluation framework that's been developed and used for the experimental evaluation. Chapter 5 describes the infrastructure tradeoffs and presents the results obtained using the above framework. Chapter6 discusses various aspects of congestion control mechanism in sensor networks and presents theoretical model for Congestion Manager module. Finally Chapter 7 presents conclusions and future work. These chapters are over-viewed in some more detail in the remainder of this chapter.

Chapter 2 provides an introduction to the wireless micro-sensor networks. It defines the model of sensor networks upon which this thesis is based and discusses the real world applications for sensors. The main goal of this chapter is to provide the necessary background for the remainder of this thesis. This chapter also overviews related work.

Chapter 3 presents the taxonomy of sensor network models, which is a systematic classification of sensor network models along axes that influence communication. The taxonomy forms the theoretical background for this thesis. The taxonomy will aid in defining appropriate communication infrastructures for different sensor network application sub-spaces, allowing network designers to choose the protocol architecture that best matches the goals of their application. In addition, this taxonomy will enable new sensor network models to be defined for use in further research in this area. We hope that the taxonomy we have presented will be helpful in designing and evaluating future network protocols for wireless micro-sensor networks. We hope that this taxonomy will assist in developing relevant simulation models to enable empirical study of the performance of the different sensor network organizations and assist in making design and deployment decisions.

Chapter 4 describes evaluation framework thats been developed and used for protocol evaluation for sensor-networks. This chapter describes various design decisions behind the development of the framework. Subsequent chapters of this thesis use this framework to carry out the experimental studies for infrastructure tradeoff analysis and protocol development. The framework also allows sensor network designers to explore alternative design decisions and carry out what-if analysis.

Chapter 5 describes the study of protocol evaluation and infrastructure tradeoffs. It discusses the effect of infrastructure tradeoffs on the performance of a sensor network. First, we systematically increased the deployed sensor density and the required reporting rate and observed the performance of the network. When the offered load from the sensors to the network exceeded the capacity of the network, the performance starts dropping according to both network and application level metrics. Thus, by simply deploying more sensors, we may end up harming the performance of the network. This argues for intelligent management of the infrastructure by the network protocol: a form of congestion avoidance is needed that is significantly different from congestion avoidance in the data network sense. In particular, the network protocol must balance the offered load to the network against the required accuracy at the observer. This leads to the Congestion-Manager module discussed in the following chapter. Chapter 6 proposes a congestion control mechanism for sensor networks. Data delivery in sensor networks is not end-to-end, making traditional congestion management techniques inappropriate. The chapter outlines the design of a congestion management module for sensor networks. It classifies the different congestion control mechanisms and discusses their merits and demerits. Finally, Chapter 7 presents conclusions and future work that we intend to pursue.

Chapter 2

Background – Sensor Network Overview

This chapter presents introduction to wireless micro-sensor networks. It defines the sensor network and discusses the characteristics of a typical sensor network. Then we describe hardware organization of a typical micro-sensor network.In the nest section we focus on the a real world application of a sensor network.. In the end we consider a case study of an existing micro sensor architecture.

2.1 Introduction

Advances in hardware and wireless network technologies have placed us at the doorstep of a new era where small wireless devices will provide access to information anytime, anywhere as well as actively participate in creating smart environments. One of the applications of smart spaces is *a sensor network*; network that is formed when a set of small-untethered sensor devices those are deployed in an ad hoc fashion cooperate on sensing a physical phenomenon. Sensor networks hold the promise of revolutionizing sensing in a wide range of application domains because of their reliability, accuracy, flexibility, cost-effectiveness, and ease of deployment.

To motivate the challenges in designing sensor networks, consider the following scenarios: sensors are rapidly deployed in a remote inhospitable area for a surveillance application; sensors are used to analyze the motion of a tornado; sensors are deployed in a forest for fire detection; sensors are attached to taxi cabs in a large metropolitan area to study the traffic conditions and plan routes effectively; and smart Kindergarten [40] where sensor networks are deployed to create a developmental problem-solving environment for early childhood education. Clearly, there is a wide range of applications for sensor networks with differing requirements.

2.2 Sensor Network Characteristics

In this paper, we use the following terminology:

- *Sensor*: The device that implements the physical sensing of environmental phenomena and reporting of measurements (through wireless communication). Typically, it consists of five components– sensing hardware (transducer), memory, a battery, an embedded processor, and a transceiver.
- *Observer*: The end user interested in obtaining information disseminated by the sensor network about the phenomenon. The observer may indicate *interests* (or queries) to the network and receive responses to these queries. Multiple observers may exist in a sensor network.

- *Phenomenon*: The entity of interest to the observer that is being sensed and potentially analyzed/filtered by the sensor network. Multiple phenomena may be under observation concurrently in the same network.

In a sensing application, the observer is interested in monitoring the behavior of the phenomenon under some specified performance requirements (e.g., accuracy or delay). In a typical sensor network, the individual sensors sample local values (*measurements*) and disseminate information as needed to other sensors and eventually to the observer. The measurements taken by the sensors are discrete samples of the physical phenomenon subject to individual sensor measurement accuracy as well as location with respect to the phenomenon.

2.3 Sensor Hardware Organization

This section describes the hardware components of a typical micro-sensor. A micro-sensor typically consists of following five components namely

1. Transducer

2. Microprocessor

3. Memory

4. Transceiver (radio)

5. Battery

Lets us consider the use of each of these components one by one.

- Transducer: A transducer is an electronic device, which converts energy from one form to another. Common examples include microphones, loudspeakers, thermometers, position and pressure sensors, and antenna. Transducer enables the sensor to collect information about the phenomenon from the environment in one domain and represent it in another domain. Transducer accuracy affects the collective accuracy of the network for a given reporting frequency.

- Microprocessor: Each micro-sensor runs the specialized application on top of this microprocessor. Due to cost, power and form-factor considerations, a small microprocessor with limited computational power is typically used. Thus, it is not realistic to expect the sensors to carry out sophisticated computation.

- Memory - Memory provides the space for the computation. The amount of memory present on a sensor node affects the amount of buffering it can carry out. Memory is also a constrained resource for reasons of cost, form factor and power.

- Transceiver - After collection the information about the phenomenon the sensor needs to communicate this information to the observer. The transmission range of sensor network radios is limited for power considerations. Also, the transmission bandwidth is likely to be more limited than wireless data networks.

- Battery - Battery power is the primary resource on a sensor; it is required in order to support all the above mentioned hardware. Extending the battery life is often the most important consideration for sensor network protocols.

2.4 Real-world application

Let us now consider how the hardware components will interact by taking a real world example such as animal tracking [7]. Imagine a scenario where thousands of tiny micro-sensor devices are deployed to detect animals in a forest. Each sensor has a library or database of signals, where each signal in the library represents a signal corresponding to a waveform generated by a particular animal. If this signal database is not going to be updated in future then the database can be burnt on ROM else the sensor needs some secondary storage device such as a hard drive in a desktop. If an animal walks in, then the transducer in the sensor will detect/sense the animal and capture the waveform for that animal. It can then store the obtained signal in RAM and now can compare this signal with those, which are in the database to find any potential match. The application required to match the signals can be burnt on ROM as well. To run this application the sensor uses the given micro-processor. Once it finds match and figures out which animal is in range it can then transmit this information to other sensors or to the observer using its transceiver.

2.5 Case-study: Microsensor architecture

This section is a case study of microsensor architecture. We will consider a micro sensor developed by Rockwell Scientific company [33]. The specifications can provide a user some more insight into the hardware organization of a typical micro sensor network.

Figures 2.1, 2.2, 2.3 are front-views of the processor, the radio and the power supply modules in the above mentioned micro sensor.

Figure 2.1: Processor board front-view, courtsey Rockwellscientific.com

Table 2.1: Hardware specifications, courtsey Rockwellscientific.com

Package	
External Dimensions	$2.75inx2.625inx3.5in$
Internal Stack of 5 boards	$2.25inx2.25in$
Processor Module	
Processor	Intel StrongARM 1100 @133 MHz, 150 MIPS
Power Dissipation	Max: $< 300mW$, Typical: $< 200mW$, Idle:$< 40mW$, Sleep: $< 0.8mW$
Memory	1 MB SRAM, 4 MB FLASH memory
GPIO	26 lines
Radio Interface	3 wire $RS - 232$
Sensor Interface	4 wire SPI and USB
External Interface	JTAG, USB, and $RS - 232$
Integrated Radio Module Modem	Conexant RDSSS9M spread spectrum
Data Rate	100 Kbps
Radiated RF Power	1 mW, 10 mW, 100 mW
Range	> 100 meters at 100 mW
Frequency	ISM band, 902 – 928 MH, divided into 40 channels
Controller	Embedded $65C02$ microcontroller with 32 KB SRAM and 1 MB bootable FLASH memory
Other	4 bit ADC for battery voltage monitoring
Power Supply Module	
Input Voltage	4 – 12 V
Output Voltages / max Current	1.5 V / 160 mA; 3.0 V / 20 mA; 3.3 V / 300 mA
Sensor Modules	
Seismic	Mark IV geophone
Acoustic	Knowles BL1785 microphone, 4 Hz - 2 KHz (in design stage)
Magnetometer	Honeywell $HMC1001$, sensitivity = 1 lb of iron at 6 feet (in prototype test)
Accelerometer, Temp, Pressure	20 KHz accelerometer bandwidth, combined with temperature and pressure sensor

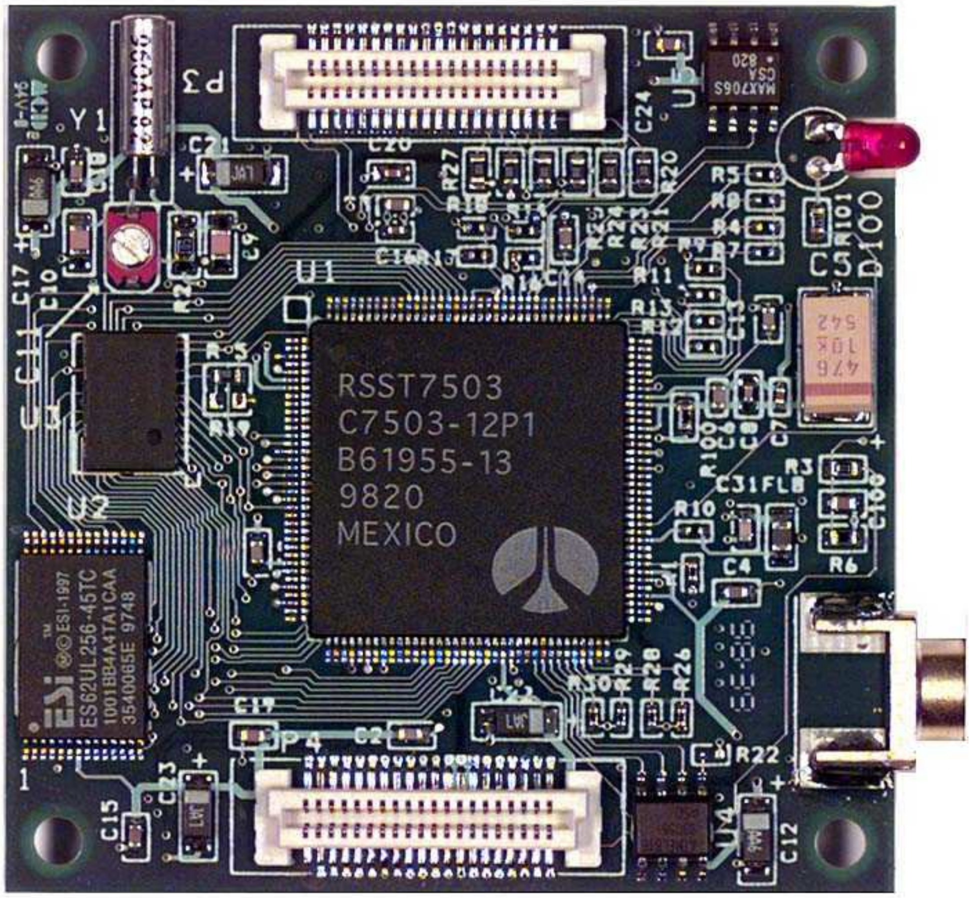


Figure 2.2: Radio front-view, courtsey Rockwellscientific.com

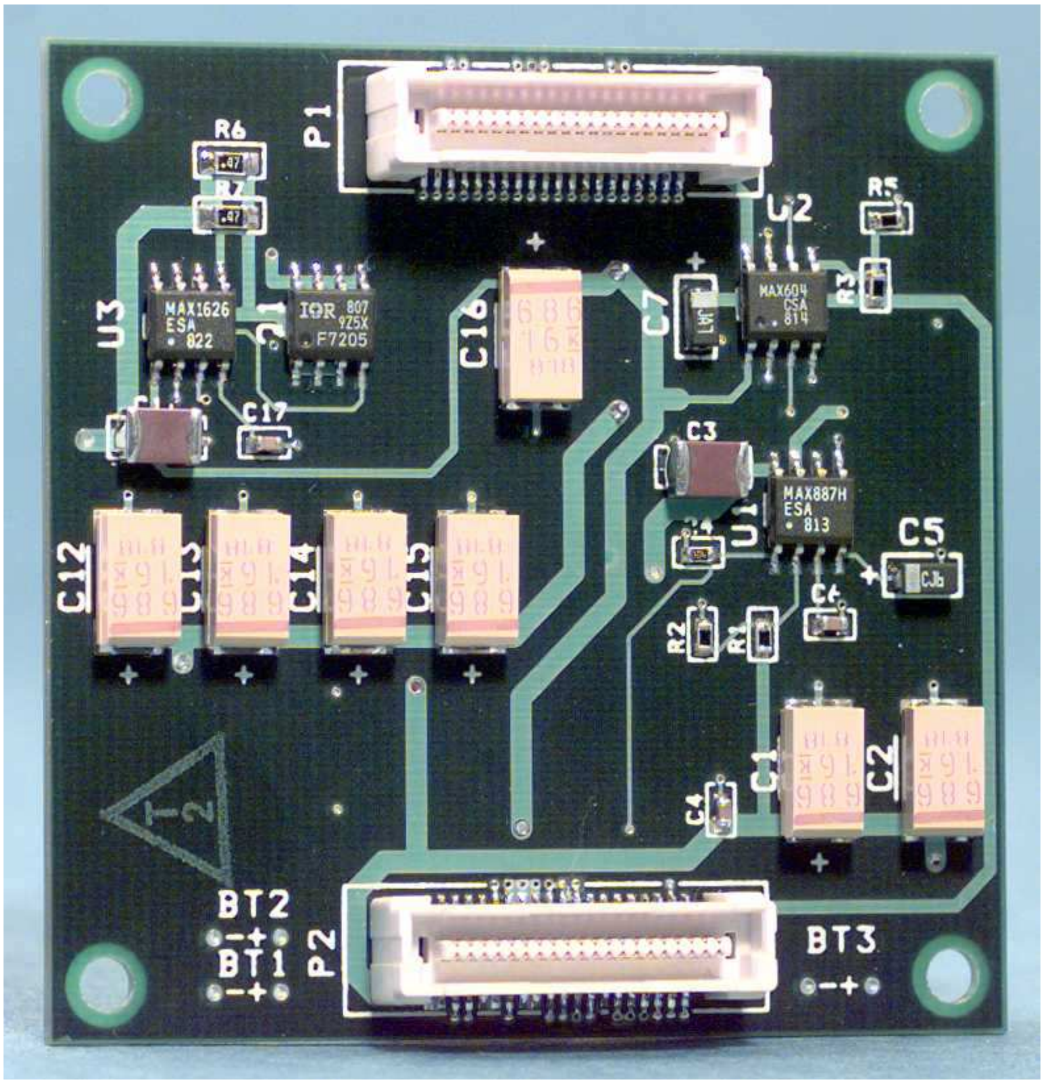


Figure 2.3: Powersupply front-view, courtsey Rockwellscientific.com

Chapter 3

A Taxonomy of Wireless Micro-Sensor Network Models

[1]

In future smart environments, wireless sensor networks will play a key role in sensing, collecting, and disseminating information about environmental phenomena. Sensing applications represent a new paradigm for network operation, one that has different goals from more traditional wireless networks. This chapter examines this emerging field to classify wireless micro-sensor networks according to different communication functions, data delivery models, and network dynamics. This taxonomy will aid in defining appropriate communication infrastructures for different sensor network application sub-spaces, allowing network designers to choose the protocol architecture that best matches the goals of their application. In addition, this taxonomy will enable new sensor network models to be defined for use in further research in this area.

[1] This chapter is based on an article published in ACM Mobile Computing and Communication Review [42]

3.1 Introduction

There is a wide range of applications for sensor networks with differing requirements. We believe that a better understanding of micro-sensor network requirements as well as the underlying differences between micro-sensor applications is needed to assist designers. To this end, in this study we attempt to classify wireless micro-sensor networks. In particular, we classify the aspects of wireless micro-sensor networks that we believe are most relevant to communication. We examine the characteristics and goals of typical micro-sensor networks as well as the different types of communication that are required to achieve these goals. We compare different data delivery models and network dynamics to create a taxonomy of wireless micro-sensor network communication. We believe that this taxonomy will aid network designers in making better decisions regarding the organization of the network, the network protocol and information dissemination models. Furthermore, it will aid in developing realistic sensor network models and benchmarks for use in future sensor network research.

The remainder of this chapter is organized as follows. Section 3.2 overviews performance metrics of interest for sensor networks. In Section 3.3, we describe sensor network architectures. Section 3.4 classifies the communication models present in sensor networks and makes the distinction between application and infrastructure related communication. Section 3.5 classifies the data delivery models. In Section 3.6, the network organization and dynamics are classified. Section 3.7 presents case studies of existing sensor network protocols, showing how they fit into the taxonomy described in this chapter. Finally, Section 3.8 presents a summary and some concluding remarks.

3.2 Performance Metrics

We propose using the following metrics to evaluate sensor network protocols.

- *Energy efficiency/system lifetime.* As sensor nodes are battery-operated, protocols must be energy-efficient to maximize system lifetime. System lifetime can be measured by generic parameters such as the time until half of the nodes die or by application-directed metrics, such as when the network stops providing the application with the desired information about the phenomena.
- *Latency.* The observer is interested in knowing about the phenomena within a given delay. The precise semantics of latency are application dependent.
- *Accuracy.* Obtaining accurate information is the primary objective of the observer, where accuracy is determined by the given application. There is a trade-off between accuracy, latency and energy efficiency. The given infrastructure should be adaptive so that the application obtains the desired accuracy and delay with minimal energy expenditure. For example, the application can either request more frequent data dissemination from the same sensor nodes or it can direct data dissemination from more sensor nodes with the same frequency.
- *Fault-tolerance*: Sensors may fail due to surrounding physical conditions or when their energy runs out. It may be difficult to replace existing sensors; the network must be fault-tolerant such that non-catastrophic failures are hidden from the application. Fault-tolerance may be achieved through data replication (e.g., the SPIN protocol [13]). However data replication itself requires energy; there is a trade-off between data replication and energy-efficiency. We suggest that the data replication

should be application-specific. The data which have higher priority according to the application might be replicated for fault tolerance and the other data might not be.

- *Scalability*: Scalability for sensor networks is also a critical factor. For large-scale networks, it is likely that localizing interactions through hierarchy and aggregation will be critical for ensuring scalability.

3.3 Sensor Network Architecture

A sensor network is a tool for measuring and relaying information about the phenomenon to the observer within the desired performance bound and deployment cost. As such, the organization of the network may be viewed as follows:

1. Infrastructure: The infrastructure consists of the sensors and their current deployment status. More specifically, the infrastructure is influenced by the characteristics of the sensors (e.g., sensing accuracy, memory size, battery life, transmission range) and deployment strategy (e.g., sensor density, sensor location, sensor mobility).

2. Network Protocol: The network protocol is responsible for creating paths and accomplishing communication between the sensors and the observer(s).

