UNIVERSITY OF SOUTHAMPTON

Autonomous Energy Efficient Protocols and Strategies for Wireless Sensor Networks

by

Paritosh Padhy

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the
Faculty of Engineering, Science and Mathematics
School of Electronics and Computer Science

May 2009

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Paritosh Padhy

The aim of this research is to develop a model of a sensor network that will endeavour to monitor a hostile environment (one where communication within the network is difficult and the network entities are under risk due to physical damage). In this context, the study identifies the following key characteristics. A wireless sensor network (WSN) is a wireless network consisting of spatially distributed devices using sensors to monitor physical or environmental conditions at different locations. In addition to one or more sensors, each node in a WSN is typically equipped with a radio transceiver or other wireless communications device, a small micro controller, and an energy source, usually a battery. The size constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and bandwidth. Of these, energy is the most important since it is required for everything else. Thus, it directly influences the life-span of the nodes, hence, that of the system as a whole. Furthermore, the environment itself, where these sensor nodes are deployed, plays a big role in influencing the entire architecture of the network hardware platform and protocols that govern its smooth functioning. As a result the protocols required for governing the actions of the sensor nodes need to be designed accordingly.

Against this background, this research facilitates the developent of an environmental sensor network called GlacsWeb (deployed inside a glacier in Norway) which focuses on providing useful information about sub-glacial dynamics. GlacsWeb nodes are deployed under very hostile conditions. The strain from the moving ice may damage the nodes and the en-glacial water bodies may carry the nodes far out of transmission range from a centrally located base station. For these reasons GlacsWeb nodes have a high rate of failure. In order to effectively tackle this problem, this research develops GW-MAC (a Medium Access Control protocol) which focuses on efficiently connecting GlacsWeb nodes in an ad-hoc manner.

Moreover, the poorly understood nature of the glacier imposes further challenges in the area of sensing. Sub-glacial behaviour appears vary across the entire large mass of ice. For this reason, there is a strong need for nodes to make autonomous decisions to adapt their observation patterns and communication patterns accordingly to ensure maximum data is gathered with minimum consumption in energy. The study, therefore, develops USAC (A Utility Based Sensing and Communication Model for an Agent-Based Sensor Network), that provides a measure of utility by combining the task of both sensing and communication by the sensor nodes. The model, at first, develops a sensing protocol in which each agent node locally adjusts its sensing rate based on the value (importance) of the data it believes it will observe. Then, it details a communication protocol that finds optimal routes for relaying this data back the network base station based on the cost of communicating (derived from the opportunity cost of using the battery power for relaying data) it.

Both GW-MAC and USAC have been tested in simulation and have shown to perform better than other similar models.

Contents

List of Figures

List of Tables

Nomenclature

$S^i_{assigned}$	Slot assigned to node i in down-link mode
S^i_{uplink}	Slot assigned to node i in up-link mode
S_{frame}	Total number of time slots in a frame
Φ	Threshold time for nodes to switch to *listen only* mode
$T_{transmit}$	The time at which the data packet is transmitted
$T_{preamble}$	The time taken for the packet reach the receiver from the transmitter
T_{switch}	The time taken for the transceiver to switch on and stabilise
H	Length of the packet header
P	Payload of the packet
x_n	Data sample acquired at time step n
$\boldsymbol{t}$	Input vector consisting of M variables
$\boldsymbol{w}$	Weights assigned to each input variable t_j within the input vector
$p(.) or f(.)$	Probability density function of (.)
$\delta_{KL}(f_1, f_2)$	Relative entropy between probability distributions f_1 and f_2
β_i	The point at which a phase change i occurs in observed data
$\mathcal{I}$	A set of agents

K	Number of discrete radio transmission power levels
pt_i^k	Transmission power level k of agent node i
$n_i(pt_i^k)$	Set of neighbours of agent node i at transmission power level k
$t_i^j(data)$	Amount of time a data packet takes to transmit from i to j
Et_i^j	Energy consumed in transmitting data from node i to j
E_i^{sense}	Energy consumed by node i in sensing new data
$E_i^{receive}$	Energy consumed by node i in receiving a data packet
$v_i^{sense}(t_n)$	Value of the data sensed by node i at time n
$c_i^j(relay)$	Cost of forwarding a data packet from node i to j
$c_i^j(originate)$	Cost of originating a data packet from node i to j

Acknowledgements

First and foremost, I would like express my heartfelt gratitude towards my supervisors Dr. Kirk Martinez and Professor Nick Jennings for their continuous guidance, overwhelming support and constant encouragement throughout. Without their help and insightful advice, this thesis would lack structure and difficult to comprehend. It is only through their constant review and revision that my written word has become strong and my arguments clear. I would like to acknowledge some outstanding individuals with whom I have collaborated with during my research. I shall begin with Dr. Rajdeep Dash - a wonderful friend and colleague, who has carried me through the troughs of academic life and has always been willing to have lengthy discussions on several rough ideas. Others include: Dr. Royan Ong, Dr. Ahmed Elsaify and Dr. Gang Zou for their invaluable discussions on embedded systems engineering and radio transmission theory; Professor Jane Hart, Dr. Katherine Rose and Alistair Riddoch, all of whom provided me with invaluable knowledge about glaciology and the overall impetus behind the GlacsWeb project; and finally, Dr. Jigar Patel and Nimet Patel for their inspirational presence and creative chats that helped me remain positive.

I also gratefully acknowledge the financial support provided by the School of Electronics and Computer Science studentship and the GlacsWeb project. This meant that I was not hampered by financial constraints and could whole-heartedly dedicate my time to research. Additionally, the project has provided me with the context of my research, and a team consisting of Daniel Miles, Sarita Ward, Hannah Brown, Matthew Westoby, David Vaughan-Hirsch, Celine Regault, Natalie Jarman and Sarah Stafford - All of whom have helped me acquire data relevant to my study, have patiently listened to my work and provided me with constructive feedback.

I also owe a significant amount of appreciation for individuals outside of research and these include: Shivam Desai, Rupesh Patel, Raviraj Shah and everyone within the Senior Common Room at Glen Eyre Halls of residence - friends who have always been there to lighten up the mood during days of hardship. In particular, I am extremely grateful to Krupa Thakrar, who has been there at every hour of need and given me the confidence and strength to believe in myself.

In closing, I would like to thank my family for their support. My parents and my brother have taken have always stood behind me and their wisdom has been the founding stone for the way I think.

To my parents for their love and encouragement…

Chapter 1

Introduction

Modern computing has evolved from single computer systems to a large scale distributed network of such systems that are capable of providing on-demand access to a range of facilities. Prominent examples of such facilities include the Web (Berners-Lee et al., 2001), the Grid (Foster and Kesselman, 1999), peer-to-peer (Milojicic et al., 2002) networks and the ethernet (Spurgeon, 2000). This evolution has been fuelled by the need of the users to find and access information distributed across different systems, applications and even different geographic locations. Furthermore, increasing development and continuous miniaturisation in wireless and processor technology has dawned an era of ubiquitous computing where information processing is slowly but thoroughly being integrated into everyday objects and activities, and in doing so, engaging several thousands of devices and systems simultaneously. Potential applications include health and home care (Yang, 2006), environmental monitoring (Estrin et al., 2002) and intelligent transport systems (Zhang et al., 2005). Such applications have lead to a greater degree of user knowledge of, or control over, the surrounding environment, whether at home or in an office or a car. Ubiquitous computing is the third wave of computing technologies to transpire since computers first emerged. The first wave involved the *mainframe computing* era where one computer was shared by many people through workstations. The second wave involved the *personal computing* era where one computer was used by one person, requiring constant interaction. The third wave is seen as one person using many computers embedded in the environment allowing technology to recede into the background. This emerging wave of computing has brought with it a number of new challenges associated with the ability to model, design and build such complex distributed systems.