3. Application/Observer: The observer(s) interests in the phenomenon are queries from the observer(s) about the phenomenon as approximated by the distributed data that the sensors are capable of sensing. These queries could be static (the sensors are preprogrammed to report data according to a specific pattern) or dynamic. The network may participate in synthesizing the query (for example, by filtering some sensor data or fusing several measurements into one value); we consider such intelligence to be part of the translation process between observer interests and low-level implementation.

In this work, we focus on classifying issues that influence the second level: the network protocol. We discuss the other two levels only with regard to issues that influence communication. Thus, we do not address the difficult problem of translation between the observer query and the specific low-level interests. This translation could be done by the application software at the observer and/or the sensor nodes, or directly by a human observer. Similarly, we do not discuss the engineering of the infrastructure.

We also note that there is a significant opportunity for optimizations that cut across the three organizational levels. For example, Bhatnagar et al. discuss supporting QoS for sensor networks [3]. More specifically, they discuss discriminating among the type of data that the sensors are reporting and preferentially treating high priority data (for example, by giving it priority in forwarding and using redundancy to increase the chance of its reception). This is an example of an optimization where application-level knowledge provides hints to the network protocol. As another example, consider the case where the deployment of the sensors is chosen to mirror the expected motion pattern of the phenomenon or the interests of the observer. Such a deployment strategy incorporates application knowledge in the infrastructure design.

The network protocol in a sensor network is responsible for supporting all communication, both among sensor nodes as well as between the sensor nodes and the observer(s). The performance of the protocol will be highly influenced by the network dynamics, as well as by the specific data delivery model employed. In order to determine how the network protocol behaves for different scenarios, it is important to classify these features. In the following sections, we classify the different types of communication required in a sensor network and then look at the possible data delivery models and network dynamics.

3.4 Communication Models

There are multiple ways for a sensor network to achieve its accuracy and delay requirements; a well designed network meets these requirements while optimizing the sensor energy usage and providing fault tolerance. By studying the communication patterns systematically, the network designer will be able to choose the infrastructure and communication protocol that provide the best combination of performance, robustness, efficiency and deployment cost.

Conceptually, communication within a sensor network can be classified into two categories: *application* and *infrastructure*. The network protocol must support both these types of communication. Application communication relates to the transfer of sensed data (or information obtained from it) with the goal of informing the observer about the phenomena. Within application communication, there are two models: cooperative and non-cooperative. Under the cooperative sensor model, sensors communicate with other sensors to realize the observer interest. This communication is beyond the relay function needed for routing. For example, in a clustering protocol a cluster-head and the sensor nodes communicate with each other for information dissemination related to the actual phenomenon. In-network data processing [13, 10, 11] is an example of co-operative sensors. Non-cooperative sensors do not cooperate for information dissemination.

Infrastructure communication refers to the communication needed to configure, maintain and optimize operation. More specifically, because of the ad hoc nature of sensor networks, sensors must be able to discover paths to other sensors of interest to them and to the observer regardless of sensor mobility or failure. Thus, infrastructure communication is needed to keep the network functional, ensure robust operation in dynamic environments, as well as optimize overall performance. We note that such infrastructure communication is highly influenced by the application interests since the network must reconfigure itself to best satisfy these interests. As infrastructure communication represents the overhead of the protocol, it is important to minimize this communication while ensuring that the network can support efficient application communication.

In sensor networks, an initial phase of infrastructure communication is needed to set up the network. Furthermore, if the sensors are energy-constrained, there will be additional communication for reconfiguration. Similarly, if the sensors are mobile or the observer interests dynamic, additional communication is needed for path discovery/reconfiguration. For example, in a clustering protocol, infrastructure communication is required for the formation of clusters and cluster-head selection; under mobility or sensor failure, this communication must be repeated (periodically or upon detecting failure). Finally, infrastructure communication is used for network optimization. Consider the Frisbee model, where the set of active sensors follows a moving phenomenon to optimize energy efficiency [5]. In this case, the sensors wake up other sensors in the network using infrastructure communication.

Sensor networks require both application and infrastructure communication. The amount of required communication is highly influenced by the networking protocol used. Application communication is optimized by reporting measurements at the minimal rate that will satisfy the accuracy and delay requirements given sensor abilities and the quality of the paths between the sensors and the observer. The infrastructure communication is generated by the networking protocol in response to application requests or events in the network. Investing in infrastructure communication can reduce application traffic and optimize overall network operation.

3.5 Data Delivery Models

Ideally, the observer interest is specified in terms of the phenomenon, allowing the observer to be oblivious to the underlying sensor network infrastructure and protocol. The query is implemented as one or more specific low-level interests (e.g., requesting a specific sensor to report a specific measurement at some specific interval). Sensor networks can be classified in terms of the data delivery required by the application (observer) interest as: *continuous*, *event-driven*, *observer-initiated* and *hybrid*. These models govern the generation of the application traffic. In the continuous model, the sensors communicate their data continuously at a prespecified rate. The authors in [11] showed that clustering is most efficient for static networks where data is continuously transmitted. For dynamic sensor networks, depending upon the degree of mobility, clustering may be applicable as well. In the event-driven data model the sensors report information only if an event of interest occurs. In this case, the observer is interested only in the occurrence of a specific phenomenon or set of phenomena. In the observer-initiated (or request-reply) model, the sensors only report their results in response to an explicit request from the observer (either directly, or indirectly through other sensors). Finally, the three approaches can coexist in the same network; we refer to this model as the hybrid model.

Thus far, we have only discussed data delivery from the application perspective, and not the actual flow of data packets between the sensors and the observer; this is a routing problem subject to the network protocol. For any of the above-mentioned models, we can classify the routing approach as: flooding (broadcast-based), unicast, or multicast/other. Using a flooding approach, sensors broadcast their information to their neighbors, who rebroadcast this data until it reaches the observer. This approach incurs high overhead but is immune to dynamic changes in the topology of the network. Research has been conducted on techniques such as data aggregation that can be used to reduce the overhead of the broadcast [17, 13, 11]. Alternatively, the sensors can either communicate to the observer directly (possibly using a multi-hop routing protocol) or communicate with a cluster-head using one-to-one unicast. Finally, in a multicast approach, sensors form application-directed groups and use multicast to communicate among group members. The observer could communicate with any member of the group to obtain the desired data. A major advantage of flooding or broadcast is the lack of a complex network layer protocol for routing, address and location management; existing sensor network efforts have mostly relied on this approach (e.g., [17, 13]). However, the overhead of a broadcasting approach may be prohibitive.

It is likely that the interaction between the data delivery model from the application and the routing model employed by the network protocol will significantly impact the performance of the network. Consider a scenario where a sensor network is deployed for intrusion detection. In this case, the data delivery model is event driven – the event being an intruder entering the area. If the network level routing model is flooding based, in such a case physically co-located sensors will in general sense the intruder at the same time and try to send data to the observer simultaneously. These concurrent communications in the neighborhood will contend with each other for the use of the communication medium, raising: (1) the probability of loss of critical information; and (2) the latency in event reporting. A similar problem is studied by Woo and Culler [43].

3.6 Network Dynamics Models

A sensor network forms a path between the phenomenon and the observer. The goal of the sensor network protocol is to create and maintain this path (or multiple paths) under dynamic conditions while meeting the application requirements of low energy, low latency, high accuracy, and fault tolerance. Without loss of generality, this discussion assumes a single observer. Multiple observers can be supported as multiple instances of a single observer. More sophisticated protocols could also capitalize on the presence of multiple observers to merge related interests and/or optimize communication.

The problem of setting up paths for information dissemination is similar to the problem of routing in ad hoc networks [18]. However, there are a few critical differences, including: (i) the sensors are not generally addressed individually; rather, the interest is in the set of sensors that are *in a position to contribute to the active observer interests.* The sensors could be addressed by attributes of the sensor (e.g., their capabilities) and/or the phenomenon (e.g., the sensors close to a lion in a habitat monitoring scenario). The mapping between the observer interest and a set of sensors is influenced by the network dynamics and the application; and (ii) nodes along the path can take an active role in the information dissemination and processing. In this respect, sensor networks are similar to Active Networks [41] whereas ad hoc networks are traditional "passive" networks.

There are several approaches to construct and maintain a path between observer and phenomenon. These will differ depending on the network dynamics, which we classify as: *static sensor networks* and *mobile sensor networks.* We focus on mobility because it is the most common source of dynamic conditions; other sources include sensor failure and changes in observer interests.

Static Sensor Networks

In static sensor networks, there is no motion among communicating sensors, the observer and the phenomenon. An example is a group of sensors spread for temperature sensing. For these types of sensor networks, previous studies have shown that localized algorithms can be used in an effective way [17, 11]. The sensors in localized algorithms communicate with nodes in their locality. An elected node relays a summary of the local observations to the observer, perhaps through one on more levels of hierarchy. Such algorithms extend the lifetime of the sensor network because they trade-off local computation for communication [11]. In this type of network, sensor nodes require an initial set-up infrastructure communication to create the path between the observer and the sensors with the remaining traffic exclusively application communication[2].

Dynamic Sensor Networks

In dynamic sensor networks, either the sensors themselves, the observer, or the phenomenon are mobile. Whenever any of the sensors associated with the current path from the observer to the phenomenon moves, the path may fail. In this case, either the observer or the concerned sensor must take the initiative to rebuild a new path. During initial set-up, the observer can build multiple paths between itself and the phenomenon and cache them, choosing the one that is the most beneficial at that time as the current path. If the path fails, another of the cached paths can be used. If all the cached paths are invalid then the observer must rebuild new paths. This observer-initiated approach is a *reactive* approach, where path recovery action is only taken after observing a broken path.

[2]Note that if energy is limited among the nodes, the network will require infrastructure communication to maintain a path between the observer and the phenomenon as nodes run out of energy.

Another model for rebuilding new paths from the observer to the phenomenon is a sensor-initiated approach. In a sensor-initiated path recovery procedure, path recovery is initiated by a sensor that is currently part of the logical path between the observer and the phenomenon and is planning to move out of range. The sensor might perform some local patching procedure to build a new path by broadcasting a *participation request* for a given logical flow to all its neighboring sensors. Any one of the neighboring sensors can send a *participation reply* message to the given initiator sensor indicating willingness to participate and become a part of the requested path. If none of the neighboring sensors respond, the sensor can default to sending a path invalidation request to the observer so that the observer can start building the path. This is similar to soft hand-off in traditional Mobile IP based networks [15]. This sensor-initiated approach is a *proactive* approach where path recovery operations are begun in anticipation of a future broken path.

Dynamic sensor networks can be further classified by considering the motion of the components. This motion is important from the communications perspective since the degree and type of communication is dependent on network dynamics. We believe that each of the following require different infrastructures, data delivery models, and protocols:

- *Mobile observer.* In this case the observer is mobile with respect to the sensors and phenomena. An example of this paradigm is sensors deployed in an inhospitable area for environment monitoring. For example, a plane might fly over a field periodically to collect information from a sensor network. Thus the observer, in the plane, is moving relative to the sensors and phenomena on the ground.
- *Mobile sensors.* In this case, the sensors are moving with respect to each other and the observer. For example, consider traffic monitoring implemented by attaching sensors to taxis. As the taxis move, the attached sensors continuously communicate

with each other about their own observations of the traffic conditions. If the sensors are co-operative, the communication paradigm imposes additional constraints such as detecting the link layer addresses of the neighbors and constructing localization and information dissemination structures. From previous work [17], we know that the overhead of maintaining a globally unique sensor ID in a hierarchical fashion like an IP address is expensive and not needed. Instead, these sensors should communicate only with their neighbors with the link layer MAC address. In such networks, the above-mentioned proactive algorithm with local patching for repairing a path can be used so that the information about the phenomenon is always available to the observer regardless of the mobility of the individual sensors.

- *Mobile phenomena.* In this case, the phenomenon itself is moving. A typical example of this paradigm is sensors deployed for animal detection. In this case the infrastructure level communication should be event-driven. Depending on the density of the phenomena, it will be inefficient if all the sensor nodes are active all the time. Only the sensors in the vicinity of the mobile phenomenon need to be active. The number of active sensors in the vicinity of the phenomenon can be determined by application specific goals such as accuracy, latency, and energy efficiency. A model that is well-suited to this case is the Frisbee model [5].

It is important to note that the effect of mobility in sensor networks is fundamentally different than that in traditional wireless networks. Mobility in ad hoc networks has been addressed from the point of view of mobility of one or more of the communicating nodes during communication. However, since the sensors themselves are of no interest to the observer, their mobility is not necessarily of interest; rather, the sensor network must adapt its operation to continue to reflect the observer interests in the presence of mobility. Thus, the mobility of the sensing nodes themselves should be handled in a different way than for ad hoc networks; for example, a node that is moving away from a phenomenon may choose to hand-off the responsibility of monitoring to a closer node as it drifts away.

3.7 Case Studies and Related Work

In this section we consider several existing protocols for sensor networks and analyze them in the context of our taxonomy.

Ad hoc routing protocols may be used as the network protocol for sensor networks. However, such protocols will generally not be good candidates for sensor networks because of the following reasons: (i) sensors have low battery power and low available memory; (ii) the routing table size scales with the network size; (iii) these networks are designed for end to end communication and react inappropriately to mobility; (iv) their addressing requirements may be inappropriate for sensor networks [10]; and (v) ad hoc routing protocols do not support cooperative dissemination. More specifically, multihop routing protocols such DSR [19] and AODV [27] support the creation and maintenance of paths to route packets from source to destination. Sensor network studies have shown that application specific in-network data processing is essential to maximize the performance of the sensor-network [10, 11]. As ad hoc routing protocols do not inherently support data aggregation or fusion, they will not perform well in sensor network applications.

From an operational perspective, it is interesting to see the parallel between ad hoc routing protocol and the sensor network taxonomy. It appears that proactive protocols such as DSDV [28] are more appropriate to continuous data delivery since they proactively maintain paths throughout the network. In fact, one can think of the link state update function in these protocols as a form of continuous data delivery. Similarly, reactive protocols such as DSR [19] appear better suited for event-driven or query based information dissemination. In addition, a similar distinction can be made based on the network dynamics: the more dynamic the network, the better the reactive appraoches.

LEACH is an energy efficient protocol for sensor networks designed for sensor networks with continuous data delivery mechanism and no mobility [11]. LEACH uses a clustering architecture where member nodes send their data to the local cluster-head. Cluster-heads aggregate the data from each sensor and then send this information to the observer node. LEACH uses rotation of the cluster-head in order to evenly distribute the energy load. Once clusters are formed, cluster members uses TDMA to communicate with the cluster-head. Thus LEACH is suitable for networks where every node has data to send at regular intervals. However, it needs to be extended for event-driven models as well as for mobile sensors.

Directed Diffusion (DD) is a data-centric protocol, where nodes are not addressed by their addresses but by the data they sense [17]. Data is named by attribute-value pairs. In directed diffusion the interest is expressed by observer nodes in term of a query which diffuses through the network using local interactions. Once a sensor node that satisfies the query (source node) is reached, that node starts transmitting data to the sink node, again using local interactions. The absence of a notion of a global id (e.g., IP address) makes directed diffusion efficient for networks with mobility as well. Directed diffusion is applicable for event-driven and query-driven networks as defined in our taxonomy. The localized interactions allow the protocol to scale to large networks; DD scales as a function of the number of active interests present in the network.

The Publish/Subscribe model has been proposed for mobile networks by Huang and Gracia-Molina [44]. In this model, communication is typically anonymous, inherently asynchronous and multicasting in nature. From an application perspective, it also appears that the publish/subscribe model captures the relationship between the observer and phenomenon for some applications. More specifically, this model has desirable properties from the perspective of sensor networks; since the communication is not end-to-end, anonymous communication with application-specific multicast group formation is a viable approach. From an implementation perspective, asynchronous communication helps to preserve energy and increase the life-time of the network.

Ratnasamy et al. [31] present an alternative classification of sensor networks based on the data dissemination model. They propose that data dissemination can be done in at least three ways: (1) external storage - pass all the data to the observer and let them process this information; (2) local storage - information about the event is stored locally by the sensors; and (3) data-centric storage - data is stored by name and queries are directed by that name to the corresponding sensor. Clearly, the choice of the model will influence the communication patterns within the network. We view this as an application level decision.

3.8 Conclusion

The overall communication behavior in a wireless micro-sensor network is application driven. We believe that it is useful to decouple the application communication used for information dissemination from the infrastructure communication used to configure and optimize the network. This separation will aid network designers in selecting the appropriate sensor network architecture that will best match the characteristics of the communication traffic of a given application. This will allow the network protocol to achieve the application-specific goals of energy-efficiency, low latency, and high accuracy in the sensing application.

We also believe that a sensor-initiated proactive path recovery approach with local patching will be beneficial in efficient information dissemination in wireless micro-sensor networks.

We plan to study the behavior of various communication protocols for the different application sub-spaces described in this chapter. This will be done through analysis and simulation to determine the advantages and disadvantages of existing approaches, such as DSR (Dynamic Source Routing) [14], directed diffusion [17], and LEACH [11]. We hope that the taxonomy we have presented will be helpful in designing and evaluating future network protocols for wireless micro-sensor networks.

Often, it is possible to implement a sensor network for a specific phenomenon in a number of different ways. Consider the problem of monitoring a tornado. One option would be to fly airplanes to sense the tornado (mobile phenomenon; mobile sensors; continuous data delivery). Another would be to have a sensor grid statically placed on the ground and report data as the tornado passes through (mobile phenomenon; static sensors; continuous data delivery). Yet another would be to release lightweight sensors into the tornado (static phenomenon; mobile sensors; continuous data delivery). The primary concern here is the ability of the sensor network to report the desired level of accuracy under latency constraints within an acceptable deployment cost. The accuracy is a function of the sensing technology of the sensors and their distance from the phenomenon. However, since the performance is measured at the observer end, it is also a function of the performance of the communication model. We hope that this taxonomy will assist in developing relevant simulation models to enable empirical study of the performance of the different sensor network organizations and assist in making design and deployment decisions.

Chapter 4

Evaluation Framework/Experimental Environment

This chapter describes the evaluation framework that has been developed and the experimental setup that has been used for studying protocols for sensor-networks. ns-2 is an event-driven "Network Simulator" in public domain. The framework extends ns-2 to simulate a continuous phenomenon (a phenomenon that exists continuously in both time and space) and a discrete phenomenon (a phenomenon that exists at discrete points in both time and space) . We have modeled a phenomenon, an observer and sensors as an extension to ns-2. Contrary to prior studies, the environment models a phenomenon explicitly to enable the evaluation of different infrastructure tradeoffs. This chapter is organized as follows: Section 'Need-for an evaluation tool' describes the need for the evaluation tool. Design Goals section describes the design decisions behind the framework architecture. Framework-internals section describes the framework internals and the last section describes the experimental setup used for infrastructure tradeoffs study.

4.1 Need for an evaluation tool

A sensor network is a collection of sensors that observe phenomena and relay measured observation to an observer: any study of sensor network design will invariably involve modeling of these components. We developed a framework within ns-2 because it is an open source project with a large and active user community.

Sensor-network is an emerging field, so there is a lot of research going on to develop efficient protocols for sensor networks. It would be natural to compare those protocols over a range of scenarios. To the best of our knowledge there is no evaluation tool available at this point in time, where these protocols can be compared with each other. [1] In order to further motivate importance of fair protocol comparison let us consider the following example: Consider a case, where sensor network designer wants to build a network for animal tracking. In that case there are various options available to him across different layers namely infrastructure, network and application. By infrastructure options we mean number and deployment strategy of sensors. By options at the network layer we mean availability of different networking protocols such as LEACH, Directed diffusion, DSR (regular ad-hoc protocols). At the application layer designer can specify options like data-delivery model (e.g., continuous data delivery, event-driven data delivery etc.) Also, the designer would specify performance metrics at the application layer such as lifetime, delay, accuracy etc.. . . .

[1] In this chapter terms framework and evaluation tool are used interchangeably

The designer would be interested in finding out what is the best infrastructure organization (number of sensors and deployment strategy) and which is the most suitable protocol for his application so that his requirements at the application layer are met. Thus we need a tool to carry out such what-if analysis. By what-if analysis we mean, the tool should be able to answer the questions like these:
If there is no mobility, data-delivery mechanism is continuous, what is the most suitable protocol when lifetime should be maximized ?
If there is no mobility, data-delivery mechanism is event-driven, how does LEACH and DSR perform with respect to each other in terms of lifetime and latency etc. ?
Answering such questions would be helpful in resolving many system level issues in a sensor-network.

Thus we decided to build a framework and use to evaluate some of the system level issues in a sensor-network. However, note that our aim was not to provide answers to all the system level issues is sensor networks but to build a framework that can help to answer these questions.

4.2 Design goals

Let us consider the design goals of the evaluation framework:

1. Realistic modeling - *The model should reflect the real behavior of the system and should no introduce any artificial problems in the system due to modeling limitations.* Realistic modeling of sensor network components and interaction between them is crucial for protocol evaluation. We strive to build a realistic model to study the system behavior. We are interested to study the components of sensor network and their interaction.

We have thus considered a typical sensor network to be composed of three components namely sensor themselves, phenomenon to be sensed and the observer because a real world sensor network will invariably have these three components present in it. *Separation of phenomenon gives us a very clean framework since we can now model phenomenon characteristics such as mobility independent of observer and sensors.* A phenomenon can be either a discrete phenomenon or a continuous phenomenon. An animal moving in a forest represents a discrete phenomenon because it is not present at all the time in the sensing range of all the sensors. Temperature sensing represents a continuous phenomenon since any sensor can report temperature in its surrounding at any point in time. Our infrastructure supports both the discrete phenomenon sensing as well as continuous phenomenon sensing. In addition, application specific goals such as accuracy, latency and lifetime can be measured realistically.

2. Parameterization - Main goal of the framework is protocol evaluation for sensor networks. So the user should be able to evaluate a new protocol for a given network over the same scenario and under the same circumstances by simply plug-in a new protocol. Our infrastructure provides the necessary mechanism, while leaving the actual policy decisions to the sensor network designer. We believe that the separation of policy versus mechanism is a crucial decision in the framework design. We support accuracy measurement and as a proof of concept. We have implemented a mechanism to study the accuracy and energy depletion tradeoffs. In the implementation we assume each sensor is reporting the coordinates of the phenomenon based on just its own readings. We have chosen a policy that supports non-uniform accurate sampling of the phenomenon by the sensors. We believe that such policy reflects the real world

situation. One mechanism we have implemented to support that policy is accuracy based on the relative distance between sensors and a phenomenon. This means sensors, which are closer to a phenomenon can sense the phenomenon more accurately compared to the sensors, which are relatively far from the phenomenon. Consider an example of animal tracking sensor network, the sensors which are say within 50 m can have a better knowledge of the phenomenon as compared to the once which are say 100 m away from the phenomenon assuming homogeneous sensors. However we also support uniformly accurate sampling of the phenomenon based on the user specified parameter, which means that all the sensor which can sense the phenomenon can report about the phenomenon with the same accuracy.

The main advantage of the frame work is neither policy nor mechanism is hard coded. The user can tailor the framework for its applications just by plugging-in its own protocol and other relevant parameters. To the best of our knowledge this is the first framework, which supports such flexibility. Let us consider the architecture of a sensor in detail from the simulation point of view. For each sensor we have tried to emulate the subcomponents namely, transducer, transceiver and battery power. We have also simulated a a few typical sensor-network applications, which are executed on top of a general-purpose microprocessor. We have also considered the restrictions on memory since more memory demands both increase in size and increase in battery power consumption. We have used the energy model provided by ns-2 to model battery power and we have used the same parameters such as initial batter power, transmit energy, receive energy similar to the one used by directed diffusion protocol, in ns-2.1b8a version.

To model the transducer states we have assumed that a transducer operates in two modes namely, sleep mode and active mode for power-efficiency. A sensor will periodically turn on its transducer to check the status of the phenomenon. For example in case of a discrete phenomenon, the a sensor will turn on its transducer at a time interval of every Δt seconds and check whether it can sense in its sensing range. If it can sense the animal, then it will report its reading to the observer using its transceiver. This model can support both continuous as well as event-driven/phenomenon-driven model. [2]. For an event-driven model absence of animal during an active state, which occurs periodically, will not result in reporting about the animal thus saving energy.

Actually we have implemented two possible alternatives for transducer functioning. First model is called *an active-sensor model, since a sensor pro-actively detects an 'animal in range event' and the second is called* a passive-sensor model, since the animal sends a dummy signal to the sensor about an event(its presence).In passive-sensor alternative the transducer is continuously in the active state and as soon as the animal walks into the sensing range the animal sends a dummy signal to the sensor and then the sensor starts reporting to the observer. In active-sensor model a sensor switches its transducer to active state periodically to check occurrence of an event.Active-sensor model has disadvantage that if an animal walks into the the range (if the event happens) in between two consecutive wake-up points (of the transducer), say at Δ and $\Delta(t+1)$ respectively, then the sensor will not be able to sense it till $\Delta(t+1)$ time. However, this is compensated by the savings in power. In the passive-sensor model even though there is no lag in animal detection it demands the transducer to be turned on continuously thus draining more power. In sensor networks power is a scare resource thus we have decided to opt the active-sensor model for all our experiments.

[2]In this thesis terms 'event-driven' and 'phenomenon-driven' are used interchangeably. In the context of this thesis both the terms mean the same thing since presence of phenomenon is an event and phenomenon drives the overall communication

Also we do not assume the global synchronization of the sensor clocks. This has a side effect of non-concurrent wake-up timings for sensors. This means even though an animal walks into region R at time $t1$, that will be sensed by some sensor among the group of sensors at time less than $\Delta(t+1)$ - $t1$ since the Δt and $\Delta(t+1)$ of the neighboring sensors are not synchronized. This is a truly distinct distributed systems application, where the system is getting benefited because of non-synchronized clocks. Since the clocks are non-synchronized this will result is non-concurrent transmissions resulting in fewer collisions. If clocks would have been synchronized, animal being detected by the all the sensors at the same time would have resulted in concurrent transmissions causing more collisions.

4.3 Framework Internals

Framework is coded as an extension to ns-2 simulator. The framework consists of c++ code and TCL code. Framework itself is mostly coded in C++. TCL provides the required front-end for its usage. Phenomenon, observer and sensor nodes are coded in C++. However, the intended user of the framework will use TCL interface to specify the relevant parameters. To elaborate this further, the user will specify initial setup parameters such as number of sensors, deployment of these sensors (in terms of sensor coordinates), mobility pattern, networking protocol, traffic pattern (such as continuous or event driven) etc. in a TCL script. By changing different parameters, fair comparison of factors such as different network protocols or different deployment strategies is possible. Complexity of the framework is totally transparent to the user.

Table 4.1: Parameters used in the simulation studies.

Simulation area	$800x800\ m^2$
Transmission range	$250\ m$
Startup-Energy	$10000\ J$
Discrete phenomenon sensing range	$200\ m$
Phenomenon speed	random between $1 - 2\ m/s$
MAC Protocol	802.11
Bandwidth	$2\ Mbps$
Transmit Power	$0.660\ W$
Receive Power	$0.395\ W$

4.4 Experimental Setup

The experimental setup consists of setting up the sensors themselves, observer and phenomenon. We have considered the following topologies:-

1. Uniformly distributed sensors
2. Randomly deployed sensors
   (a) Uniform density
   (b) Non-uniform density

In all the scenarios following parameters are constant:-

4.5 Experimental Study

Uniform deployment scheme:- In this scheme we consider a sensor network organized in grid like fashion. We have considered grids of four sizes namely $5x5grid$,$10x10grid$,$12x12grid$ and $15x15grid$. A $10x10grid$ means there are 98 sensors, 1 observer and 1 phenomenon (98+1+1 = 100). Each point in the grid represents a sensor. Such deployment is possible only in case where the physical environment is not inhospitable. The reason behind different grid sizes is to vary the density of the sensors per unit area to study infrastructure tradeoffs.

Random-deployment scheme:-

Uniform Density :- In this scheme sensors are randomly deployed in the given area. The density of sensors per unit area is more or less equal due to randomness. If the physical environment in inhospitable, then we can imagine a plane flying over the area and sensors are thrown out randomly. In order to make a fair comparison of random deployment with grid deployment in terms of energy, goodput, accuracy etc. we kept the number of sensor same. Thus for random deployment we consider 100, 144 and 225 sensors. With 25 sensors in random deployment, the network was too sparse for connectivity so for all random deployment scenarios we did not consider the case with 25 sensors. Also the phenomenon mobility is identical across all the scenarios. This was done to make a fair comparison.

Non-uniform Density :- If the sensor network designer has some idea about the behavior of the phenomenon such as mobility of the phenomenon, or if the designer is interested only in some specific parts of the animal life cycle then the designer can take advantage of the pre-knowledge and deploy the sensor-network accordingly. For example if the designer knows that a lion is generally present near the lake, then he can deploy more sensors for better coverage around the lake. We have also considered such biased deployment in our experimentation. For fair comparison we have kept the number of sensors same as that of above two cases. However in this we deploy the sensors with non-uniform density, thus creating few"hot-spot" areas in the network. The area where the density of sensors is considerably more than that of other subareas is considered as a "hot-spot". Here again the number of sensors was kept same as of the above cases and also the total area was the same '$800x800meters^2$, however the deployment was choses as follows:-

1. For 100 sensors - 25 sensors in $(0,0)-(400,400) rectangle$ and 75 sensors in $(400,400)-(800,800) rectangle$

2. For 144 sensors - 25 sensors in $(0,0)-(400,400) rectangle$ and 119 sensors in $(400,400)-(800,800) rectangle$

3. For 225 sensors - 25 sensors in $(0,0)-(400,400) rectangle$ and 200 sensors in $(400,400)-(800,800) rectangle$

following figures are the sample of topologies used for experimentation.

Figure 4.1 Figure 4.2 Figure 4.3 Figure 4.4 Figure 4.5 Figure 4.6 Figure 4.7 Figure 4.8 Figure 4.9

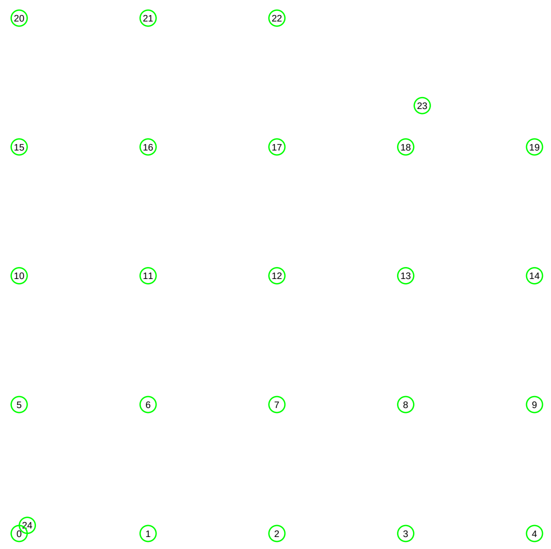


Figure 4.1: 5x5 Grid.

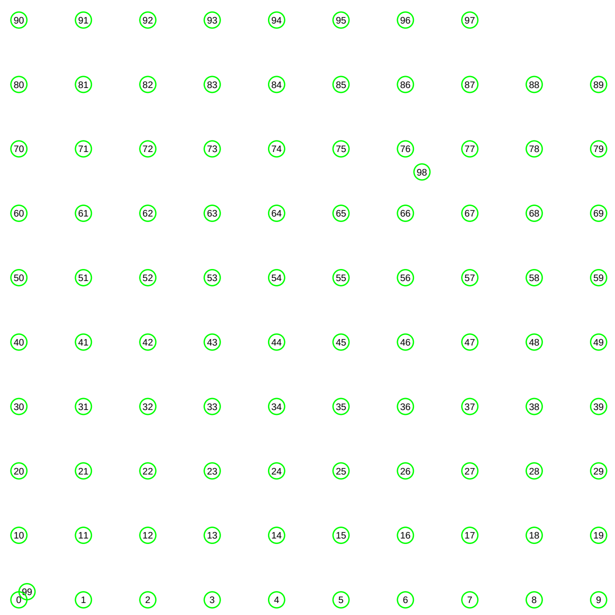


Figure 4.2: 10x10 Grid.

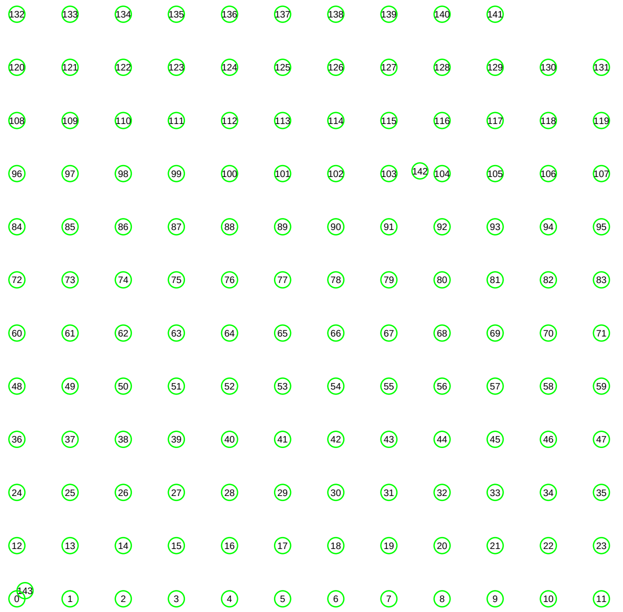


Figure 4.3: 12x12 Grid.

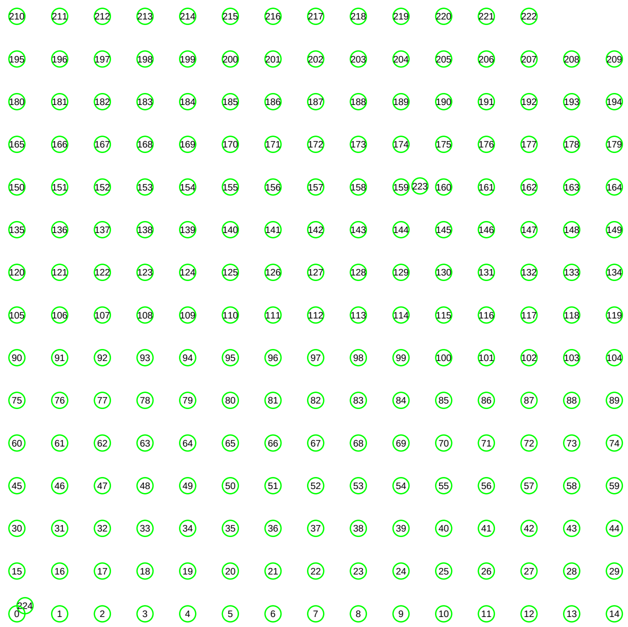


Figure 4.4: 15x15 Grid.

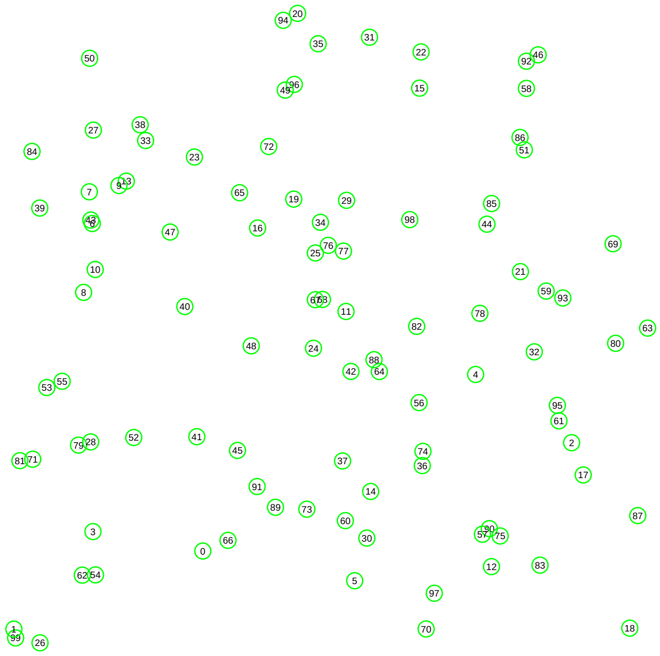


Figure 4.5: True Radnom deployment for 100 sensors

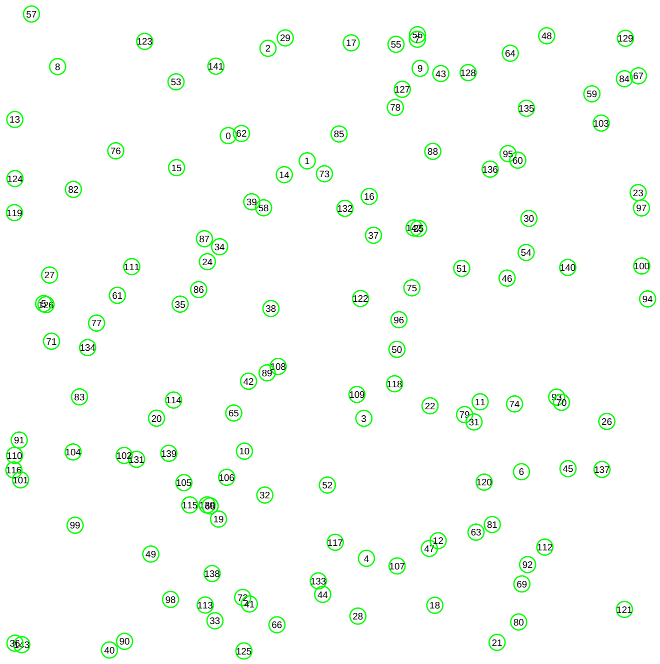


Figure 4.6: True Random deployment for 144 sensors

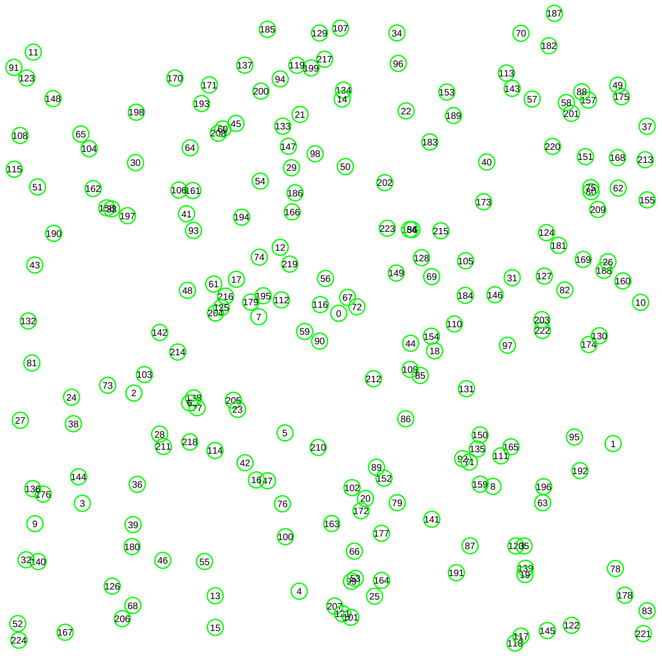


Figure 4.7: True Random deployment for 225 sensors

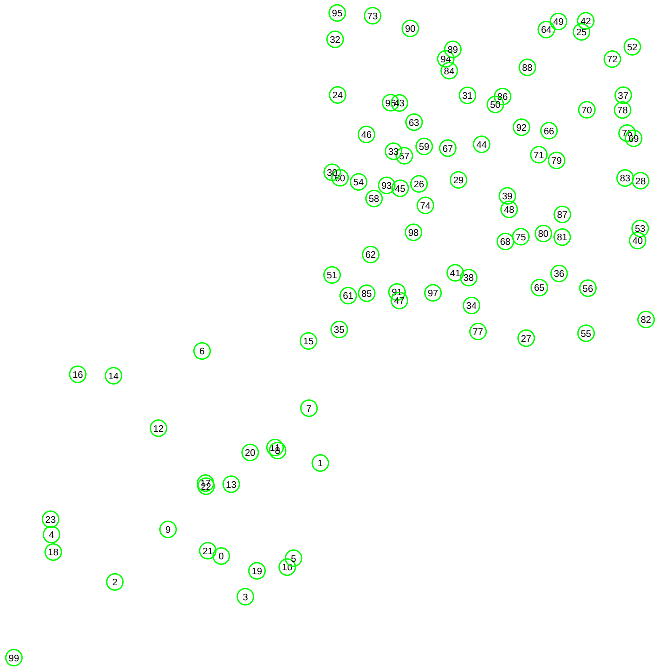


Figure 4.8: Biased Random deployment for 100 sensors

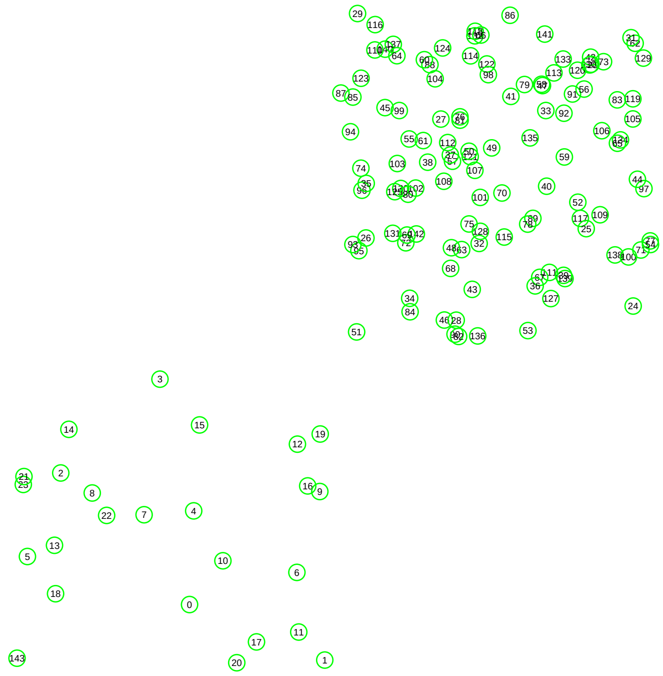


Figure 4.9: Biased Random deployment for 144 sensors

Chapter 5

Infrastructure Tradeoffs for Sensor Networks

[1]

In a sensor network, the infrastructure (in terms of the sensor capabilities, number of sensors, and deployment strategy) plays a significant role in determining the performance of the network. In this chapter, we study the effect of infrastructure decisions on the performance of a sensor network. We study the effect of the infrastructure for two types of network delivery models (phenomenon driven and continuous) and different network protocols (DSR, DSDV and AODV). We show the performance both in terms of network efficiency as well as meeting the application accuracy and latency demands. By exploring the criteria for effective infrastructure configurations, we open the door for network optimizations that control the effective topology to better achieve the application requirements.