The number of entitities in such systems is steadily growing and there is an increasing need for distributed networks that operate under a decentralised control regime, that are open (individual components can enter and leave at will) and that contain a number of components representing distinct stakeholders with different aims and objectives. To tackle the challenges posed in such systems, it has been contended that agent-based approaches, with their emphasis on autonomous actions and flexible interactions, are a natural computational model (Jennings, 2001). In this context, a software *agent* can be viewed as an autonomous entity capable of flexible problem-solving. This thesis will consider an agent to be a goal driven software based autonomous computer system that will also have the ability to react to external stimuli and interact with other agents and humans (see Section 2.6.1). Now, agent-based solutions generally involve several agents (i.e. a multi-agent system (Wooldridge, 2001)) interacting to provide a solution for a given problem. Agents are well suited to a variety of applications (see Jennings (1999) and Luck et al. (2005) for a general review and Section 2.6 for a review specific to sensor networks) although they are particularly appropriate for applications that involve open, complex and distributed computation and communication across networked computers.

Typically, open systems are viewed as those that can be influenced by events outside the defined system boundary. In such systems monitoring the entitities and the interactions between them can be a difficult task. New entities can enter the system and existing entities from inside can leave the system at will. This characteristic makes it difficult for the entities to have knowledge about all others that are present within at any given time. With this uncertainty in in a multi-agent environment, these entities must be able to make decisions and successfully interact with others.

In such cases, examples of successful interactions include optimal partner selection for collaboration and task delegation to individuals who will perform the activity efficiently. Achieving such success is a challenging task, and therefore, this study will seek to develop an infrastructure that assures efficient interactions between entities within a system by addressing the uncertainty present in their decision making. This study will do so by addressing two fundamental design issues. First, it will specify the *protocol* that governs the interactions between these entities. This covers issues such as how their actions translate into an outcome, what range of actions are available and whether the interactions occur over a series of steps or are one-shot. Second, given the prevailing protocol, it defines the *strategy* (mapping from state history to action) for each entity.

The remainder of this chapter provides motivation behind the various concepts used to address the two issues described above. Having provided the motivation, Section 1.1 describes the running scenario that forms the basis of this research in this thesis, and research aims are presented in Section 1.3. The chapter concludes by providing a summary of key research contributions and an overview of the whole thesis.

1.1 Managing Actions in Human Societies

Distributed agent-based systems share a number of common problems with human societies, such as problems with communication, collaboration, negotiation and assurance of good interaction between individuals (human or agent). Now, humans exist in a society, and likewise agents exist in their own virtual society; it is within this virtual community that there is a need to assure efficient interactions. The main motivation behind this is the availability of only limited resources to the agents such as battery energy, information processing capabilities and communication bandwidth (discussed Section 2.1) that constrains them in their interactions with other agents. Therefore, when seeking to engender efficient interactions in distributed systems, it is natural to look at their human societal counterpart for inspiration. This is because, humans are cooperative in nature, yet at the same time have individual characteristics. They have the capability to act freely and take initiative even in societies which are governed by a higher authority. This is relevant to distributed systems where individual nodes are expected to make autonomous decisions whilst at the same time adhere to norms set for the benefit of the society as a whole. More specifically we examine the role that regulation, cost-benefit analysis and mutual cooperation play in assuring efficient interactions in a human society.

First let us consider the notion of regulation and guidelines. Often in human societies we make decisions based on our surroundings or the laws that govern our actions. For example, we may decide to drive past a cross roads junction; the decision to drive past this junction will typically include an assessment of the traffic lights (if any) and road traffic signs (if there are not any lights) by consciously looking out for other cars approaching the junction. Presumably, if we are not able to make an assessment of the traffic lights or the road signs (due to their lack of), we would find it difficult to observe other vehicles approaching the junction as we may not be expecting them and this may result in an accident. In our example, we hope to consult a controller mechanism (traffic lights) or we can make use of existing guidelines (the road signs) to help us cross the junction safely. In the case of the former, with the traffic lights installed at the junction, we can expect to cross with full confidence. In the case of the latter, we can be cautioned about any oncoming danger and informed as to who has the actual right of way so that accidents can be avoided.

More generally, it can be said that most actions taken by humans are rationalised based on the concept of *regulation*. Here, regulation is simply a set of rules or laws structured by a higher authority and it forms an integral component of our reasoning and decision making. In this context, following these structured set of decrees helps us make appropriate decisions. The concept of regulation is illustrated in our example of traffic lights. When we approach the junction, we follow the traffic rule that red means stop and green means go. In this way, the regulation of the traffic signal ensures that the junction is controlled safely. However, in an isolated area (for example a rural village) a traffic signal might not have been necessarily set up. In this case, we cannot rely on a central mechanism to control the approaching vehicles but we still have to ensure that accidents do not occur. Here, traffic can be controlled through road signs indicating an approaching junction, highlighting which one is the minor road and which one is the major road and/or by placing a roundabout to avoid collisions. These can be used as *guidelines*, which may be used in the absence of rules.

As we have discussed, regulation and guidelines streamline our actions. However, they do not explain why we take those actions in the first place. Thus, we now turn to the issue of cost benefit analysis since this plays a huge role in our decision making. As humans, we implicitly evaluate the benefit and consequences of our actions before we actually execute them. For example, our taxi may be driving across the junction because it is important for us to reach some destination beyond. Exactly how important it is for us to reach that particular destination also depends on the purpose of our visit and how much it would cost us to get there. If we are on our way to a train station, just for fun, but are aware that the cab fare and train ticket is going to be very expensive, then we would simply turn back and return home (a better idea would be to never leave home in the first place). However, if we intend to take the train to visit the nearest hospital in order to undergo a major operation or to meet family members who are unwell, then we would seek to reach there at any (affordable) cost. Simply put, we are willing to spend as much as we can, on available resources, as long as we think that the consequences of our actions would be worth the buy. On the other extreme, we are shy spenders (unless we have a vast amount of wealth) and like to save our money for a future event that might demand better use of it.

Now, while our budget plays an important role in determining whether we should go ahead with our actions or not, we cannot exclusively rely on it. Prior to budgeting, it is imperative to figure out means for us to take those actions. In other words, talk of the budget would not exist if we did not have access to the taxi or train (assuming it was impossible to get to the destination alternatively). Thus, in our example, we are relying on the cab driver to get us to the train station and the train to take us to our destination. In normal society we ought to pay the cab driver a certain amount of money in return for the service he or she provides. However, if the cab driver is a best friend or a close relative, he or she would be glad to take us to the station without any charge, thereby cutting our travelling costs and still help us to get to our destination. Furthermore, if the cab driver requires a certain parcel that needs to be delivered to someone at an intermediate train station during the course of our journey, we would readily oblige to the task since it would not cost us any more to do it. The point here is that, individuals in society thrive on mutual cooperation. However, they only do so as long as they feel affiliated to a common establishment which in turn facilitates their own ability to achieve personal goals (Binmore, 2006).