[1] This chapter is based on an article published in ACM Workshop on Sensor Networks and Applicaitons

5.1 Introduction

Sensor networks represent a new paradigm for reliable environment monitoring and information collection. They hold the promise of revolutionizing sensing in a wide range of application domains because of their reliability, accuracy, flexibility, cost-effectiveness, and ease of deployment. Furthermore, in future smart environments, it is likely that sensor networks will play a key role in sensing, collecting, and disseminating information about the environment.

A sensor network is a tool for distributed sensing of one or more phenomena, and reporting the sensed data to one or more observers. As such, the performance of the network is best measured in terms of meeting the accuracy and delay requirements of the observer. Additional performance metrics include the life time of the network, cost of the sensors and their deployment, fault tolerance and scalability [42].

Conceptually, a sensor network is organized as a three layer system: (1) infrastructure: refers to the physical sensors (their physical characteristics and capabilities), the number of sensors and their deployment strategy (how/where they are deployed); (2) networking protocol: responsible for dissemination of the sensed data by creating and maintaining paths between the sensors and the observer(s); and (3) the application: responsible for translating the observer interests into specific network-level operations. Cross-cutting optimizations across the three levels are possible to improve the performance of the network.

Although there is a large body of work in building and networking sensors (a good bibliography of sensor network research can be found on this website [34]), these studies focus on optimizing the application and networking protocol to improve performance. In contrast, this study considers the tradeoffs in the infrastructure design and their implications on performance and the design of the networking protocol. We also study the effect of biasing the deployment to reflect the phenomenon motion pattern on the performance of the network.

Intuitively, it appears that a denser infrastructure leads to a more effective sensor network because higher accuracy is likely and a larger aggregate amount of energy is available in the network. However, if properly not managed, a denser network will lead to a larger number of collisions and potentially to congestion in the network; this will increase latency and reduce energy efficiency. Moreover, the large number of samples reported by the sensors may exceed the accuracy requirements of the observer. Thus, simply increasing the reporting rate or the number of sensors may actually harm the performance of the network. We study this tradeoff using different application scenarios (phenomenon driven vs. continuous update data reporting) and for different infrastructure configurations.

One of the lessons learned from this study is that a form of congestion control is necessary to make sure that the reported samples do not exceed the capacity of the network. In addition, this control is necessary to optimize the lifetime of the network while meeting the minimum accuracy requirements of the application. Thus, the congestion control must not only be based on the capacity of the network, but also on the accuracy level required at the observer. The traffic in a sensor network is different from conventional networks; it is a collective communication operation with redundancy. Thus, the network protocol has the flexibility of meeting the performance demands by controlling the reporting rate of the sensors, controlling the virtual topology of the network (by turning off some sensors for example), or optimizing the collective reduction communication operation (by fusing data along the way for example). We note that this application driven congestion control is different, and at a lower level, from proposals to incorporate application dependent processing and/or data aggregation within the network.

The remainder of this chapter is organized as follows. In Section 5.2 we overview the role of the infrastructure and discuss the available deployment strategies. Section 5.3 overviews the modeling approach and the evaluation environment. In Section 5.4 we present the experimental study. Section 5.5 overviews some related work. Finally, Section 5.6 presents some concluding remarks.

5.2 Infrastructure Features

The infrastructure of a sensor network refers to the characteristics of the individual sensors, the number of sensors deployed, and the deployment strategy. We will discuss each of these in turn.

5.2.1 Sensors' Capabilities

A sensor typically consists of five components: sensing hardware, memory, battery, embedded processor, and trans-receiver. These components affect the performance of the sensor and ultimately that of the network. For example, the accuracy of the sensing hardware or *transducer* will affect the accuracy of the sensing at the observer. Similarly, the size of the memory affects the buffering space at the sensors and the ability of the network to handle transient bursts in traffic. The battery size determines the amount of energy available at the sensor and affects the lifetime of the network. The capabilities of the embedded processor determine the level of optimization that is possible at the sensors without introducing excessive loss of power or intolerable levels of delay. Finally, the characteristics of the transreceiver determine the transmission range of the network and the capacity of the transmission channel. Improving the characteristics of any of these subsystems increases the cost, form factor or both for the sensor. Thus, within the available budget for the sensor network, the designer must decide whether to invest in a large number of inexpensive sensors, or a smaller number of expensive, higher quality ones.

5.2.2 Number of Sensors

Intuitively, for a given type of sensor, increasing the number of sensors deployed in the field should result in a better performing network with respect to the metrics identified earlier; otherwise, why pay the extra cost. Consider: (1) the accuracy of the sensing should improve since there are more sensors in a position to report on the phenomena; (2) the available energy within the network increases; and (3) the additional sensor density offers the potential for a better connected network with more efficient paths between the sensors and the observers. However, increasing the number of sensors in turn results in a higher number of sensors reporting their results per unit time. If this increased load exceeds the capacity of the network in terms of access to the shared wireless medium as well as congestion in intermediate nodes, increasing the number of active sensors may end up adversely affecting the performance of the network.

With respect to capacity, the problem can be viewed in terms of collision and congestion. To avoid collisions sensors that are in the transmission range of each other should not transmit simultaneously. Consider sensors $1 \ldots M$, each with transmission range r, that are arranged in a chain. For any given sensor S_i, sensors located in the range $loc(i) - r$ and $loc(i) + r$ should not transmit at the same time. Research by Woo et al. [43] has addressed some of the issues with the collision problem, trying to improve upon existing MAC layers.

To the best of our knowledge, congestion has not been addressed by past studies. We consider a phenomenon driven reporting model where a sensor reports if it is in range of the phenomenon. Assume that we have N sensors out of which M sensors are in range of the phenomenon at a given time T. Assume that the M sensors are in interference range with each other (e.g., the transmission range is greater than or equal to the sensing range). Of the M reporting sensors, each sensor $b(S_i)$ will transmit data toward the observer with bit rate b_i. The total data in transit from time T to $T + \delta$ where δ is the average latency can be expressed as

$$Data = \sum_{i=1}^{M} b(S_i) \tag{5.1}$$

If this value reaches a certain fraction of the channel capacity, congestion will occur [21]. If C_total is the total channel capacity then

$$\sum_{i=1}^{M} b(S_i) <= \alpha C_{total} \tag{5.2}$$

where α is a fraction of the capacity dictated by the self-interference that arises in multi-hop connections (α is typically around 0.25 [22]). Thus, the upper bound on the reporting rate is dictated by the channel capacity. On the other hand, application specific criteria such as the required accuracy places a lower bound on the reporting rate; the reporting rate should be high enough to satisfy the desired accuracy. At any point in time the number of active sensors should be such that the application specified accuracy requirements are met. If, in order to meet the accuracy requirements, $C_{application}$ is the required channel capacity then we have:

$$C_{application} <= \sum_{i=1}^{M} b(S_i) <= \alpha C_{total} \tag{5.3}$$

$$C_{application} <= \alpha C_{total} \tag{5.4}$$

to support the application requirements.

Note that not all sensors are equal in terms of accuracy: depending on the location, a specific sensor may have a higher quality data sample, or a combination of sensors may together provide a higher accuracy than another combination. However, we can qualitatively comment on the factors on which the number of active sensors depends. From a networking perspective, it depends on factors such as the geographic locations of the reporting sensors, buffer lengths, and packet processing times. From an application perspective, the value of information sensed by the sensor needs to be considered as well. As was discussed earlier, if a sensor is providing some unique information about some feature of the phenomenon, then the application might require that sensor to report irrespective of the location of that sensor. Thus, application level information must be used in determining what sensors to report and when to meet the application performance metrics. We intend to pursue such protocols in the future.

5.2.3 Deployment Strategies

Finally, it is important to consider the deployment strategy for the sensors (e.g., their distribution within the phenomena field). We consider three deployment strategies: (1) random deployment – the sensors are "sprayed" with a uniform distribution within the field; (2) regular deployment – the sensors are placed with some regular geometric topology in the sensor field (for example, a grid); and (3) planned deployment – sensor deployment is planned (for example, biased to provide higher sensor density in areas where the phenomenon is concentrated). It is unclear whether regular deployment will offer advantages over uniformly distributed random deployment; if it does not, random deployment is preferable because of its low cost.

In the remainder of this chapter, we will evaluate these infrastructure tradeoffs for two types of monitoring disciplines (phenomenon driven and continuous reporting), and different routing protocols. The evaluation environment and modeling approach are presented in the next section.

5.3 Evaluation Environment

In order to model the complex relationships described above, we have developed an evaluation environment within the NS-2 simulator [2]. Contrary to most sensor network studies, we have made the phenomenon explicit and decoupled it from the sensor network organization. This allows us to study the effect of varying the design within the sensor network using scenarios that are independent of it. We model two types of phenomena: (1) discrete phenomena (for example, animals in a habitat monitoring application [5]); and (2) continuous phenomena (for example, the temperature in a temperature tracking application). For each of these types of phenomenon, the sensors wake up periodically according to some user defined schedule, take samples of the phenomenon and report their results if required by the application.

To model the transducer states, we have assumed that a transducer operates in two modes: sleep and active to model the low duty cycle necessary for power efficient operation. The transducer periodically wakes up and enters the active state to check the status of the phenomenon. For example, in the case of a discrete phenomenon, the transducer will wake up at a time interval of every δt seconds and check whether it can sense the phenomena, (e.g.,an animal) in its range. If it can then it will report the reading to the observer using its transceiver. This model can support both continuous as well as phenomenon driven model. For a phenomenon driven model absence of animal during periodic active state will not result in reporting about the animal and the sensor will go back to the sleep state.

The environment also decouples the three levels of the sensor network: infrastructure, protocol and application. The reasoning again is to provide a vehicle to allow comparison of "apples to apples"; we can study the effect of varying the design at each of these levels on the performance of the network under uniform assumptions. For example, we can study the effect of changing the network protocol for a given application and infrastructure. In this chapter, we study the effect of varying the infrastructure on the performance of the network for different applications and network protocols.

In this work, we considered an application with a discrete phenomenon that moves around in a square grid (e.g., animal tracking) as well as an application with a continuous phenomenon that can always be sensed (e.g., temperature sensing). We also considered two application level scenarios: (1) continuous update: the sensors periodically report their local measurement to the observer; and (2) phenomenon driven: sensors report their measurements to the observer periodically, but only if they have data of interest to report (in this case, the discrete phenomenon is within detection range). Other scenarios can be easily constructed; for example, scenarios with multiple phenomena or multiple observers can be directly generated.

We are interested in application-level performance; conventional network performance metrics such as throughput are of secondary interest. We consider the following performance metrics.

1. Accuracy: The accuracy of a measurement at a sensor is specific to the physical transducer and the nature of the phenomenon. In general, we assume that the measurement has a tolerance that increases with the distance between the sensor and the phenomenon. At the observer, it is likely that multiple samples will be received from the different sensors. These samples must be combined intelligently to produce a more accurate estimate of the location of the phenomenon. It is possible to bias the estimate towards sensors with higher confidence (closer to the phenomenon) and towards more recent samples.

2. Latency: Latency refers to the delays in obtaining the samples at the observer due to network congestion, the duty cycle of the sensors, or intelligent filtering of sampled data. For real time sensing applications, delays in reporting the state of the phenomenon leads to a loss in accuracy. For the purposes of this study, we report only the packet latency within the network.

3. Energy efficiency and fault tolerance: the energy efficiency of the network may be measured in different ways. For now, we report the energy expenditure within the network.

4. Goodput: Goodput is the ratio of the total number of packets received by the observer to the total number of packets sent by all the sensors over the simulation time.

5. Scalability is also of interest. While we do not investigate scalability directly, efficient data reporting and reducing network load is conducive to scalability.

5.4 Experimental Study

We considered a scenario where a discrete phenomenon is being tracked by sensors placed in a square grid of dimensions 800 meters by 800 meters. We assumed that each sensor data sample has a uniformly distributed tolerance of +/- 5% of the actual distance between the sensor and the phenomenon. In the phenomenon driven scenarios, only the sensors within a discrete phenomenon sensing range report their estimate of the location of the phenomenon to the observer. The packet size was fixed at 100 bytes unless specified. We used the energy model from the Directed Diffusion sensor network protocol study [17]. The buffer space available at each sensor is of size 5 packets; a larger buffer size will enable the network to withstand a higher level of transient congestion but will not help with sustained overloading of the network. From its initial position, the phenomenon walks towards a random destination with a speed randomly chosen between 1 m/s and 2 m/s.

The parameters used in our simulations are summarized in Table 4.1. Changing these parameters will have an effect on the capacity of the network and the offered load, but due to space limitations this effect is not pursued.

Figures 4.4, 4.5 and 4.8 show a sample of the topologies that were used for the experiments. Figure 4.4 shows a 15x15 grid topology, which includes 223 sensor nodes, 1 observer node and 1 phenomenon node. Figure 4.5 shows a network with 100 sensors randomly distributed. Figure 4.8 shows a biased network, where the designer has some idea about the phenomenon mobility. In this case, if the designer knows that the phenomenon moves in general in upper right corner (hot-spot), then sensors can be deployed with non-uniform density so that more sensors are located in the hot-spot region.

We first establish the basic infrastructure configuration tradeoffs using the following

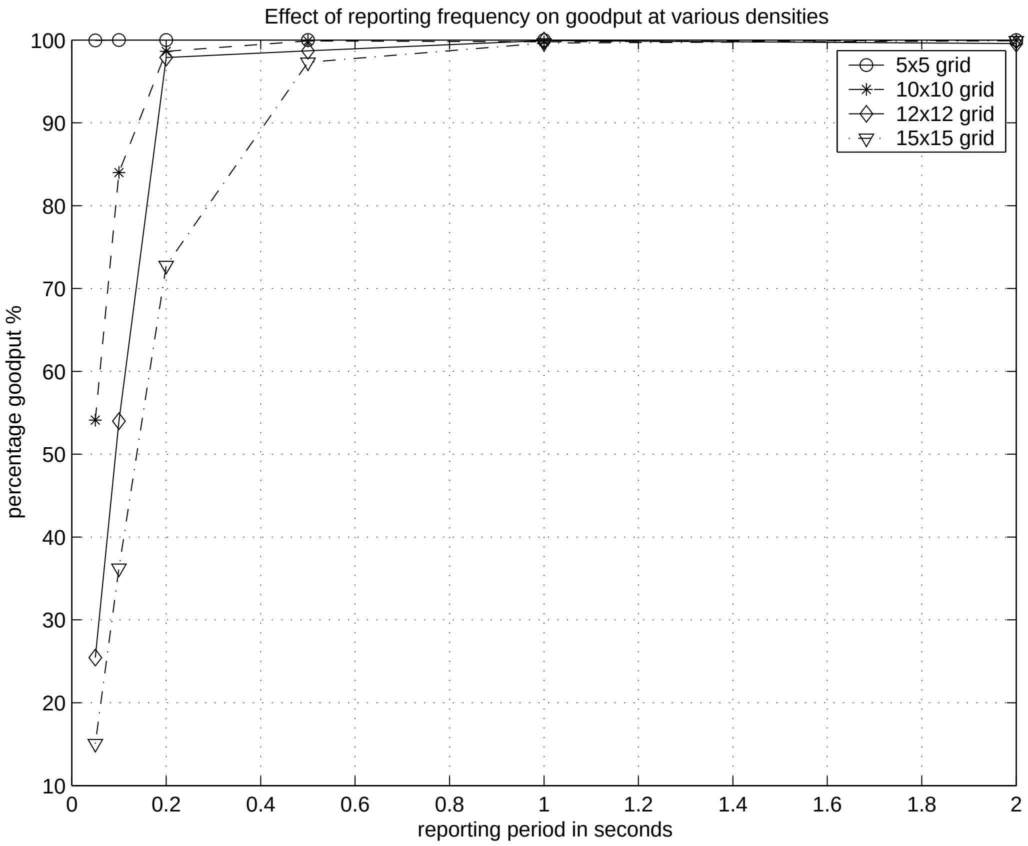


Figure 5.1: Goodput as a function of network density and sensor reporting period (grid deployment).

parameters. We used Dynamic Source Routing (DSR) [19] as the networking protocol and explore both grid deployment and random deployment of the sensors. Finally, we explore biasing the deployment pattern to match the phenomenon motion pattern. Each simulation was run for 50 seconds, and every point represents the average of three different random seeds. Unless stated otherwise, the data deliver model was phenomenon driven.

5.4.1 Basic Infrastructure Tradeoffs

Goodput and Delay Study

In the first set of experiments, we study the effect of increasing the sensor density on the efficiency of the network. Figure 5.1 shows the goodput of the network as a function of the

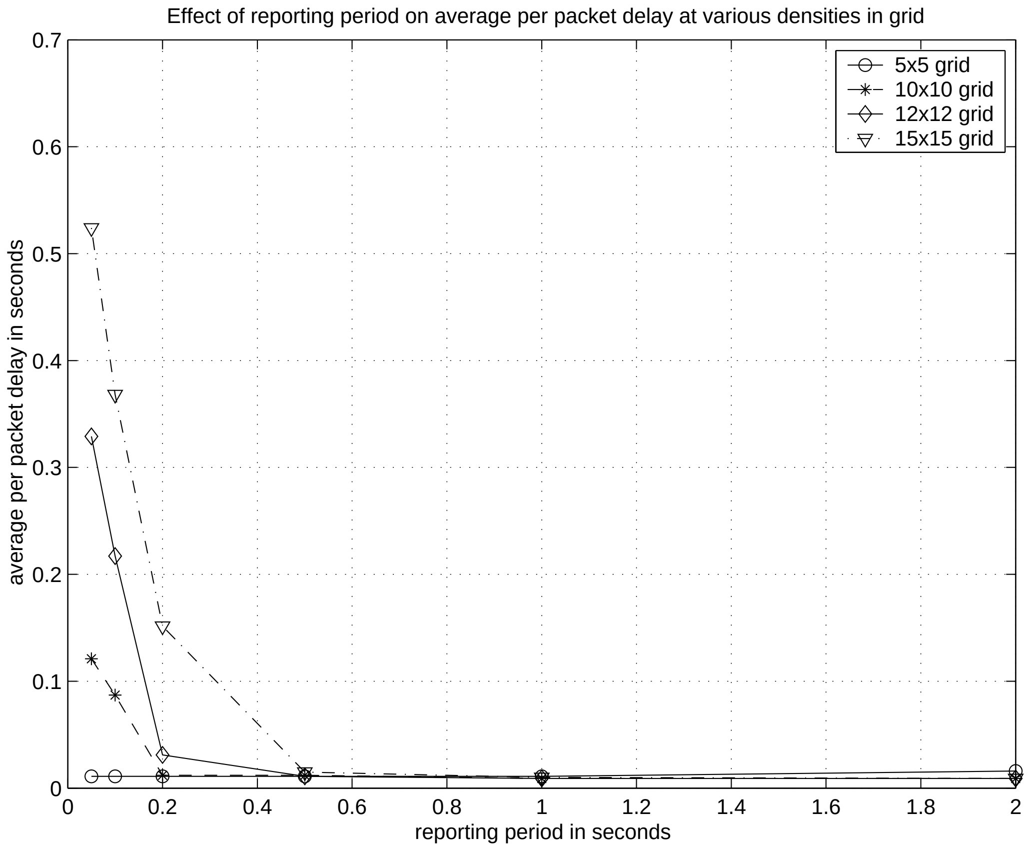


Figure 5.2: Delay as a function of network density and sensor reporting period (grid deployment).

reporting period for several levels of network density. The deployment strategy was regular; sensors were placed in square grids with the stated number of sensors per side. We first note that as the data rate increases (reporting period decreases), the goodput drops when the rate exceeds the capacity of the network and sensed packets start to be dropped. It is interesting to note that the drop in goodput is more pronounced for the denser networks. This is due to the larger number of sensors close to the phenomenon effectively increasing the offered load to the network, resulting in more collisions and a higher number of packets dropped due to congestion. This effect is corroborated by the packet latency results (Figure 5.2): the latency increases with the data rate as well as the density of the network.

We repeated these experiments for random deployment, keeping the same number of

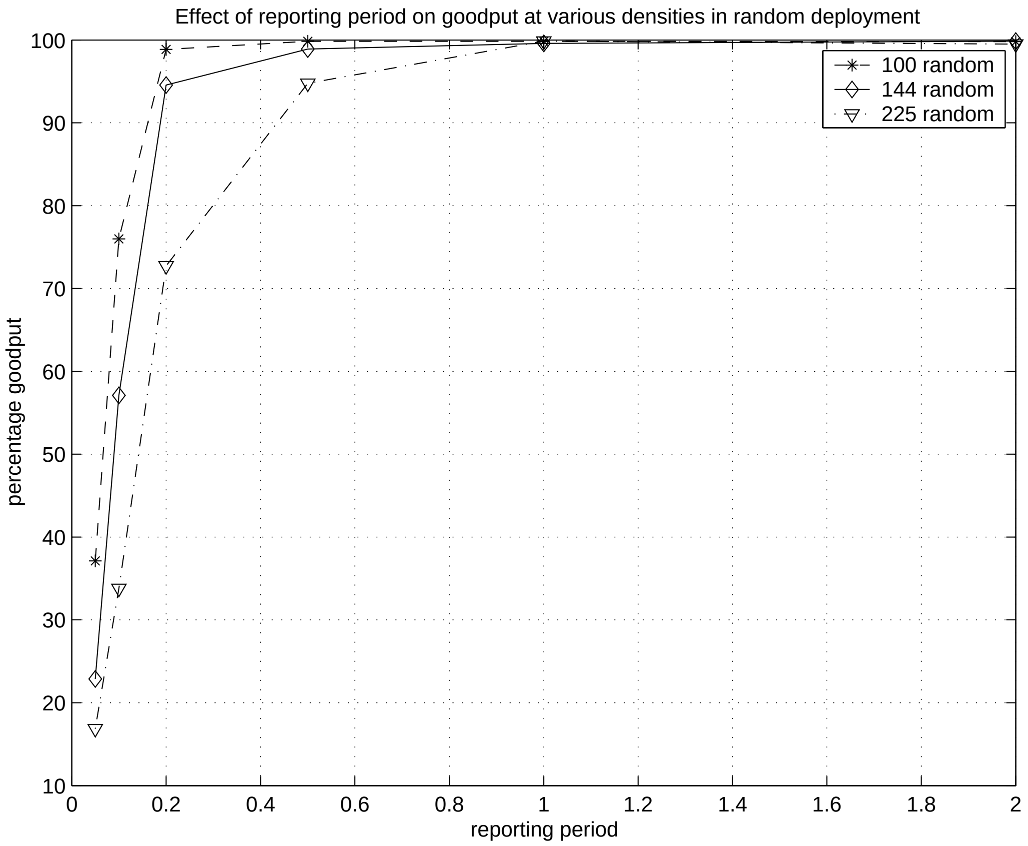


Figure 5.3: Goodput as a function of network density and sensor reporting period (random deployment).

sensors as each of the studied grids. The results for goodput (Figure 5.3) and delay (Figure 5.4) do not show appreciable differences in comparison to grid deployment. Note that we do not consider the scenario with 25 sensors, as was done in the grid case, because the network was too sparse to maintain connectivity with random deployment.

Accuracy Study

In terms of application performance, we measured the accuracy of the tracking of the phenomenon position. More specifically, the observer generates an estimate of the phenomenon location based on the samples it receives from the sensors. We measured the error in these samples in the following way. We discretized time into small slots and averaged the sam-

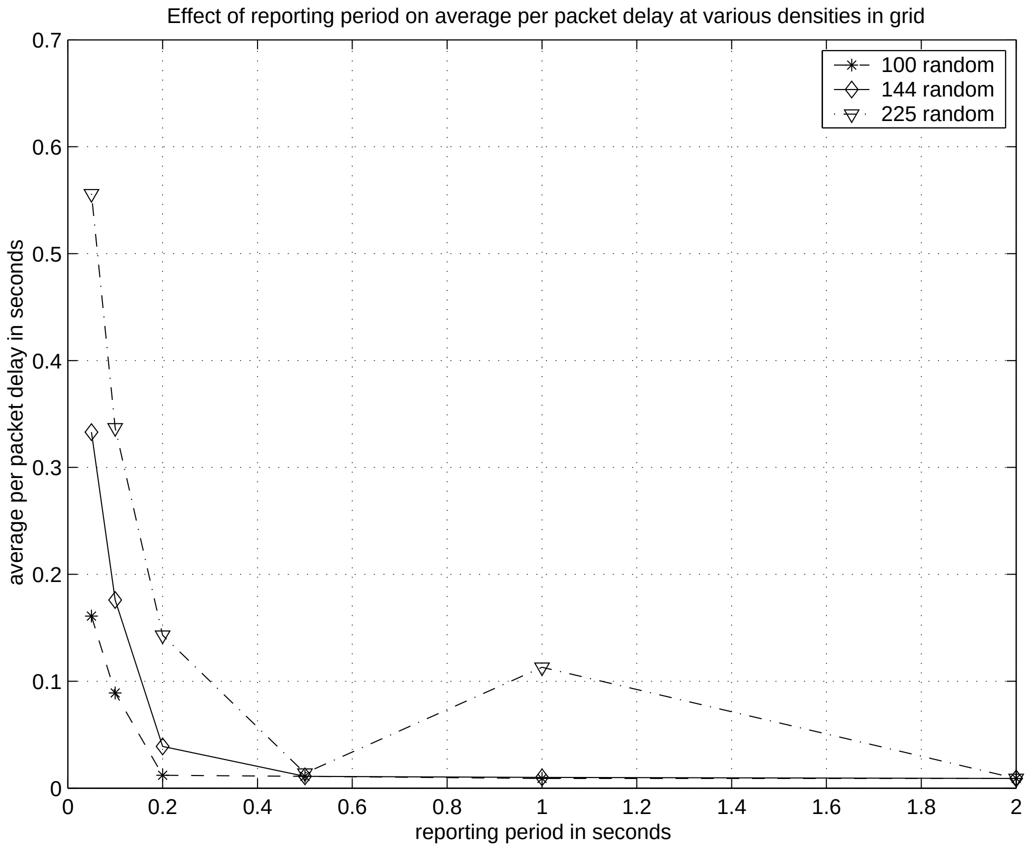


Figure 5.4: Delay as a function of network density and sensor reporting period (random deployment).

ples received in each slot. We then compared this average to the actual location of the phenomenon at that time. The error is the square root of the sum of the square of the difference between the estimated location and the actual location averaged over the number of slots in the simulation. More specifically,

$$E = \frac{\sqrt{\sum_{i=0}^{n} (S(i) - A(i))^2}}{n} \tag{5.5}$$

where $S(i)$ is the sensed value in time slot i, $A(i)$ is the actual value at time slot i, and n is the number of slots in the duration of the simulation. This is a proof of concept approach to calculating error; any statistical measure for correlating the measured value against the

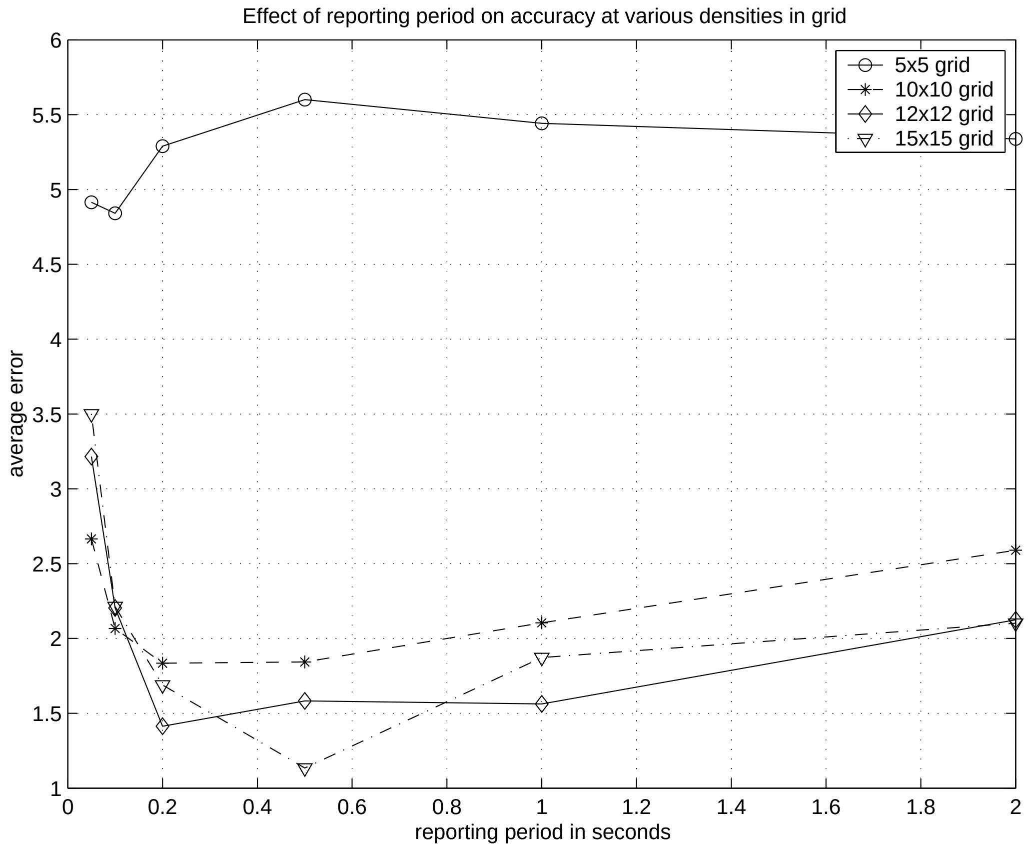


Figure 5.5: Error as a function of network density and sensor reporting period (grid deployment).

actual value will suffice.

Figure 5.5 shows the average error for the grid deployment strategy under different densities and for different reporting periods. At high reporting rates, network capacity is exceeded, as was observed in the previous graphs. Because of the latency in the receipt of the samples and the loss of many samples, the error value is high. On the other hand, if the reporting frequency is low, not enough samples are obtained and the average error rises. With sparse networks (example 5x5 grid) the error is higher when the network is not saturated because the number of sensors in a position to measure the phenomenon and the average distance between a sensor and the phenomenon increases. For such scenarios, the error is minimized with a higher reporting frequency; the additional samples reduce

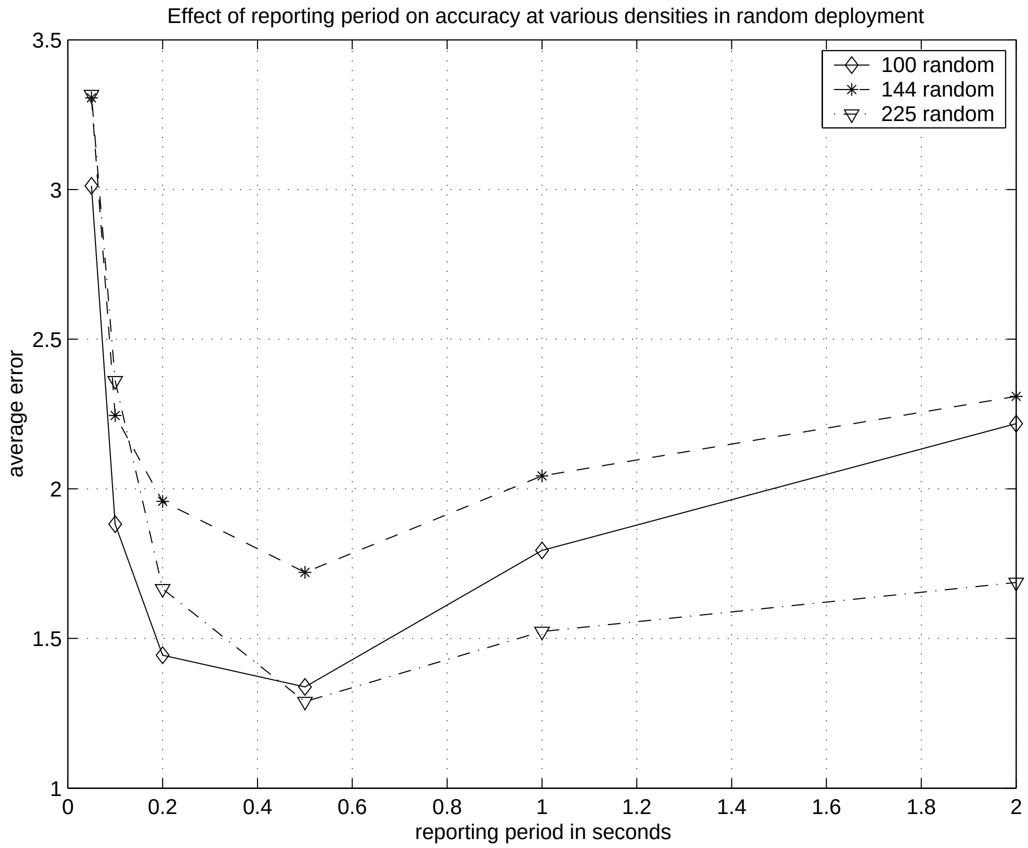


Figure 5.6: Error as a function of network density and sensor reporting period (random deployment).

the error and the network is slower to saturate because there are fewer sensors competing for the shared air space. With random deployment (Figure 5.6) the same pattern can be observed.

Energy-Efficiency Study

The energy depletion in the network is shown in Figure 5.7 and Figure 5.8 for the grid sensor deployment and random deployment respectively. The energy depletion is a function of the reporting rate as well as the density of the network. Recall that the density of the network in the phenomenon driven scenario correlates with the number of nodes that report their data. However, as suggested by the goodput results, a large portion of this energy is wasted

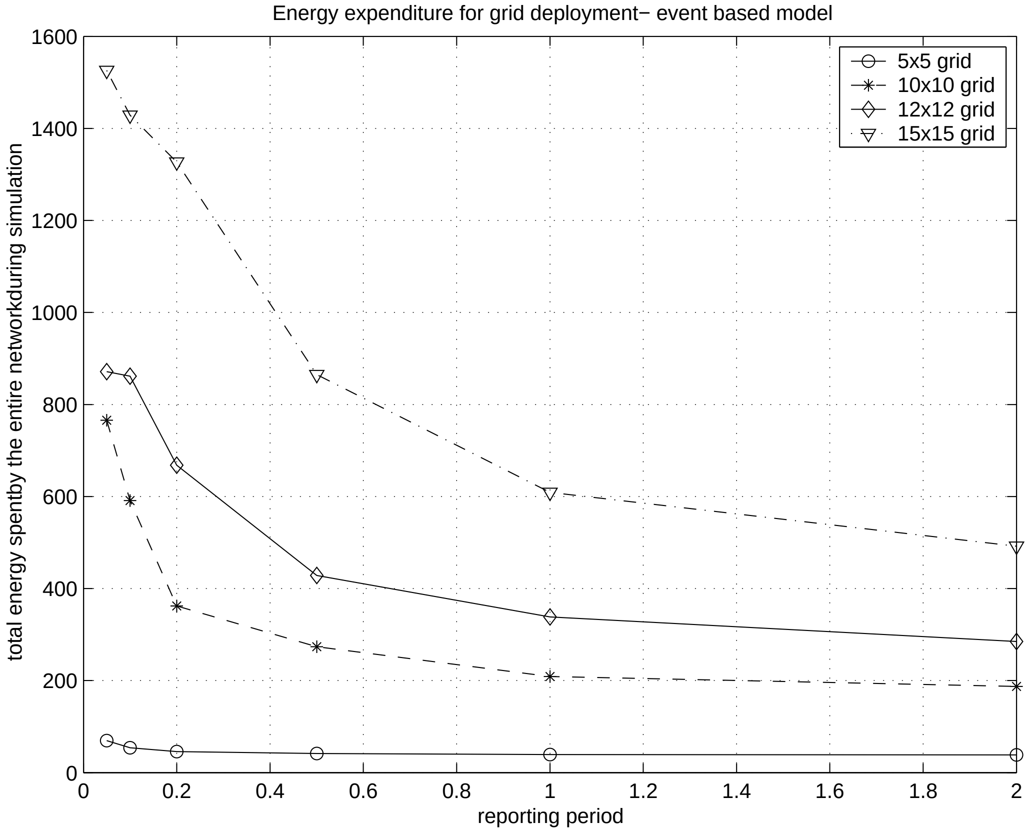


Figure 5.7: Energy depletion as a function of network density and sensor reporting period (grid deployment).

when the capacity of the network is exceeded. Moreover, the additional cost incurred to buy more sensors will not be rewarded by a higher lifetime for the network because the depletion rate also increases. In fact, when we consider the normalized energy expenditure per sensor (as Figure 5.9 shows for grid deployment) the average sensor gets depleted more quickly with higher density. Thus, the lifetime of the network likely drops with increased density even though we start with a much higher total available power in the network! Accordingly, there is a need for intelligent management of the infrastructure from an energy perspective as well.