In summary, this section has discussed how in society, rules and regulations (set up by a higher authority), cost-benefit analysis of actions (from every individual's perspective) and mutual cooperation (between those individuals) help address the uncertainty in making their decisions. These simple concepts can be applied to a distributed system of connected workstations where resources such as communication bandwidth is scarce.

Here, communication of data packets over a solitary medium such as a wire or air is anologous to cars crossing the road junction. Estimating the importance of a data packet and then evaluating it against the energy expenditure for transmission is comparable to the cost-benefit analysis amongst humans. Finally, data packets may have to pass through intermediate routers if travelling long distances and this concept is equivalent to mutual cooperation. Bearing in mind the main challenges of distributed systems lie in connecting workstations and resources in a transparent, open, scalable and fault-tolerant way, this study focuses on a special subset of such systems called *Sensor Networks* which shall be discussed in the next section.

1.2 Running Scenario: Sensor Network

Ubiquitous computing (also a form of pervasive computing) has been in development for close to two decades and although a few core technologies such as weather beacons and traffic sensors have already emerged, it still remains some way from becoming a fully operational reality. This is because of the particular challenges posed in the development of battery powered technologies and user interfaces. Sensor networks are perhaps the most prominent example of such an environment and, for this reason, is the research area this thesis shall be primarily focusing on.

A sensor network is a network of small sensor nodes where each node typically consists of a micro-controller, a radio transceiver, a power supply and one or more sensors for sensing the physical environment (Steere et al., 2000; Delin et al., 2005; Chong and Kumar, 2003; Lorincz et al., 2004). As such, they may require decentralised control regime (pertaining to both the way that the sensors perform their tasks and where information is distributed amongst the nodes) and are potentially open systems with several distinct stakeholders. Hence, they provide a compelling area for the application of multi agent systems since they are dynamic systems where decisions and actions have to be executed at several points (this is discussed in greater detail in Chapter 2). Specifically, each agent (residing in the micro-controller) makes decisions with regard to the following aspects:

1. Task Scheduling. The agent node decides the timing and nature of the sensing task which should be carried out by the attached sensor.
2. Resource Allocation. The agent node decides on the apportionment of the limited resource (for example power, bandwidth and/or computational resource) between the different tasks it may be required to carry out.
3. Communication Protocol. The agent node decides the source nodes it wishes to receive data from and destination nodes it wishes to forward data to. It also decides which data to transmit.

Thus, a sensor network, where each node has the potential to make autonomous decisions on its own actions and resource usage, can be naturally represented as a multi agent system. Now, in cases where all sensor nodes are owned by a single stakeholder, this could be modelled using a cooperative multi agent system approach in which agents are designed to work in tandem towards the system goal (Padhy et al., 2006; Deshpande et al., 2005). However, there are also an increasing number of applications emerging where each node (or a group of nodes) may be individually-owned by various stakeholders. Examples of such a scenario include traffic control application (where each node is owned by a particular vehicle) (Wu et al., 2005), pico-satellite projects where several companies own their own satellites monitoring a particular area (Heidt et al., 2000), and in disaster relief examples where different organisations share information gathered by their own sensor nodes to coordinate efforts in a natural disaster (Jennings et al., 2006). In such applications, the nodes operate competitively instead of cooperating, and, as such, attempt to optimise their own gain from the network at a cost to the overall performance of the entire network.

In both such systems (selfish and cooperative), there are two important design issues that need to be addressed. First, there is a need to specify a set of rules that govern the activities of the nodes. These cover issues such as how the actions of the agents translate into an outcome, what range of actions are available to the agents, and whether the interactions between agents occur over a series of steps or are one-shot and so on. Second, given an existing rule, a strategy (mapping the state history to action) needs to be defined for each agent node. Although it is possible for the system designer to enforce both rules and strategy for each agent node in an environment (Cao et al., 1997; Ogren et al., 2004) where all entities cooperate to achieve an overall network objective, it is not always the case in an environment where nodes adopt a selfish approach since they are owned by different stakeholders. Given this, a system designer can either impose a set of rules (to guarantee certain properties of the system) (Rogers et al., 2005) or design a strategy space for each agent maximising its benefit (Dash et al., 2005) within the system but not both. For this reason, the research investigation in this thesis will be based on a cooperative multi-agent system instead of a network comprising selfish agents so that we can extensively cover both design issues.

A typical sensor network involves several nodes making various observations (sensing) about the environment in which they are deployed. To this end, scientists are realising the need to expand to the point where information from numerous such sensor networks will be aggregated at higher levels to form a picture of a system at resolutions that is not feasible with current technology. In order to achieve this objective, issues relating to power, communications, deployment and maintenance of such networks need to be resolved. In particular, these issues are:

1. Nodes may be deployed in a harsh environment where communication channels

are weak (due to obstacles or distance) and failure rate is high (sensors may be physically broken or get carried far away by factors relating to wind, water or terrain movement). The physical adjustments required to protect the node and network is discussed in Chapter 3 and 4.

2. Nodes are required to communicate data to a central sink node or base station (one that collects data from the network). Now, an organised structure for communication may not exist in such a scenario leading to all nodes transmitting data simultaneously (in which case the sink node would not be able to understand any communication). This scenario is addressed in Chapter 4.

3. Nodes are typically constrained by the amount of power which is available to them. This leads to the node being able to carry out only a few of the total tasks that are demanded. As a result, chapters 4 and 5 discuss implementation of mechanisms where nodes are incorporated with limited capacity.

4. Nodes are not aware about the topology of the entire network but only their neighbours. In such a scenario, if a particular node requires to communicate to the base station, it may not know which neighbour to communicate via. This leads to uncertainty in deciding the best path of communication (one where data gets communicated to the intended destination the quickest whilst consuming very little energy). This aspect is covered in Chapter 5.

5. Nodes observe the environment at static time intervals. This can either result in missing observations of crucial events (in an environment that changes frequently) or unnecessary energy waste in activating sensors (in an environment that changes very slowly). Thus, there is a strong need for an adaptive sampling mechanism that will enable nodes to make observations effectively and efficiently. Furthermore, nodes are not aware of the importance of their observed data. They treat all observations equally and incur the same cost of transmission for all of them irrespective of their worth. This is highly inefficient and a data valuation model needs to be considered. These aspects are covered in Chapter 5.

Against this background, this thesis argues that there is a strong need for appropriate measures to be in place for a sensor network to work in an efficient manner. Specifically, these measures can be utilised to account for uncertainty within the environment and the willingness and capability of all entities to perform actions in a cooperative manner such that overall network objectives are fulfilled. Having thus provided the scenario for this thesis, the next section provides an overview of the aims this research.

1.3 Research Aims

We have come to establish that this study involves investigating and understanding the same resource management problems faced in distributed computing. However, this thesis will be focusing its work in the context of a *dynamic* (one where the environment and network conditions continuously evolve) wireless sensor network. Until now, researchers have developed several models for autonomous wireless sensor networks (See Chapter 2 for a review); however, most of these models tend to focus on a specific aspect of the wireless sensor network and overlook other components rendering the system, as a whole, difficult to implement. Therefore, the general aim of this thesis can be stated as the following:

To develop a mechanism, which takes into account environmental and other available information, and aids decision-making amongst nodes in open and dynamic environments, particularly in relation to the formation and management of an agent-based wireless sensor network.

In order to achieve this general aim, the research in this thesis will revolve around GlacsWeb (Martinez et al., 2004), a real life environmental sensor network project that was developed at Southampton University in UK and deployed in Brikdalsbreen glacier in Norway (see Chapter 3 for more details). In particular, this general aim of our study and the research project it is strongly tied to can be broken down as follows:

Aim: 1
Establish a network that miniaturises the physical design of the sensor nodes. More specifically, aim to install all necessary modules (battery, microcontroller, radio etc.) required to formulate a node in a *small compact* architecture.

Aim: 2
Most sensor nodes are limited in their energy source (batteries). Therefore, it is essential that the node hardware modules do not consume too much power in their operational state. For this reason, the thesis will aim to design sensor nodes that operate under very *low power* conditions.

Aim: 3
Further to the low power hardware design of the sensor nodes, it is equally important to address the rules that govern the interaction between the nodes. For this reason the study will aim to develop protocols, both at data-link and network routing level such that nodes *minimise communication* in a resource efficient manner and *prolong their lifetime*.

Aim: 4
In open systems, and especially in a dynamic sensor network, it is likely that nodes

will be required to interact with others they have not yet had experience with. For this reason, this study will aim to enhance the above model by designing, in order to form an up-to-date topology, an infrastructure that will enable the network nodes to *operate normally even when nodes enter or leave the network.*

Aim: 5
The context of sensor networks in general, introduces certain problems that the mechanism must address. One such problem is the non-deterministic nature of the network; at any point in time it is difficult to determine which nodes are accessible and which are not. To this end, it is essential that the mechanism can cope with the absence of nodes it relies on, and is therefore distributed and robust (able to cope with network failure) due to the nature of its application domain. Furthermore, the large number of nodes within the sensor network, and the ability of this number to grow dynamically, requires the mechanism to be *scalable*.

1.4 Research Contributions

The research reported in this thesis stems from the analysis and design of a real life sensor network and the protocols that govern the interaction between the sensor nodes in such a network. This research thus provides the following insights into these two crucial aspects of sensor networks:

- **Designing a Hardware Platform:** Designing a hardware platform is a big challenge in the field of sensor networks. This takes accoung of the physical design of the system that includes the size of the nodes, the choice of sensors and microcontrollers to be installed, the preferred medium of communication (radio, wire etc.) and so on. It is imperative to design a new platform if there is a specific problem in mind for which existing platforms would not suffice. Furthermore, if cost and size become a limiting issue, the need for a new platform becomes more prominent. In this context, this study will discuss a real life environmental sensor network that was designed for deployment underneath a glacier and why some of the off-the-shelf platforms(Chapter 3) were not chosen. Here, the thesis considers the benefits of having an indigeneous platform and how it enables research effort of this thesis.

- **Communication Protocols:** With new hardware platforms allowing a large number of communication components to be integrated onto a single chip, small sensor nodes are able to connect via radio links, and are able to perform tasks which traditional (wired) single sensor nodes find hard to match (For example, detecting danger spots in a disaster area or in our case monitoring the inhospitable and dynamic sub-glacial environment). This ability to connect an untethered collection of sensor nodes to a fixed centre requires constant collaboration amongst nodes. Here, communication amongst nodes is facilitated by protocols that keep them connected in an energy efficient manner. In such cases, the study finds that traditional communication protocols are not suitable for low-power sensor networks. Furthermore, many existing sensor network communication protocols are designed for specific hardware platforms and applications and not suitable for our glacial sensor network. Therefore, novel protocols are designed (in Chapters 4 and 5) to deal with such an application.

To summarise, the work described in this thesis addresses a number of issues that arise when designing a sensor network for monitoring a sub-glacial environment. In effect, the aim is to address the challenges faced in a real sensor network (as opposed to over simplified theoretical assumptions found in literature) and devise solutions in that regard. Specifically, the state of the art is advanced in the following way:

1. Identifying challenges in a real sensor network. We design a custom made wireless sensor network (GlacsWeb) with nodes to monitor and observe a sub-glacial environment. In this case the study describes the solutions to power management, radio communications and other challenges faced in the system together with a discussion of the performance of the final system. The nodes deployed within the glacier provide 18 months of observed data, which presents an insight into the design decisions of the final system. Having installed different versions of the system in successive years (2003-2006) this work further discusses the lessons learnt from installing a real life environmental sensor network in an extreme environment (Padhy et al., 2005; Martinez et al., 2006).

2. GW-MAC: A Medium Access Control Protocol for GlacsWeb. The research develops an energy-efficient medium access control protocol that takes into account the physical layer properties of GlacsWeb's hardware platform and the specific requirements of the deployed network to provide a robust and energy efficient communication link between nodes. The general architecture of GWMAC is based on scheduling and time division multiple accesses (TDMA). GWMAC is based on a centralised Time Division Multiple Access (TDMA) that completely eliminates collisions, overhearing and idle listening. In doing so, we perform extensive series of simulations to evaluate our claim (Elsaify et al., 2007).

3. Utility Based Sensing and Communication Protocol. The study develops a utility-based mechanism to manage sensing and communication in cooperative multi-sensor networks such as GlacsWeb. In this context, we first develop a sensing protocol in which each sensor node locally adjusts its sensing rate based on the value (indicating importance) of the data it believes it will observe. In addition, the work details a communication protocol for a medium size network of up to 40

nodes that finds optimal routes for relaying this data back to the sink based on the cost of communicating it (derived from the opportunity cost of using the battery power for relaying data) (Padhy et al., 2006).

The research carried out in relation to this thesis has resulted in the publication of the following papers which are reported within this thesis:

- Padhy, P., Martinez, K., Riddoch, A., Ong, H. L. R. and Hart, J. K. (2005). Glacial Environment Monitoring using Sensor Networks. Proc. of 1st Real-World Wireless Sensor Networks Conference, Stockholm, Sweden. (Chapter 3)
- Hart, J.K., Martinez, K., Ong, H.L.R., Rose, Kathryn C., Padhy, P (2006). A wireless multi-sensor subglacial probe: design and preliminary results. Journal of Glaciology. Volume 52 (178). 389 397. (Chapter 3)
- Martinez, K., Padhy, P., Elsaify A., Zou G., Riddoch, A., Ong, H.L.R., Hart, J.K. (2006). Deploying a sensor network in an extreme environment. Proc. of IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC 2006), Taichung, Taiwan. (Chapter 3)
- Elsaify A., Padhy, P., Martinez, K., Zou, G. (2007). GW-MAC: A TDMA based MAC protocol for a glacial sensor network. Ajunct poster/demo proceedings of 4th European Conf. on Wireless Sensor Networks, Delft, Netherlands. (Chapter 4)
- Elsaify A., Padhy, P., Martinez, K., Zou, G. (2007). GW-MAC: A TDMA based MAC protocol for a glacial sensor network. Proc. of 4th ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks. Chania, Crete Island, Greece. (Chapter 4)
- Padhy, P., Dash, R.K., Martinez, K. and Jennings, N.R. (2006). USAC: A multi-agent model for a glacial sensor network. Proc. of 4th Fourth European Workshop on Multi-Agent Systems, Lisbon, Portugal. (Chapter 5)
- Padhy, P., Dash, R.K., Martinez, K. and Jennings, N.R. (2006). A utility-based sensing and communication model for a glacial sensor network. Proc. of 5th Int. Conf. on Autonomous Agents and Multi-Agent Systems, Hakodate, Japan. (Chapter 5)
- Padhy, P., Dash, R.K., Martinez, K. and Jennings, N.R. (2007). A utility-based sensing and communication model for a glacial sensor network. Ajunct poster/demo proceedings of 4th European Conf. on Wireless Sensor Networks, Delft, Netherlands. (Chapter 5)

1.5 Thesis Structure

This section outlines the structure of the report giving a summary of the work presented in each chapter.