To summarize, in agreement with intuition, increasing the network density can result in higher accuracy, but only if the sensing traffic is kept below the network capacity. This is an

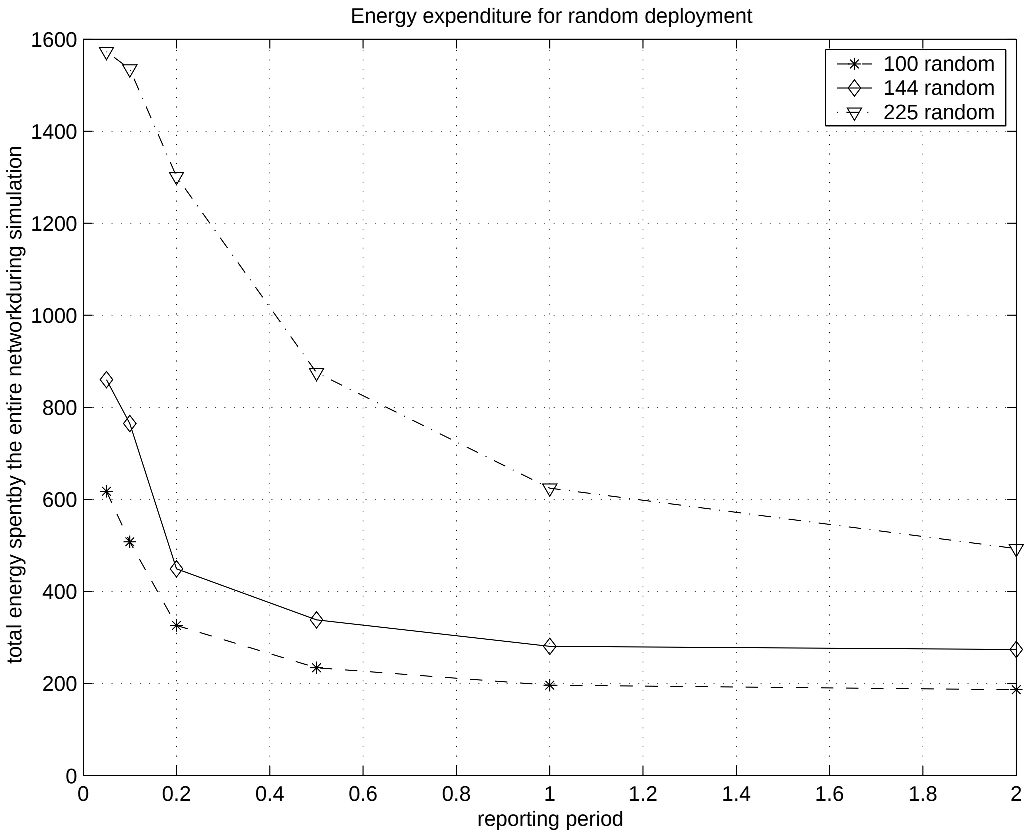


Figure 5.8: Energy depletion as a function of network density and sensor reporting period (random deployment).

expanded form of the congestion control requirement for regular computer networks; due to the redundant collective communication nature of sensor network traffic, the network has the ability of controlling what data gets reported to meet the observer requirements. It is likely that the observer is satisfied with less than the optimal achievable accuracy. Thus, the network protocol must control the available infrastructure and the reporting discipline to meet the accuracy requirements while minimizing the energy expenditure. The sensor network must converge on a good accuracy to reporting pattern/energy solution. This may be achieved, for example, by deciding to turn off some sensors, by adapting the reporting frequency, or by fusing sampled data within the network.

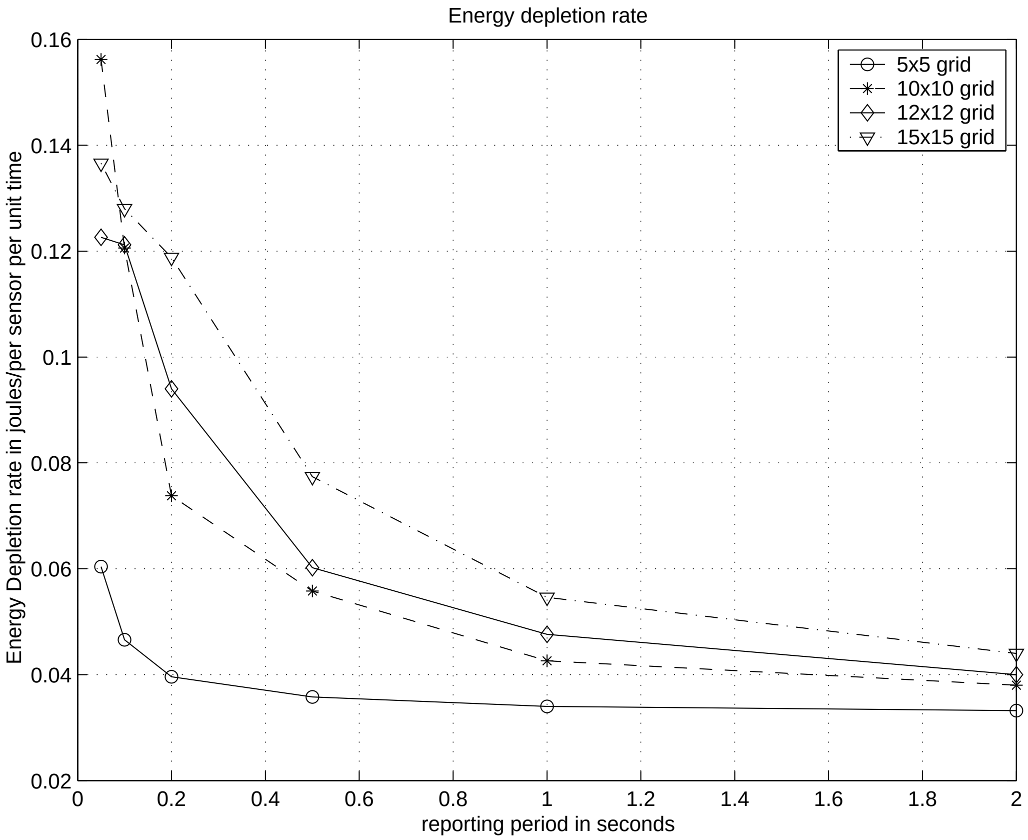


Figure 5.9: Energy depletion per sensor as a function of network density and sensor reporting period (grid deployment).

5.4.2 Continuous Update Reporting Model

For the continuous update reporting model (all sensors report continuously), the offered load was significantly higher than the phenomenon driven model. Energy depletion results (not shown) displayed this effect. As can be seen in Figure 5.10, the goodput values using continuous update reporting were significantly lower than for phenomenon driven traffic. Error is not directly comparable across the two scenario types.

5.4.3 Controlled Deployment

In this experiment, we study the effect of biasing the deployment to the phenomenon's motion pattern. In this experiment, the phenomenon was restricted to move in the right

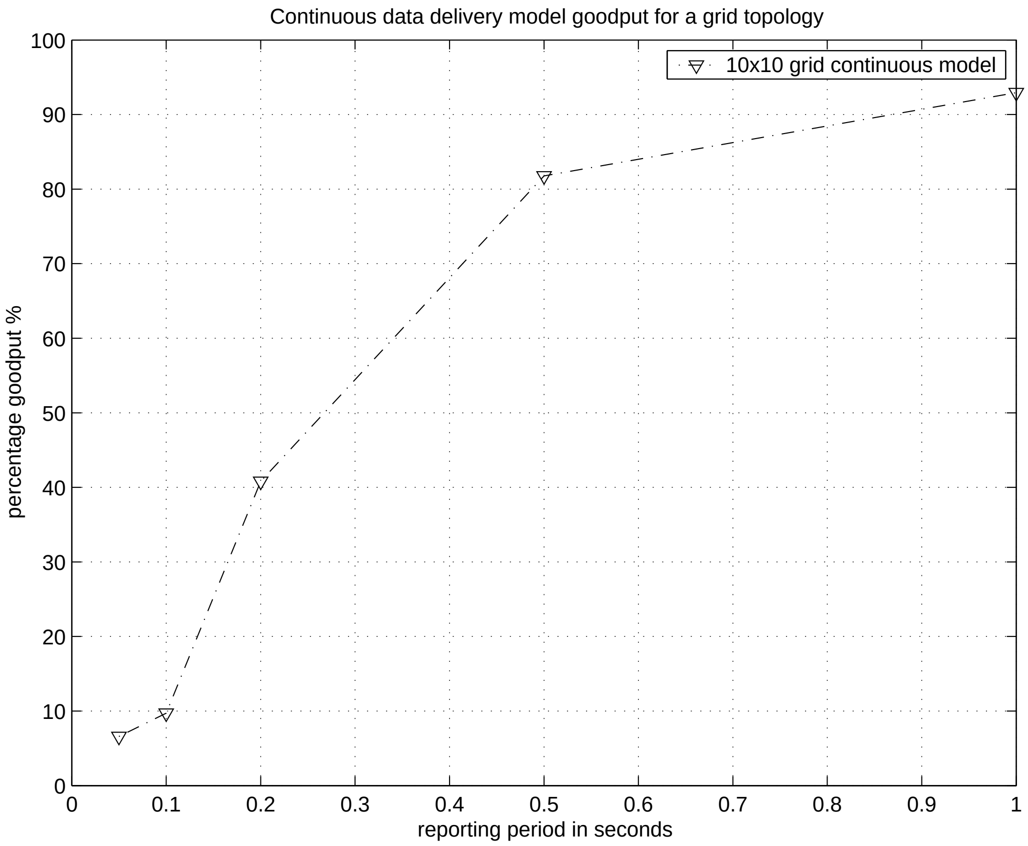


Figure 5.10: Goodput as a function of sensor reporting period for continuous update traffic and a 10x10 grid.

half of the 800 meters by 800 meters field. Furthermore, the deployment of the sensors was skewed to reflect this fact: the density of the sensors in the left half was kept fixed (and low) while the density of the sensors in the right half was increased. Figure 5.11 shows the accuracy using biased deployment vs. grid deployment. As can be seen from the figure, the desired effect of increasing the accuracy was achieved (the average error is lower in the biased deployment case). In fact, with biased deployment, a network of 100 sensors performs better than one with 144 sensors that are deployed in a grid. However, note that with aggressive reporting the network saturates faster under biased deployment, since the average number of nodes within reporting range of the phenomenon increases. This effect was also seen in the goodput results (not shown). The increased accuracy comes at the cost

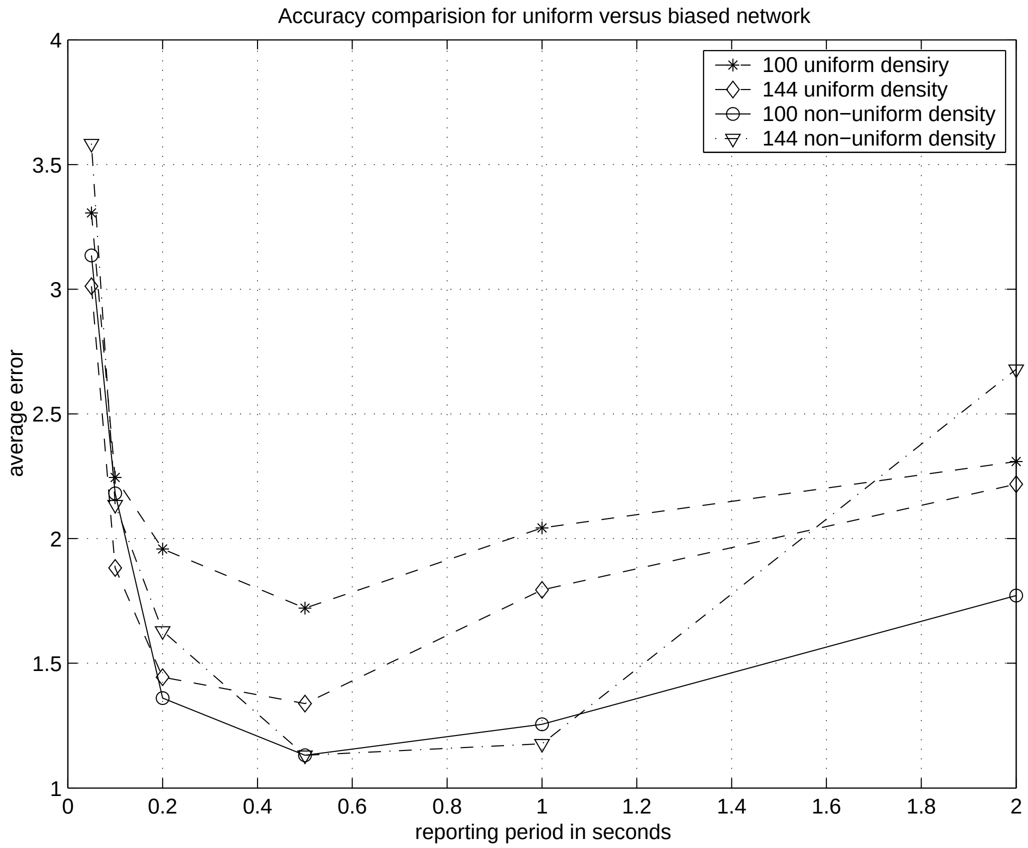


Figure 5.11: Average error comparison – controlled vs. grid deployment.

of extra energy depletion as well (results not shown). These results again argue that the network protocol should carefully manage the infrastructure.

5.4.4 Other Supporting Experiments

In order to further investigate the congestion problem, we conducted experiments with different packet sizes. The main purpose of these experiments was to study the effect on goodput and accuracy with a change in bandwidth, where an increase in packet size corresponds to a reduction in bandwidth. The results shown in Figures 5.12 and 5.13 show that congestion becomes a serious problem with low bandwidth (packets with large size), as goodput drops dramatically and the average error increases appreciably.

We also conducted experiments with the observer at different relative distances with

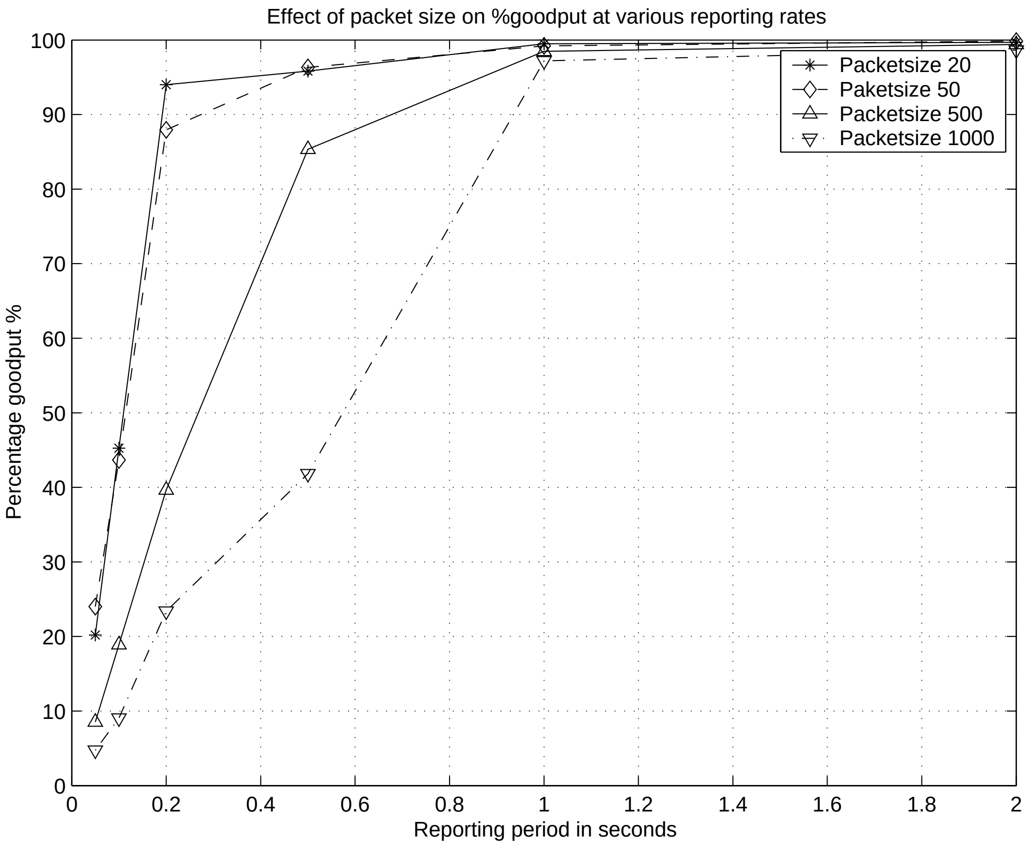


Figure 5.12: Goodput as a function of packet size and sensor reporting period for a 15x15 grid.

respect to the phenomenon, moving closer to the phenomenon in each case. Results shown in Figure 5.14 indicate that with an increase in path length, goodput decreases from 85% when the observer is closest to the phenomenon to 30% when the observer is furthest from the phenomenon. This agrees with the previous results in that the further the observer is from the phenomenon, the longer the average path from the sensors to the observer and the more data must be transmitted throughout the network, increasing network load and causing congestion.

Although the results we have presented and the conclusions we have drawn should be not be heavily impacted by the network/routing protocol (ignoring in-network processing), we investigated the effect of using other routing protocols. We investigated AODV [27]

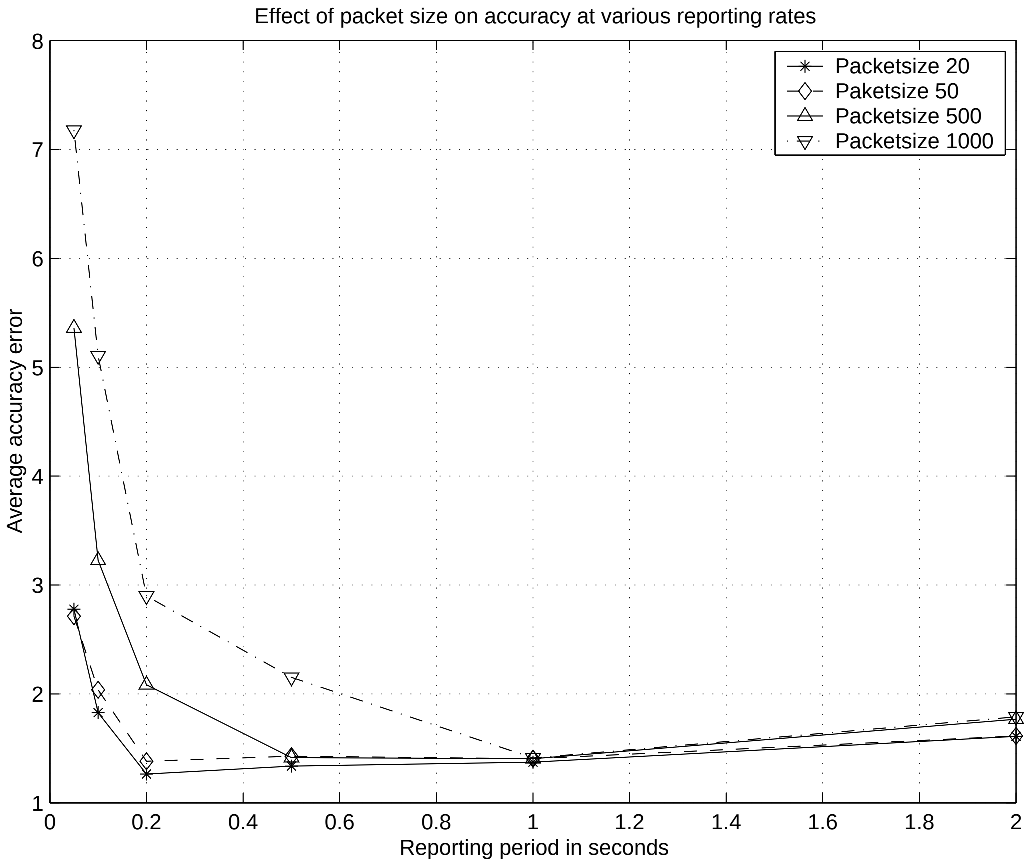


Figure 5.13: Error as a function of packet size and sensor reporting period for a 15x15 grid.

which, like DSR is reactive routing protocol. In addition, we studied DSDV [28], which is a proactive protocol. The results are shown in Figure 5.15. AODV performed almost identically to DSR, while DSDV was considerably poorer in all cases.