The study begins with a review of the relevant literature in the field. The purpose of the review is twofold. Firstly, it serves the purpose of providing the reader with background to the problem and its domain. Secondly, through the review of the literature the study elicits a list of requirements for the sensor network model it aims to develop. More specifically, Chapter 2 begins with a general discussion of sensor networks and its limitations. Building upon this, through the analysis of the various layers of abstraction in a sensor network, the study enumerates a number of issues that most sensor network applications encounter. In doing so, it also examines the state of the art in sensor network protocols from a variety of domains, to discover how some of the previously enumerated issues are addressed. Finally, the thesis concludes with a summary which presents a list of requirements of the required sensor network model.

Chapter 3 presents GlacsWeb - a novel, real-life, deployed environmental sensor network. The chapter provides a description of the core components of the network alongside a report on the implementation strategies in light of the challenges faced during deployment. It also discusses the importance of design factors that influenced the development of the overall system, its general architecture and communication systems. This chapter essentially takes shape of a mini-case study thereby forming the basis of further research in Chapters 4 and 5. It concludes with a summary that includes aims that are specific to this research.

Chapter 4 investigates the importance of multi-hopping and ad-hoc networking within a shared-medium network such as GlacsWeb. In such a network where several nodes are independently sensing, transmitting and relaying each others data, it is extremely important to have a data-link infrastructure to ensure that node-to-node communication is carried out in a reliable, fair and power-efficient manner. This chapter presents the design of an efficient medium access control protocol for the ad-hoc probes deployed within the GlacsWeb network with its primary aims including achieving multi-hopping of data without any radio packet collisions.

Chapter 5 takes a much higher level approach and investigates the use of utility functions within GlacsWeb in order to maximise the data gathered by the sensor nodes, whilst minimising their power consumption. It presents USAC, which is a novel sensing and communication model for agent based sensor networks. More specifically, this chapter describes how an agent within the sensor network combines the task of sensing and communication via a utility function in order to minimise power consumption and maximise the value of data collected at the base station.

Finally, Chapter 6 summarises the main achievements of this thesis and how well they satisfy the requirements discussed in this chapter. It also discusses the broad future research direction that has been identified for GlacsWeb and other similar applications.

Chapter 2

Literature Review

The aim of this research is to develop a model of a sensor network for a particular application discussed in Chapter 3. However, before we go about formulating an effective solution in order to develop this model, we need to review the necessary literature concerning wireless ad hoc networking techniques.

Section 2.1, provides an introduction to sensor networks. This section also highlights why protocols and algorithms proposed for traditional wireless ad-hoc networks are not well suited to the unique features and application requirements of sensor networks. In doing so, it will provide the background for the work for this research by positioning it in light of the challenges that need to be addressed when designing a sensor network.

Following this, Section 2.3 provides a brief overview of the sensor network protocol stack where the different layers of abstraction, for various protocols that govern the interaction between entities of a sensor network, are identified. This overview forms a short summary of sections 2.4 to 2.8 which provides a detailed discussion of the work carried out in literature for each layer in the protocol stack.

The chapter concludes in section 2.9, which highlights the open issues not addressed by the state of the art, and provides a summary of detailed requirements for the research.

2.1 Introduction to Sensor Networks

This thesis focuses on the study of wireless sensor networks and any references made to sensor networks imply a wireless medium involved unless otherwise stated specifically. Sensor networks are different from traditional ad hoc networks. They are usually composed of nodes several orders of magnitude higher than the nodes in an ad hoc network. These nodes consist of four basic units embedded on to them which are categorised as

1. **Sensing unit.** This is considered as the primary unit of the node because it fulfills the principal objective of observing the target or environment. It may consist of one or more sensors to track various characteristics such as vision, temperature et cetera.

2. **Processing unit.** This is typically a micro-controller that is responsible for performing controlling functions on the sensor node. Some typical tasks include activating sensors, converting analogue sensor measurements to digital signals, storing data in temporary memory, aggregating raw data, enabling and disabling the transceiver for communication and generating clock cycles to keep track of time.

3. **Transceiver unit.** This unit comprises of a combined circuitry of a transmitter and a receiver. Both transmitting and receiving actions consume energy however a node cannot transmit and receive data packets simultaneously.

4. **Power unit.** This is typically a battery and is usually the only source of energy for the sensor node. All the remaining units of the node are heavily dependent on this unit to conduct their operations.

In addition to these units, the nodes can also comprise of application dependent components such as a location finding unit or a mobilising unit.

Sensor nodes are deployed to take advantage of the physical proximity to the actual target (place of an event in the environment) they monitor or sense to simplify signal processing. Most of these nodes are prone to failure either due to limited power supply or simply because they are subject to hostility faced by external factors in the environment. For this reason, these nodes may have to be densely deployed. Furthermore, for specific applications (Martinez et al., 2004; Britton and Sacks, 2004; Mainwaring et al., 2002), the position of the sensor nodes also may need not be predetermined or engineered during deployment in order to gain maximum exposure to the events being monitored.

Now, nodes can often be deployed in harsh environments leading to a high failure rate. This, coupled with their wireless nature, alters communication links within the network and implies that the topology of a sensor networks in such environments is always prone to change. Although communication between nodes takes place as peers, these nodes also use a broadcast communication paradigm. This means that all nodes within receiving vicinity of a transmitting node can hear the transmission. Unreliable communication links and lack of an infrastructure (central controlling mechanism) makes it necessary for these nodes to have self-configuring characteristics in order to establish a network structure. Data from these nodes can be aggregated and usually collected at a special type of node called the *Base Station* or the *Sink*. This node is the gateway to external networks where data can be sent for further analysis.

Continuous advancements in wireless technology and miniaturization have made the deployment of sensor networks for various applications increasingly feasible. Examples of such applications include Jet Propulsion Laboratory's sensor web in Antarctica (Delin et al., 2003) to monitor micro-climate, Huntington Garden's sensor web project to monitor botanical conditions in an urban environment (Delin et al., 2005), the CORIE observation system to advance the understanding of Columbian river-dominated estuaries and plumes (Steere et al., 2000), examining sub-glacial sediment to follow glacier dynamics (Martinez et al., 2004), monitoring fences for security (Wittenburg et al., 2007), sea-bird habitat monitoring at Great Duck Island (Cerpa et al., 2001) and capturing infra-sound data to monitor volcanic eruptions (Lorincz et al., 2004). The latter two are discussed in greater detail towards the end of Chapter 3. The research efforts in these projects are constantly thriving to a pervasive future in which sensor networks would expand to a point where information from numerous such networks (for example glacier, river, rainfall, avalanche and oceanic networks) could be aggregated at higher levels to form a picture of the environment at a much higher resolution. However, all these networks have some basic short-comings that greatly hinder their functionality and are discussed next.