5.5 Related Work

Because of the unique requirements on sensor network nodes, several groups have proposed architectures for sensor nodes [1, 6, 9, 20, 29, 30, 32, 38, 43]. On top of these architectures, several studies targeted the development of power-efficient medium access protocols (e.g., [37, 39, 43]). Networking and data dissemination issues have also received

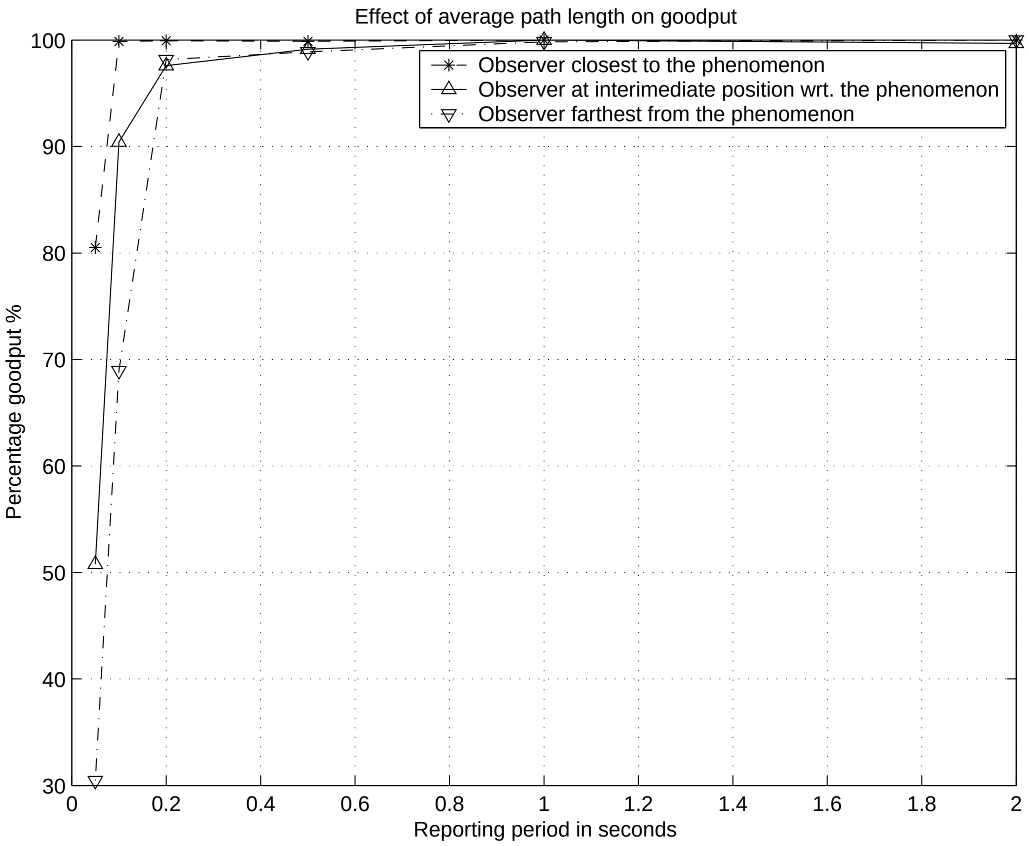


Figure 5.14: Goodput as a function of variation in path length and sensor reporting period for a 15x15 grid.

considerable interest. Due to the data-centric nature of sensor networks, researchers proposed alternative addressing schemes that take advantage of this fact [10, 16]. A number of routing/data aggregation approaches were also proposed [3, 12, 13, 17, 23]. A number of studies have explored implementing services for sensor networks, including positioning mechanisms [4, 26, 35], time synchronization [8] and energy scans [45]. Other studies considered specific sensor network applications and their implication on protocol design [5, 36, 40, 42].

Meguerdichian et al. define the problem of exposure in sensor networks [24] and propose localized algorithms to address it [25]. The exposure problem is the problem of determining whether a sensor network can keep track of a phenomenon that moves within the observation field. Depending on the sensor density/deployment, there could be blind spots in the

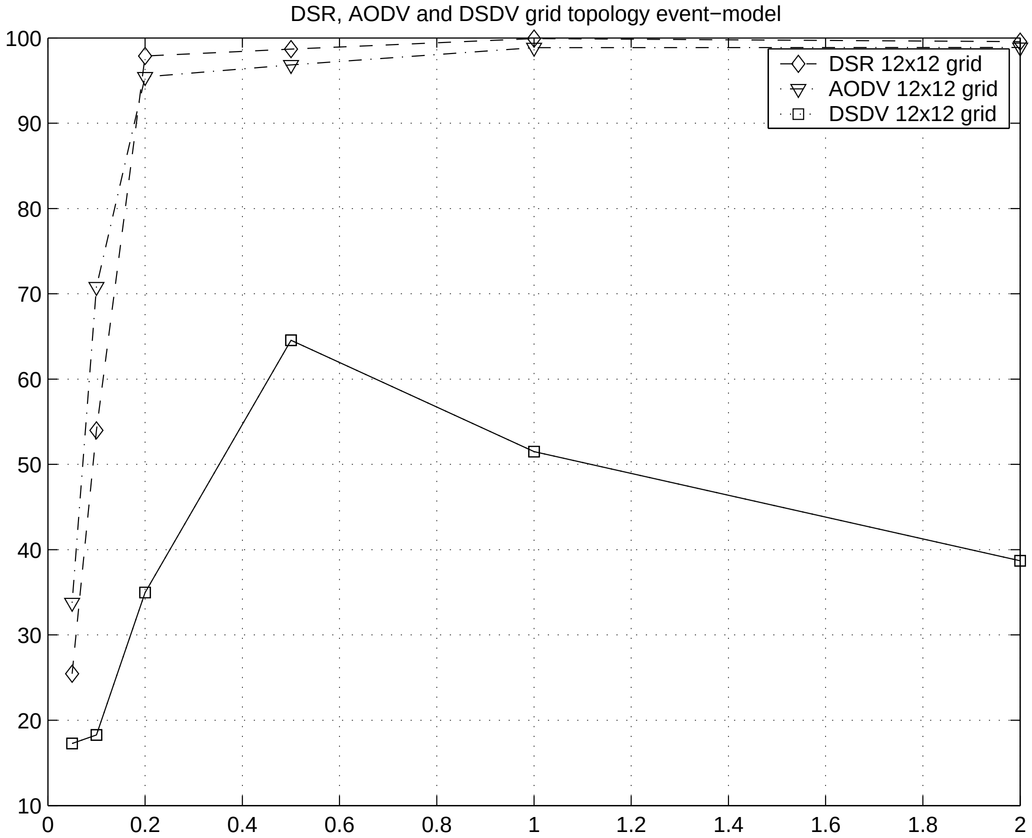


Figure 5.15: The effects of the routing protocol on goodput for a 12x12 grid.

observation field. Clearly, exposure is influenced by the deployment configuration of the sensors and is related to our work.

5.6 Concluding Remarks

In this chapter, we investigated the effect of infrastructure tradeoffs on the performance of a sensor network. First, we systematically increased the deployed sensor density and the required reporting rate and observed the performance of the network. When the offered load from the sensors to the network exceeded the capacity of the network, the performance dropped according to both network and application level metrics. Thus, by simply deploying more sensors, we may end up harming the performance of the network. This argues for intelligent management of the infrastructure by the network protocol: a form of congestion avoidance is needed that is significantly different from congestion avoidance in traditional data networks. In particular, the network protocol must balance the offered load to the network against the required accuracy at the observer.

The task of the sensor network may be viewed as a redundant collective communication process from the sensors to the observer. It is redundant in that multiple sensors may report correlated information or information with an accuracy level (e.g., reporting rate) higher than that required by the application. Thus, the congestion avoidance mechanism must converge on a reporting rate/discipline that is just sufficient to meet the performance requirements at the observer. The networking protocol may accomplish this by reducing the reporting rate per sensor, turning some sensors off and/or fusing information to optimize the collective communication operation.

We also investigated the effect of different deployment strategies on the performance of the network. We discovered no appreciable differences between grid-type deployment and random deployment for the scenarios we considered. However, biasing the infrastructure density to the phenomenon movement pattern resulted in significantly higher accuracy. This is an example of using application level information to better architect the infrastructure.

Chapter 6

Congestion Management for Sensor Network

[1]

The infrastructure tradeoffs study has shown that congestion could be a real problem for wireless micro-sensor networks. In traditional data networks, congestion is addressed either by congestion avoidance (preventing the system from entering into a congested state) or by congestion control (detecting congestion and recovering by reducing the sending rate). We have established that communication in sensor networks is not end-to-end. Furthermore, to aid scalability and reduce the load on the sensors, interactions should be localized [9]. Accordingly, novel solutions for congestion avoidance are necessary in the context of sensor networks. This chapter discusses some initial solutions to achieve this goal. The high-level goal targeted by these solutions is to meet the application specific goals such as accuracy and lifetime by shaping the traffic in an application specific way. The fundamental concept

[1] This chapter is based on an article under preparation for submission

is using application specific policies to balance the load on the network against the accuracy desired by the observer.

6.1 Introduction

Congestion is a well-known and well-studied problem in the context of data networks and the Internet. In data networks, communication is end-to-end between two communicating peers. Furthermore, the best effort model of the Internet makes it possible to treat all connections identically. Thus, unless the traffic is prioritized, all the communicating streams have same right to use the network resources such a bandwidth, buffer queues etc. For example, consider three http connections among following: (Client A, Server B), (Client C, Server D) and (Client E, Server F). Fairness is important, which means that any good congestion control/avoidance algorithm should not control/avoid congestion by shutting down traffic between some of the communicating parties say (Client A, Server B) in our example. Thus, fairness is one of the key considerations to any congestion control/avoidance algorithm in an Internet like architecture.

In contrast, in sensor networks the communication between the observer and the sensors in not end-to-end. The observer is not interested in communicating with any particular sensor or a group of sensors; in fact, it is not necessarily aware of the underlying sensor network infrastructure. Instead, the observer is interested in any sensor or group of sensors only in the nature of the information that they can supply about the phenomena under consideration. This is a crucial difference from traditional data networks that has important implications on congestion avoidance.

A second critical difference between data networks and sensor networks is in the definition of the objective function that the networks attempt to maximize. In data networks, a reasonable objective function for a rate regulating mechanism is to optimize the aggregate throughput of the network while maintaining fairness across connections. In contrast, in sensor networks, the objective that the network is most appropriately measured in application terms – does the network provide sufficient information, with acceptable delay, to the observer to meet its desired sensing needs. A coexisting objective is to supply this information with a minimum expenditure of energy to prolong the network lifetime. Thus, traditional transport protocol objectives such as maximizing throughput and achieving fairness do not apply. A good congestion control algorithm should satisfy these application specific goals by shaping the traffic in application specific way. Furthermore, in the situation where the network cannot meet the aggregate observer demands with the available capacity, the congestion control algorithm must attempt to gracefully degrade the delivered sensing performance.

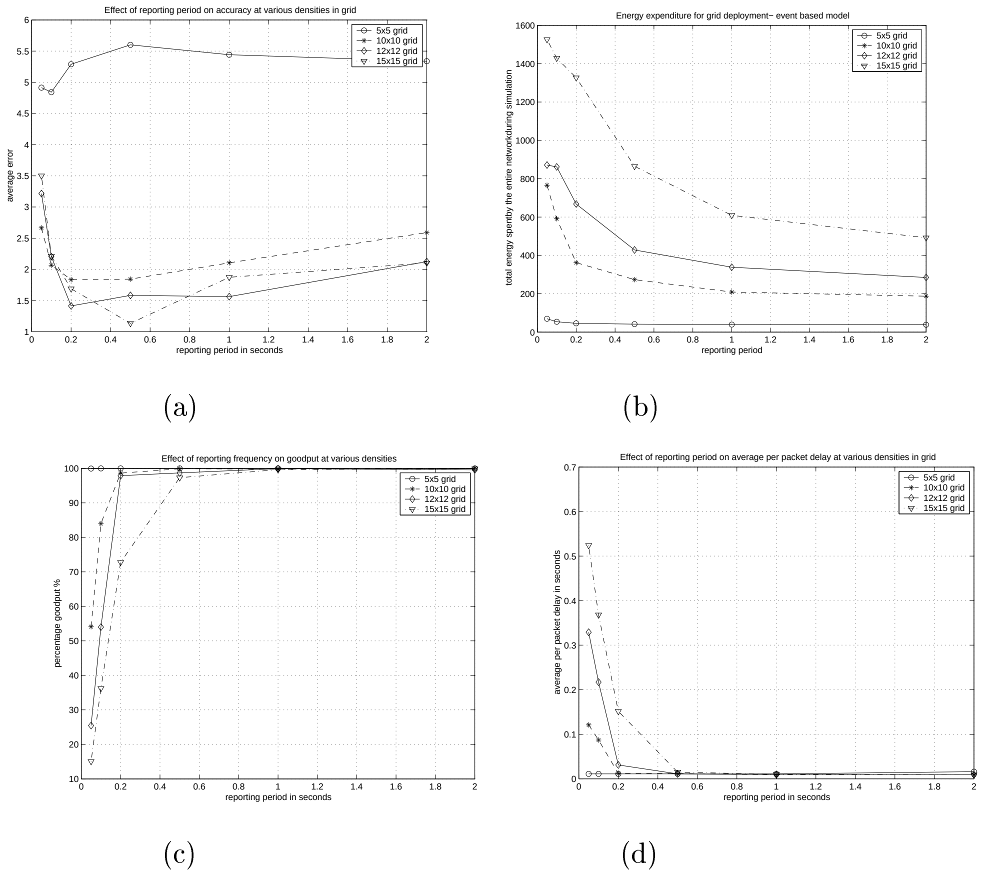
This chapter proposes some initial ideas for addressing the congestion in a sensor network context. It proposes a suite of congestion avoidance algorithms on the basis of the objectives described above. The chapter is organized as follows: The first section presents results showing congestion problem for sensor networks. Next section studies classification of congestion control and avoidance algorithms.Next section describes the design goals for the congestion avoidance algorithm. Next section presents a suite of congestion avoidance algorithms and then we present results supporting our claims. Finally we present our future research directions and conclusions for this chapter.

6.2 Congestion Problem for sensor network

Sensors are tiny devices with limited battery power. An important goal of the design of a sensor network is to extend the lifetime of the network – battery replacement may not be possible. For example consider sensors are thrown from a plane in an inhospitable area for intrusion detection. In this case due to the fact that physical environment is inhospitable it might not be possible to replace the batteries at all. Consider sensor network deployed in a forest for animal detection. In this case even though the physical environment might not be inhospitable battery replacement or re-charging might not be possible from cost or effort point of view. Congestion if not handled reduces the lifetime of network by draining network resources. Furthermore, congestion may result in poor accuracy because samples are lost in route to the observer. We believe that congestion handling is a crucial problem for sensor networks. Low bandwidth, large number of reporting sensors, low battery power makes congestion a severe problem for wireless micro-sensor networks.

Results for infrastructure tradeoffs have shown that there exists an optimal region, where if the given sensor network operates then it can achieve sufficient accuracy, high throughput at acceptable energy depletion rate. For example consider the following figures. These results are taken for a sensor network organized as a grid with varying densities in the same area. A $5x5$ grid refers to 25 sensor nodes including 1 phenomenon and 1 observer and so on.

We can see that there exists an optimal region in energy, accuracy and good-put diagrams. The right side of the optimal region represents low frequency reporting. In the right region accuracy decreases because the observer does not get enough samples. So even though the spent energy is low and good-put is high this region may not be desirable be-



(a) (b) (c) (d)

Figure 6.1: These figures show accuracy (a), energy depletion rate (b), good-put (c) and delay (d)as a function of data rate

cause not enough data may be reported to meet application requirements. In contrast, the region to the left of the optimal accuracy represents high frequency reporting to the point where congestion occurs. In this region sensors are reporting too aggressively resulting in congestion and collisions; the offered load exceeds the channel and/or network capacity and thus good-put drops dramatically. Also because of congestion, the average per packet delay increases. Due to this excessive reporting rate, as shown in the energy-plot, energy is depleted quickly in this region. Clearly, this region is not desirable because it represents high-energy depletion rate with low accuracy, low good-put and high delay.

An effective operating mode should reside above the point required to meet the application demands. It should also reside below the network capacity. A case where the application demands exceed the network capacity means that the observer requirements are not feasible in the current network and a less ambitious reporting level should be found. From application point of view the sensor network should always operate in the optimal region where moderate energy depletion rate is observed with acceptable delay, accuracy and good-put. In this region offered load does not exceed the network capacity and network resources are not underutilized. A general solution should converge to a reporting discipline (what sensors report and when) that meets the application requirements while reducing the load on the network; this is an area of interest for future research. In this chapter, we consider only controlling one aspect of ineffective operation – avoiding congestion in the event that the reporting discipline is too aggressive.

6.3 Classification of Congestion Handling mechanisms

Broadly, congestion can be addressed either by Congestion Avoidance or by Congestion Control. Congestion Avoidance is shaping the traffic in such a way that the system never enters into congested state. Congestion Control is detecting congestion and recovering from it when it occurs.

Congestion Avoidance:- Congestion avoidance is a preventive measure, where the traffic is generated in a way where the system does not enter into congested state at all. This can be further classified as

1. Feedback based
   (a) Observer based feedback
   (b) In-network feedback
   (c) Hybrid approach.
2. Methods with no explicit feedback.

All feedback based methods requires some communication in terms of feedback. The feedback might be a single step feedback or a series of forward and backward feedbacks where an oscillating system is brought to a stable state. Based on the component which provides the feedback its is further sub classified.

Lets us consider all the feedback methods in detail.

1. Observer based feedback - In this case the observer provides the feedback to a sensor network about the congestion. This represents the model where all the intelligence is

assumed with the observer and the sensor network is assumed to be dumb resembling Smart server and dumb client paradigm.

The observer uses some efficient statistical techniques to find out the behavior of sensor network in terms of application specific goals such as accuracy in real time. For example, the observer could find out variance in its estimate of accuracy as the sensors change their reporting frequency; if the variance is higher than its accuracy tolerance, it may need to increase the reporting rate. If increasing the reporting rate increases the variance, congestion is detected. Early detection is important for limiting the period that the sensor network operates in the congested part (or left part of the optimal region).

In this case the observer can notify sensors with a network-wide congestion warning broadcast message which once received by sensors can be used to alter the reporting frequency. Also some smarter technique for observer based feedback can be used.

2. In-network feedback based congestion control:- In this scheme, the sensors in the network try to do early detection of congestion based on parameters such as increase in drop rate for packets etc. If packet drop rate exceeds a certain threshold value, then it can be used as a sign for drift to-wards the left region of the optimal region. A sensor can then use some feedback mechanism such as explicitly sending a congestion warning message to the neighbors who reduce their sending frequency and in turn resend this message to their next one hop neighbors. Each such message can be generated by the original source with an id to distinguish between different instances of the same message from different neighbors.

   This puts the burdon of congestion control entirely on the sensors. However, it will be interesting to study the tradeoffs of energy overhead due to additional complexity versus efficiency in congestion control thereby saving energy.

3. Hybrid approach:- In this case a combination of both observer based feedback as well as in-network feedback can be used. Observer can take into consideration both variance in accuracy in real time as well as packet drop rate variance information gathered from the sensor network to figure out congestion early enough.

   In all the above methods a feedback message or a series of feedback message is used to avoid congestion. However due to characteristics of communication in sensor network we argue that there exist congestion avoidance methods for sensor networks, which require no communication or feedback method. Comparative study of feedback based versus no feedback based methods is one of our future goals. The next section describes the congestion avoidance algorithms which require no communication.

6.4 Congestion Avoidance Algorithms

This section describes the congestion avoidance algorithms. These algorithms are based on following assumptions:-

1. Co-related information

2. Redundant information.

3. No global time synchronization.

Let us consider these assumptions one by one in the context of a common example.Let us imagine a sensor network deployed for animal tracking with event-driven data delivery model [42]. That means the sensors wake-up periodically and check whether an event has occurred, in this case the event is presence of animal in the sensing range of the given sensor. If the animal has walked into the sensing range then sensor will gather information about the animal and send it to the observer. If we consider the topology below, then we can see that this event can be sensed by multiple sensors at the same time and thus they will start reporting about this event to the observer. With this model let us consider the assumptions.

1. Co-related information:- These sensors are going to report information about the features of the animal thus this information is highly co-related.

2. Redundant information:- We can see multiple sensors might be reporting the same information such as animal co-ordination or other features of the animal to the observer. This leads to redundancy in information.

3. No global time synchronization:- Our algorithms do not assume presence of global time synchronization. If it is present it can be used however the algorithms do not depend on its presence.

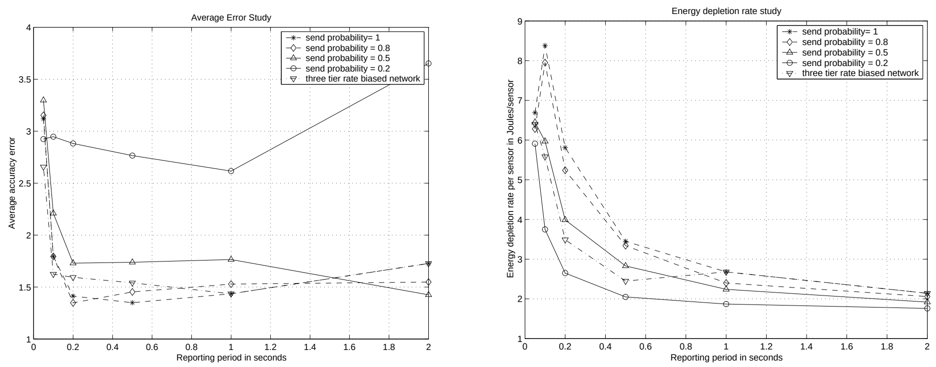
   In a sensor-network, the observer is not interested in information collected by an individual sensor but in the high level description such as animal co-ordinates, or some other feature of animal. Thus there exists a lot of redundant traffic. The crucial point is that this traffic differs from the usual Internet like architectures where each pair of connection should be treated equal due to end-to-end traffic. Redundancy in information and co-related traffic can be used by congestion avoidance algorithms to reduce total traffic by reducing redundant traffic as much as possible.

   The algorithms can be broadly classified as

   (a) Unbiased algorithm.

(b) Biased Algorithms.

(a) Unbiased algorithm:- This algorithm treats all the sensors which can sense the phenomenon. This assumes that every sensor which can sense the phenomenon has equal knowledge about the phenomenon. In our example we can see that even though an animal is in range with many sensors the sensors which are closer to the phenomenon can in general have better knowledge about the sensors. However unbiased congestion avoidance algorithm do not use this fact. This is a probabilistic algorithm, where every sensor generates a random number and then probabilistically decides whether to send its data to the observer or not depending upon this generated random number. We have done experimentation with a range of probabilities ranging from low to high where a sensor decides to transmit. The following graph shows the comparative study in terms of energy depletion rate, accuracy, good-put.



(a) (b)

Figure 6.2: These figures show accuracy (a), energy depletion rate (b), as a function of data rate

These results clearly show that, with this probabilistic unbiased algorithm, there can be significant savings in terms of energy. However, the crucial point is that there can be a range of probabilities where there is significant savings in the energy, while at the same time application specific accuracy requirements are met. As stated already there is no feedback involved in this algorithm. However, this algorithm does not adapt itself dynamically to the number of active sensors. Active sensors are the sensors that can sense the phenomenon. However, there exists a non-zero probability that all the sensors decide not to transmit in a range of time, resulting in no information getting reported to the observer. As seen in the graph for the case where a node decides to send with a probability of 0.2 there the accuracy is bad this is because of lack of enough samples. With send probabilities of 0.5 and 0.8 accuracy requirements are met with significant energy savings compared to send probability 0f 1.0.

Biased Algorithm:- In this we treat the sensors with different priorities. The main idea behind this algorithm is that, since in sensor networks there is high amount of co-related and redundant traffic, we try to prioritize the traffic in application specific way. Sensors with valuable data again in application specific way are treated with high priority. As a proof of concept and to support our claims we have implemented the policy in our framework in the following way. We assume that a phenomenon will be observed by multiple sensors and the sensors which are closer to the phenomenon can give more accurate information about the phenomenon than those which are far away from the phenomenon but phenomenon is still in the range of those sensors. For example, in our animal tracking application, we argue that sensors close to the phenomenon can better judge the phenomenon say sensors in a radius of 50 meters from the phenomenon can have more accurate information such as phenomenon coordinates or other features that those, which are say at a distance of 100 meters from the phenomenon, ie., we assume that average error scales as the relative distance between the sensor and the phenomenon. This is just one mechanism to support the policy. Applications which do not follow this notion can implement their own policy. Clearly, unbiased algorithm is a special case of the biased algorithm where all sensors in the sensing range are treated the same way.

The above results show that there can be significant savings in the energy with acceptable accuracy as compared to both protocols with no congestion avoidance as well as the proposed unbiased congestion avoidance algorithms. Thus we contend that the application specific policies are better suitable for congestion control/avoidance since the application requirements are met with optimal use of network resources. These results are taken with two categories:- (1) Sensors in the sensing range imagined to be organized in two concentric circles. The first circle consists on sensors with radius of half the sensing range and the others in the outer circle with radius greater than half the sensing distance to the sensing distance. (2) Sensors in the sensing range imagined to be organized in three concentric circles. The first circle consists on sensors with radius of one third the sensing range and the second in the middle circle with distance up to two-third the sensing distance and the outermost for the sensors which are more than two third the sensing distance from the phenomenon but within the sensing distance. Results for the second case (three-tiered network) are shown in the graphs **??**, refca-acc. With application specific knowledge coupled with infrastructure knowledge, we can perform better than unbiased algorithm.

In all these cases sensors which are close send data with a high probability and sensors which are far send the data with low probability. This is because of two reasons (1) Sensors close enough are going to be more accurate However (2) We cant not guarantee that all the information can be captured by the close sensors for example there could be some sensor not very close to the phenomenon but it might be the only sensor in that direction thus it might be the only sensor, which can disseminate some information about the phenomenon. Thus, the biased algorithm tries to take advantage of application specific knowledge sensors.

The main advantage of both biased as well as unbiased algorithms is that the decision whether to send data or not is completely local. There is no need for communication at all. The drawbacks of this algorithms is that it does not adopt itself dynamically. We are currently investigating protocols, which use feedback for making dynamic decisions about turning off or on the sensors.

6.5 Related work

Previous work with SPIN [13] uses meta data to reduce redundancy in data dissemination. Protocols like Directed diffusion [17] propose in-network processing to reduce data traffic. However we have not seen any work so far, that addresses the congestion issue. To the best of our knowledge this is the first attempt to manifest the congestion problem in sensor-networks as well as some suggestions to tackle the problem. The main advantages of the protocols proposed in this chapter are simplicity, very low overhead and potential to meet application requirements such as accuracy and energy efficiency. These characteristics make them a suitable candidate for sensor-networks, where complex protocols can not be used to avoid congestion due to hardware constraints. By this we mean due to low battery power, low bandwidth and limited processing ability, sensors can not afford to run complex protocols, but at the same time application specific requirements are to be met. Such networks are good candidates for the proposed protocols.

6.6 Future work

The proposed protocols clearly show that there exists a room for optimization. We are currently investigating better and more intelligent approaches to both shaping traffic in a better way (congestion avoidance) and congestion control. The results in this stand as a base for our further research. These protocols indicate that intelligent management of sensors is the key for efficient data dissemination. However, these protocols represent just one of the possible alernatives in the design space. We are currently developing feedback based congestion avoidance/control protocols for sensor networks.

Chapter 7

Future-Work and conclusion

This thesis has presented some exciting and promising directions for our future research. Future work includes making taxonomy more comprehensive. We would like to update our taxonomy to make it more comprehensive by adding new applications to it which have unique characteristics from those which are already there. Infrastructure tradeoffs analysis has presented congestion problem for sensor networks. As presented earlier we have some preliminary results, which show that if sensors are managed more intelligently then application requirements can be met at the same time network is prevented to go into congestion zone. However we are planning to work on a better strategy to manage these sensors to avoid congestion. Congestion avoidance/control problem is one of our future goals. From Infrastructure tradeoffs we can see that data-fusion or in-network processing is key to reduce traffic to avoid congestion as well to meet application requirements. We would like to pursue research in the direction of data-fusion or in-network by developing novel protocols which support data aggregation.

7.1 Conclusion

Sensor networks are highly specialized networks with co-operative communication. Traffic characteristics in sensor networks make them unique compared to regular network with end-to-end traffic. Thus this thesis consists of theoretical work of taxonomy of sensor networks and then evaluation framework to validate the assumptions from taxonomy. Finally we present experiments to validate out claims.

The overall communication behavior in a wireless micro-sensor network is application driven. We believe that it is useful to decouple the application communication used for information dissemination from the infrastructure communication used to configure and optimize the network. This separation will aid network designers in selecting the appropriate sensor network architecture that will best match the characteristics of the communication traffic of a given application. This will allow the network protocol to achieve the application-specific goals of energy-efficiency, low latency, and high accuracy in the sensing application. We believe that the taxonomy we have presented will be helpful in designing and evaluating future network protocols for wireless micro-sensor networks. We hope that this taxonomy will assist in developing relevant simulation models to enable empirical study of the performance of the different sensor network organizations and assist in making design and deployment decisions.

To validate taxonomy, we investigated the effect of infrastructure tradeoffs on the performance of a sensor network. First, we systematically increased the deployed sensor density and the required reporting rate and observed the performance of the network. When the offered load from the sensors to the network exceeded the capacity of the network, the performance dropped according to both network and application level metrics. Thus, by simply deploying more sensors, we may end up harming the performance of the network. This argues for intelligent management of the infrastructure by the network protocol: a form of congestion avoidance is needed that is significantly different from congestion avoidance in traditional data networks. In particular, the network protocol must balance the offered load to the network against the required accuracy at the observer.

The task of the sensor network may be viewed as a redundant collective communication process from the sensors to the observer. It is redundant in that multiple sensors may report correlated information or information with an accuracy level (e.g., reporting rate) higher than that required by the application. Thus, the congestion avoidance mechanism must converge on a reporting rate/discipline that is just sufficient to meet the performance requirements at the observer. The networking protocol may accomplish this by reducing the reporting rate per sensor, turning some sensors off and/or fusing information to optimize the collective communication operation.

We also investigated the effect of different deployment strategies on the performance of the network. We discovered no appreciable differences between grid-type deployment and random deployment for the scenarios we considered. However, biasing the infrastructure density to the phenomenon movement pattern resulted in significantly higher accuracy. This is an example of using application level information to better architect the infrastructure. We presented a general outline of congestion management for sensor networks. Finally we presented some preliminary results to demonstrate that if congestion management is done then we can achieve application specific goals such as accuracy and at the same time we can make better use of infrastructure resources.

Bibliography

[1] Asada, G., Dong, M., Lin, T., Newberg, F., Pottie, G., and Kaiser, W. Wireless integrated network sensors: Low power systems on a chip. In *European Solid State Circuits Conference* (Oct. 1998).

[2] Berkeley/LNBL/ISI, U. The ns-2 network simulator with the cmu mobility extensions, 2002. http://www.isi.edu/nsnam/ns/.

[3] Bhatnagar, S., Deb, B., and Nath, B. Service differentiation in sensor networks. In *Proc. 4th International Symposium on Wireless Personal Multimedia Communications* (Sept. 2001).

[4] Bulusu, N., Heidemann, J., and Estrin, D. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine* (Oct. 2000), 28–34.

[5] Cerpa, A., Elson, J., Estrin, D., Girod, L., Hamilton, M., and Zhao, J. Habitat monitoring: Application driver for wireless communications technology. In *Proc. ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean* (Apr. 2001).

[6] Chandrakasan, A., Amirtharajah, A., Cho, S., Goodman, J., Konduri, G., Kulik, J., Rabiner, W., and Wang, A. Design considerations

for distributed microsensor systems. In *Proc. of the IEEE 1999 Custom Integrated Circuits Conference (CICC'99)* (May 1999).

[7] Crepa, A., and Estrin, D. ASCENT: Adaptive self-configuring sensor network topologies. In *Proc. INFOCOM 2002* (June 2002).

[8] Elson, J., and Estrin, D. Time synchronization for wireless sensor networks. In *Proc. Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing* (Sept. 2001).

[9] Estrin, D., Girod, L., Pottie, G., and Srivastava, M. Instrumenting the world with wireless sensor networks. In *Proceeings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)* (May 2001).

[10] Heidemann, J., Silva, F., Intanagonwiwat, C., Govindan, R., Estrin, D., and Ganesan, D. Building efficient wireless networks with low-level naming. In *Proc. 2001 Symposium on Operating Systems Principles* (Oct. 2001), pp. 146–159.

[11] Heinzelman, W. *Application-Specific Protocol Architectures for Wireless Networks.* PhD thesis, Massachusetts Institute of Technology, 2000.

[12] Heinzelman, W., Chandrakasan, A., and Balakrishnan, H. Energy-efficient routing protocols for wireless microsensor networks. In *Proc. 33rd Hawaii International Conference on System Sciences (HICSS '00)* (Jan. 2000).

[13] Heinzelman, W., Kulik, J., and Balakrishnan, H. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. In *Proceedings*

of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99) (Aug. 1999), pp. 174–185.

[14] IETF MANET Working Group Internet Draft– Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-05.txt, 2001.

[15] IETF MobileIP Working Group Internet Draft. http://www.ietf.org/rfc/rfc2002.txt, 1996.

[16] Imielinski, T., and Goel, S. DataSpaces: Querying and monitoring deeply networked colections in physical space. In *Proc. MobiDE 1999* (1999), pp. 44–51.

[17] Intanagonwiwat, C., Govindan, R., and Estrin, D. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proc. 6th ACM International Conference on Mobile Computing and Networking (Mobicom'00)* (Aug. 2000).

[18] Internet Engineering Task Force MANET Working Group. Mobile ad hoc networks (MANET) charter. http://www.ietf.org/html.charters/manet-charter.html.

[19] Johnson, D., Maltz, D., Hu, Y., and Jetcheva, J. The dynamic source routing protocol for mobile ad hoc networks. Internet Draft, Internet Engineering Task Force, Mar. 2001. http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-05.txt.

[20] Kahn, J., Katz, R., and Pister, K. Next century challenges: Mobile networking for 'smart dust'. In *Proceedings of the Fifth Annual International*

Conference on Mobile Computing and Networking (July 1999), pp. 271–278.

[21] Li, J., Blake, C., De Couto, D., Lee, H., and Morris, R. Capacity of ad hoc wireless networks. In *Proceedings of the 2001 ACM Mobile Computing and Networking Conference (Mobicom'01)* (July 2001), pp. 61–69.

[22] Li, J., Jannotti, J., Couto, D., Karger, D., and Morris, R. A scalable location service for geographic ad hoc routing. In *Proceedings of the International Conference on Mobile Computing and Networks (MobiCom'00)* (Aug. 2000), pp. 120–130.

[23] Lindsey, S., Raghavendra, C., and Sivalingam, K. Data gathering in sensor networks using energey-delay metric. In *Proc. International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing* (Apr. 2001).

[24] Meguerdichian, S., Koushanfar, F., Qu, G., and Potkonjak, M. Exposure in wireless ad-hoc sensor networks. In *The Seventh Annual International Conference on Mobile Computing and Networking 2001* (July 2001), pp. 139–150.

[25] Meguerdichian, S., Slijepcevic, S., Karayan, V., and Potkonjak, M. Localized algorithms in wireless ad hoc networks: Location discovery and sensor exposure. In *Proc. MobiHoc 2001* (2001).

[26] Niculescu, D., and Nath, B. Ad hoc positioning system (aps). In *Proc. GLOBECOM 2001* (2001).

[27] Perkins, C., Royer, E., and Das, S. Ad hoc on-demand distance vector (aodv) routing. Internet Draft, Internet Engineering Task Force, Mar. 2001.

http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-08.txt.

[28] Perkins, C. E., and Bhagwat, P. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *ACM Computer Communications Review 24*, 4 (Oct. 1994), 234–244. SIGCOMM '94 Symposium.

[29] Pottie, G., and Kaiser, W. Wireless integrated network sensors. *Communications of the ACM 43*, 5 (May 2000), 551–558.

[30] Rabaey, J., Ammer, M., Da Silva, J., Patel, D., and Roundy, S. PicoRadio supports ad hoc ultra-low power wireless networking. *IEEE Computer 33*, 7 (July 2000).

[31] Ratnasamy, S., Estrin, D., Govindan, R., Karp, B., Shenker, S., Yin, L., and Yu, F. Data-centric storage in sensornets, 2002. Submitted to SIGCOMM'02. Available at: http://lecs.cs.ucla.edu/ estrin/papers/dht.pdf.

[32] Rockwell science center sensor network project, 2002. (Available on the web at: http://www.rsc.rockwell.com/wireless$_systems/sensorware$).

[33] Rockwell scientific company. (Available on the web at: http://wins.rockwellscientific.com).

[34] Rutgers University, C. S. D. Wireless sensor networks bibliography website, 2002. http://www.cs.rutgers.edu/ mini.

[35] Savvides, A., Han, C., and Strivastava, M. Dynamic fine-grained localization in ad hoc networks of sensors. In *Proc. 7th ACM International Conference on Mobile Computing and Networking* (July 2001).

[36] Schweibert, L., Gupta, S., and Weinmann, J. Research challenges in wireless networks of biomedical sensors. In *The Seventh Annual International Conference on Mobile Computing and Networking 2001* (July 2001).

[37] Shih, E., Cho, S., Ickes, N., Min, R., Sinha, A., Wang, A., and Chandrakasan, A. Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking* (July 2001), pp. 272–287.

[38] Sinha, A., and Chandrakasan, A. Dynamic power management in wireless sensor networks. *IEEE Design and Test of Computers 18*, 2 (Mar. 2001).

[39] Sohrabi, K., Gao, J., Ailawadhi, V., and Pottie, G. Protocols for self-organization of a wireless sensor architecture. *IEEE Personal Communications 7*, 5 (Oct. 2000), 16–27.

[40] Srivastava, M., Muntz, R., and Potkonjak, M. Smart Kindergarten: Sensor-based Wireless Networks for Smart Developmental Problem-solving Environments. In *The 7th Annual International Conference on Mobile Computing and Networking 2001* (July 2001), pp. 132–138.

[41] Tennenhouse, D., Smith, J., Sincoskie, W., Wetherall, D., and Minden, G. A survey of active network research. *IEEE Communications Magazine 35*, 1 (Jan. 1997), 80–86.

[42] Tilak, S., Abu-Ghazaleh, N., and Heinzelman, W. A taxonomy of wireless micro-sensor network communication models. *ACM Mobile Computing and Communication Review* (Apr. 2002).

[43] Woo, A., and Culler, D. A transmission control scheme for media access in sensor networks. In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking* (July 2001).

[44] Y. Huang, H. G. Publish/subscribe in a mobile enviroment. In *International Workshop on Data Engineering for Wireless and Mobile Access* (2001), pp. 27–34.

[45] Zhao, Y., Govindan, R., and Estrin, D. Residual energy scans for monitoring wireless sensor networks. In *IEEE Wireless Communications and Networking Conference (WCNC'02)* (Mar. 2002).