2.2 Limitations of Sensor Networks

The above mentioned examples like most sensor networks are dedicated to a single application and the nodes are subject to the following fundamental limitations:

1. **Energy constraints.** Each sensor node can only be equipped with a limited power supply before deployment. Thus, its life time is heavily dependent on the battery capacity. The sensing, processing and transceiver units all consume energy to carry out their respective tasks. However, most energy is consumed by the transceiver unit for communicating data. Sohrabi et al. (2000) state that whilst 3J of energy is needed to transmit 1kb of data over 100m, the same energy is required to perform up to 300 million instructions in a processor with a modest specification of 100 MIPS. This clearly suggests that the nodes should carry out more processing actions on their data to minimise transmission of raw so that considerable energy can be saved. Now, recently energy harvesting in sensor nodes has attracted a lot of interest. These have involved nodes scavenging energy from solar cells (Roure, 2005), wind turbines (Weimer et al., 2006) and from their own kinetic energy created as a result of external forces (Beeby et al., 2006). Although these studies have shown to reduce the energy constraint on nodes, it is not feasible for many networks where nodes are quite often sheltered from these natural sources of energy such as sun and wind. For this reason, this thesis chooses to focus on a network where energy harvesting is not possible and it is safe to assume that a node depleted of its battery supply can no longer be used and therefore its failure and can cause significant topological changes. Nodes may not only be responsible

for originating their own data but may also assist in relaying data from other nodes using several short distance communication links (multi-hop) that does not require much energy. However, failure of intermediate nodes may imply that those far away may have to increase their transmission power to get their data transmitted successfully to the base station at a higher energy cost effectively reducing the lifetime of the network. It is therefore imperative for the nodes to use the limited energy available to them in an efficient manner.

2. **Processing limitations.** The processing power of sensor nodes is directly linked to the amount of available energy in sensor networks. A constraint on the available energy also constrains the use of computational power. Although the processing unit consumes far less energy than the transmission unit of a node, most wireless sensor nodes are run by single low power micro-processors such as PICs (Martinez et al., 2004), MSP430 (Yang, 2006) and Atmel Atmega (Mainwaring et al., 2002). The limited computational power restricts developers to computationally cheap algorithms, especially when deployed for the purpose of real time monitoring.

3. **Communication limitations.** Due to the large number of nodes and the potentially huge spatial spread between them, radio is the preferred method of establishing communication and interaction between node transceivers. The broadcast nature of radio implies that all nodes within a given range are able to receive signals from a transmitting node. This can be very problematic when there is a large number of unattended nodes. Many nodes may transmit at similar times causing data to be corrupted. More so, since communication consumes the most power in a sensor node it is essential that the transceiver unit is used scarcely. Furthermore, bandwidth can be very limited in certain applications and the different transceivers must be used efficiently such that each node gets fair but limited access to the communication channel. Although many applications take advantage of short range multi-hop communication between nodes instead of long range communication to conserve energy, it is inevitable that communication activities in many networks may leak precious energy in various forms. Ye et al. (2002b) have identified four major sources of energy waste in such communication activities. These are highlighted below:

    - **Collision.** This happens at a receiving node when it simultaneously receives packets from two nodes transmitting at the same time. The received packets are corrupted and the transmitting nodes need to be notified to send the packets again.
    - **Overhearing.** This phenomenon occurs when nodes receive packets that are not addressed to them. Take for example the following example. Assuming there are n nodes within transmission range of each other. Then the total energy consumed for one packet transmission would be $E_{transmit} + (n-1)*$

$E_{receive}$ where E is energy. This is a huge waste since total energy should not be more than $E_{transmit} + E_{receive}$.

- **Control Packet Overhead.** In an ad-hoc sensor network, it is sometimes essential for sensor nodes to communicate long data packets with the help of short control packets. These control packets serve various purposes that include reserving the medium for transmission, acknowledging receipt of packets and requesting permission to send packets. This adds to the energy cost of the nodes.
- **Idle Listening.** In addition to the energy consumed in receiving and transmitting packets, a radio module also consumes energy in its idle mode. In this mode, the transceiver is simply turned on but it is not busy transmitting or receiving packets. Studies have observed that in idle mode, a radio module consumes $50-100\%$ of energy required for receiving. Stemm and Katz (1997) observed the $idle : receive : send$ ratio for hand held wireless devices to be $1 : 1.05 : 1.4$ while the Digitan2 Wireless LAN module specification shows that $idle : receive : send$ ratio is $1 : 2 : 2.5$ (IEEE, 1999). At the time of writing this thesis, some wireless devices such as the CC1100 module (Incorporated, 2008) have began to exhibit a ratio of almost $1 : 1 : 1$.

All sensor networks in practice have to put up with the above limitations. Of these, the constraint in energy is most important since every task that can be performed by the node is dependent on it. Thus, energy directly influences the life-span of the nodes and, hence, that of the network as a whole. Most sensor networks strive to minimise their energy usage in order to maximise their lifetime. This is a challenging task and primarily involves designing efficient communication and sensing protocols that significantly reduce energy leaks. The following section provides a review of several models proposed in literature to address this issue.

2.3 Sensor Network Protocol Stack

It is common knowledge that underlying electronic design of any system dictates the main energy usage. The focus in this thesis, however, is on the protocols that govern the behaviour of the hardware. In efforts to minimise energy consumption in a sensor node, it is important to understand the software implementation of a sensor network protocol suite first. The implementation of the sensor network protocol suite can be visualised through the protocol stack (Akyildiz et al., 2002) shown in Figure 2.1. The protocol stack is also known as the *communications stack*. Individual protocols within the suite are usually designed with only one primary objective and the modularity offered by the stack makes their design and evaluation easier. Each protocol module communicates with only two others and hence are commonly imagined as layers in a stack of protocols.

The stack shown in Figure 2.1 combines power and routing awareness, integrates data with networking protocols, communicates power-efficiently through the wireless medium and promotes cooperative efforts of the sensor nodes. The physical layer focuses on the modulation, transmission, and receiving techniques. The data-link layer is responsible for medium access control and consists of protocols that must be power aware and are responsible for minimising collisions, overhearing activities and idle listening. The network layer addresses the issue of calculating the best possible data routing path from one node to another as required by the application layer. The task management plane monitors task distribution among the sensor nodes. All the sensing tasks and control of power is coordinated by this plane.

Having briefly highlighted the main layers of the sensor network protocol stack we proceed further to discuss each one in greater detail. The following subsections provides greater insight into the relevant research conducted in literature for each layer in the stack.

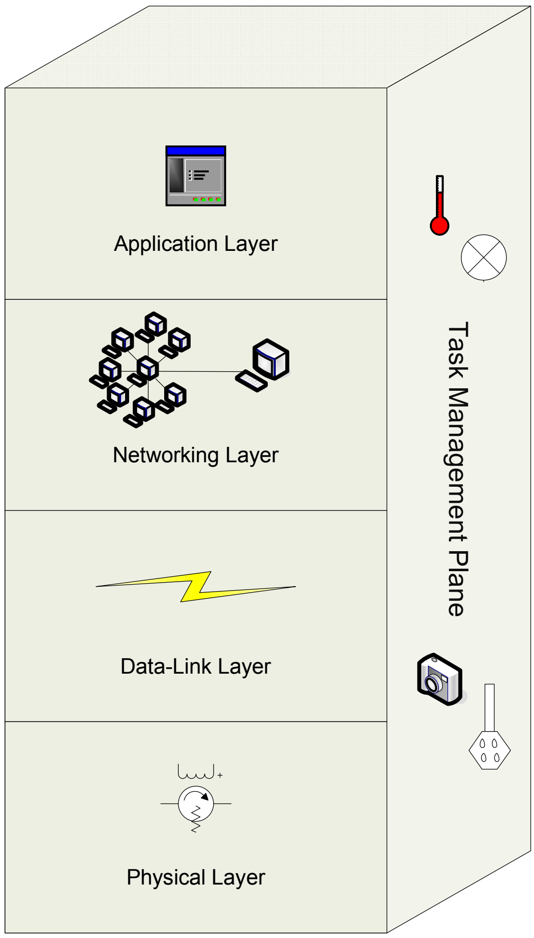


FIGURE 2.1: Protocol Stack of a Sensor Network.

2.4 Physical Layer

This layer is accountable for the selection of an appropriate frequency for communication, generating the carrier frequency, detecting signals, modulating the data and encrypting the data. It is therefore concerned with the underlying hardware and transceiver design which is beyond the scope of this research study. This study is concerned with much higher level issues and aims to focus on a more software based approach. This section is therefore, very brief and discusses the power efficiency and modulation schemes for wireless communication within a sensor network.

On the one hand, long distance wireless communication requires the transmitter to use more power and is therefore more expensive in terms of energy. Multi-hop communication over shorter distances in a sensor network can help reduce power consumption of nodes and also effectively overcome shadowing and path loss effects, if the node density is high enough. On the other hand, a multi-hop scenario can potentially lead to higher packet loss rates and, thus, higher enery consumption (See beginning of Section 4.3 for an experiment that validates this). For this reason some sensor network applications adhere to a minimum hop regime.

In terms of signal modulation a lot of work has been carried out in this layer. M-ary and Binary modulation schemes (Shih et al., 2001) have concluded that although the former can reduce the transmit on-time by sending multiple bits per symbol it results in complex circuitry and increased radio power consumption. Under start-up power dominant conditions, the binary modulation scheme is more energy-efficient. Recently, resilience to multi-path, low transmission power and simple transceiver circuitry of Ultra wide band (UWB) or impulse radio (IR) (Aiello and Rogerson, 2003) has drawn considerable interest in sensor networks. UWB employs base band transmission and thus requires no intermediate or radio carrier frequencies.

Although the physical layer is largely unexplored in the area of sensor networks it is mainly concerned with transmitter design and modulation schemes. This is not directly related to protocols and strategies that are intended to develop for a model of a sensor network and therefore we proceed to discuss the next layer.

2.5 Data-Link Layer

It is the responsibility of the data-link layer to ensure reliable point-to-point and point-to-multipoint communication in a sensor network. Hence, it carries the task of multiplexing of data streams, data frame detection and medium access control (MAC). This subsection will primarily discuss some MAC strategies for sensor networks and how they differ from traditional IP networks.

MAC protocols are developed to achieve two goals:

1. **Create Network Infrastructure.** The presence of vast number of nodes in a sensor network implies that there is a strong need to establish communication links for data transfer. MAC protocols help form basic hop by hop infrastructure that provides the network with self-organising ability.

2. **Share Communication Resources.** With so many nodes competing for the medium to communicate with one another, it is extremely important that communication resources are shared fairly and efficiently. This allows all nodes equal opportunities to transmit data.

Sensor networks are different from traditional wireless networks such as cellular (Lee, 1995), Bluetooth (Gratton, 2002), and mobile ad hoc networks (MANETs). The objective in these networks is to optimize throughput and delay (explained later). Although MANETs share similar characteristics of ad hoc deployment and self-configuration of the nodes like in sensor networks, provision of high quality of service (QoS) and bandwidth efficiency takes primary importance with such protocols.

Furthermore, the battery powered nodes in traditional networks can be easily replaced by the user and hence conservation of power is only given secondary importance. This is impractical for a sensor network as the main objective is to prolong a sensor node lifetime. Algorithms and protocols developed for traditional wireless networks include time division multiplexing with frequency hopping (Bluetooth), Z-Wave (Alliance, 2005a), Zigbee (Alliance, 2005b) and the IEEE 802.15.x suite (Gutierrez et al., 2001). Such protocols aim to solve existing specifications that are much different from that of a sensor network.

For this reason a need for new MAC layer protocols for sensor networks arises. (Ye and Heidemann, 2003) provide a list of attributes and trade offs in a sensor network that need to be considered before designing an appropriate MAC protocol. These are listed below in order of their importance:

1. **Energy Efficiency.** Prolonging a sensor node lifetime is a critical issue when a large number of nodes are involved and it is difficult to change batteries or recharge. Since radio is the biggest power consumer, the MAC layer directly controls the transceiver activities.

2. **Collision Avoidance.** This is a fundamental task of all MAC protocols. Collisions usually occur with contention based protocols. Although most contention based protocols accept some level of collision they all strive to avoid frequent ones.

3. **Scalability and Adaptivity.** The size, density and topology of a sensor network keeps changing frequently. This is attributed either to the high failure rate of nodes, the mobile nature of nodes and deployment of new nodes. A good MAC protocol should accommodate such changes gracefully.

4. **Channel Utilisation.** This is normally a secondary goal in a sensor network and reflects on how well the bandwidth of the channel is utilised in communication. In traditional networks bandwidth is the most valuable resource and service providers strive to accommodate as many users as possible. In contrast, the number of nodes in a sensor network is determined by the application.

5. **Latency.** The importance of this attribute usually depends on the application. Sensor networks used for surveillance applications are less tolerant to messaging latency. Such applications can tolerate a little messaging latency because the network speed is typically orders of magnitude faster than the speed of a physical object. The speed of the sensed object can however, place a bound on how rapidly a network must react. Sub-second latency for an initial message after an idle period may be less important than potential energy savings and longer operational lifetime. By contrast, after detection, low-latency operation becomes more important.

6. **Throughput.** This is defined as the amount of successfully transmitted data in a given time. As with latency, the importance of throughput also depends on the application. Factors that affect throughput are efficiency of collision avoidance, channel utilization, latency, control packets overhead and good physical layer design. Network applications that demand longer lifetime usually tend to accept lower throughput.

7. **Fairness.** This is the ability of different nodes to share the channel equally. Unlike traditional networks, in sensor networks, all nodes cooperate for a single common task and at any particular time, one node may have dramatically more data to send than other nodes. Thus, success is measured by the performance of the application as a whole and per-node fairness becomes less important.

The previous section has highlighted the major sources of energy waste in the previous section. All MAC protocols endeavour to minimise energy consumption from these sources due to their direct link with the transceiver modules. MAC protocols can be classified into two groups, namely schedule-based MACs and contention-based MACs. We now proceed to discuss the strategies developed under each classification in more detail.

2.5.1 Schedule-based Medium Access Control

Protocols in this class avoid communication interference by scheduling the nodes onto different sub-channels that are either divided by time (Time Division Multiple Access or TDMA), frequency (Frequency Division Multiple Access or FDMA) or orthogonal codes (Code Division Multiple Access of CDMA) as shown in figure 2.2.

2.5.1.1 Time Division Multiple Access

TDMA divides the channel into N time slots and in each slot, only one node is allowed to transmit. The N number of slots collectively comprise a frame, which repeats cyclically. The base station is responsible for allocating a unique time slot to all nodes along with the task of synchronizing their clocks. Typically, in a single hop scenario, nodes communicate only with the base station (although recent research in literature has included synchronised trees) and there is no direct, peer-to-peer communication between individual nodes. TDMA protocols are very energy efficient because they directly support low-duty-cycle operations on nodes. However, there are some vital disadvantages of TDMA that restricts its usage in wireless sensor networks. It requires the nodes to form clusters (similar to the cells in the cellular communication systems). One node is often selected as the cluster head. In a single-hop network this is the base station. In this hierarchical organization nodes are normally restricted to communicate with the cluster head and peer-to-peer communication is not directly supported.

Heinzelman et al. (2000) propose a Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol that utilises TDMA in wireless sensor networks. LEACH organizes nodes into cluster hierarchies, and applies TDMA within each cluster. The role of cluster head is frequently rotated from node to node within the cluster. Nodes only talk to the cluster head using short range radio. The cluster head in turn talks to the base station over a long-range radio. The drawback of this approach is that inter-cluster communication and interference needs to be handled by other approaches such as FDMA or CDMA.

In contrast to clusters, Capetanakis (1979); Djukic and Valaee (2007) propose a self-configuration TDMA (SC-TDMA) where the sink node constructs tree structures by spanning all the nodes. Each node cooperates with its neighbors to help construct the tree. Once a node joins the tree, its parent assigns it a collision-free time slot, and hereafter, the node only wakes up on the assigned time slot of its children and itself to reduce energy consumption efficiently.

TDMA protocols have limited scalability and adaptivity to the changes in the number of nodes. It is an absolute must for the base station to be informed when this happens so that it can adjust the frame length and slot allocation. Frequent changes in the network may be expensive and slow to take effect. Also, frame length and static slot allocation can limit the available throughput for any given node, and the the maximum number of active nodes in any cluster may be limited. Finally, TDMA protocols heavily depend on distributed, fine-grained time synchronization to align slot boundaries. Without accurate time synchronisation they fail in a grand manner.

2.5.1.2 Frequency Division Multiple Access

FDMA divides the given spectrum into channels by the frequency domain. Each transmission link between two different nodes is allocated one channel. Sohrabi et al. (2000) have proposed a self-organization protocol for wireless sensor networks that assumes multiple radio channels are available, and any interfering links select and use different sub-channels. During the time that is not scheduled for transmission or reception, a node turns off its radio to conserve energy. Each node maintains its own time slot schedules with all its neighbors, which is called a super frame. Time slot assignment is only decided by the two nodes on a link, based on their available time. It is possible that nodes on interfering links will choose the same time slots. Although the super frame looks like a TDMA frame, it does not prevent collisions between interfering nodes, and this task is actually accomplished by FDMA. This protocol supports low-energy operation. However a major disadvantage is the relatively low utilization of available bandwidth. A sub-channel is dedicated to two nodes on a link, but is only used for a small fraction of time, and it cannot be re-used by other neighboring nodes.

2.5.1.3 Code Division Multiple Access

CDMA (Glisic and Vucetic, 1998; Garg et al., 1996) employs spread-spectrum technology and a special coding scheme where each transmitter is assigned a code. It is a form of *spread-spectrum* signaling because the modulated coded signal has a much higher bandwidth than the data being communicated. Each group of nodes is given a shared code. Many codes occupy the same channel, but only nodes associated with a particular code can understand each other. Each node in a CDMA network can use all available frequencies. Adjacent nodes can transmit at the same frequency because they are separated by code channels and not frequency channels. CDMA protocols are very secure as it is virtually impossible for rogue nodes (those who enter the network to disrupt communication and gather data illegally) to decipher the code. However, CDMA can be very complex to implement in the simple architecture of a sensor node. Furthermore, they are not particularly power efficient and hence are more widely used to handle inter-cluster communication between Bluetooth piconets instead.

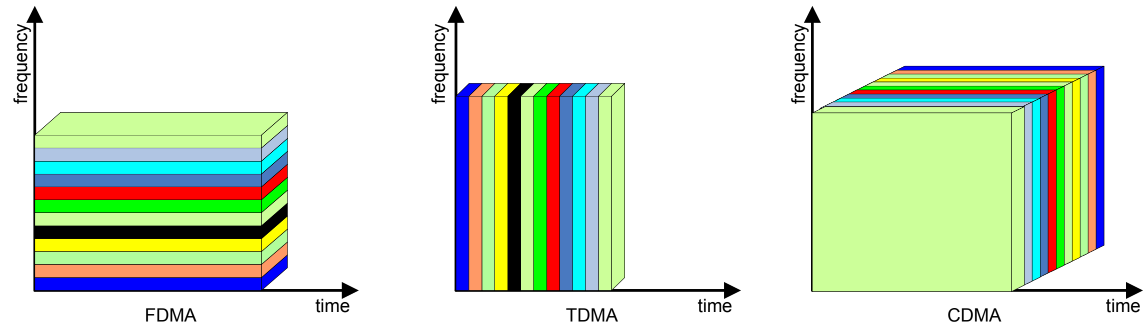


FIGURE 2.2: Schedule based protocols.

2.5.2 Contention-based Medium Access Control

Protocols under this category compete for a shared channel rather than pre-allocating transmissions. As a result, this leads to probabilistic coordination. Networks that implement such protocols are prone to packet collisions. One of the most primitive examples of a contention based MAC protocol is the ALOHA (Abramson, 1973). In pure ALOHA a node simply transmits a packet when it is generated. In slotted ALOHA (Abramson, 1985), time is divided into slots and nodes transmit at the next available slot. Packets that collide are discarded and are retransmitted later. There have been various developments in contention based protocols and some popular protocols are discussed below.

2.5.2.1 Carrier Sense Multiple Access (CSMA)

In part I of their work Kleinrock and Tobagi (1975) explain CSMA as a non-deterministic protocol in which a node verifies the absence of other traffic on the shared medium prior to commencing its own transmission. The term *Carrier Sense* explains that the transceiver module *sniffs* for a carrier wave in the air in order to detect the presence of an encoded signal from neighbouring nodes. If a carrier is sensed, the node realises that the medium is busy and holds back its own transmission until a set period of time (by default) before it begins to sense again. The term *Multiple Access* explains that multiple nodes send and receive on the medium. There are several variants of CSMA and the three most widely used are described below:

1. **Non-persistent CSMA.** If a node detects an idle medium, it decides to transmit immediately. If the medium is busy then the node backs off for a random period of time before it starts to sniff for a carrier again. The random back-off reduces the probability of collisions although if the back-off time is too long it results in wasted idle time and long access delays.

2. **1-persistent CSMA.** If the node understands that the medium is idle then it decides to transmit immediately. If, however, the medium is busy then it continues to *sniff* until the medium becomes available and then transmits immediately. This can turn out to be a very selfish strategy and may result in frequent collisions if two stations want to retransmit.

3. **p-persistent CSMA.** If the medium is idle, the node transmits with a probability of p, and delays for one time unit with probability (1-p). The time unit can is defined as the length of propagation delay. If the medium is busy the node continues to listen until medium becomes idle and then makes a decision based on the probabilities. This strategy is a good trade-off between non-persistent and 1-persistent CSMA. However, a typical problem with this strategy lies in deciding a value of p. Assuming there are N number of nodes that want to transmit, then the expected number of nodes that will attempt to transmit once the medium becomes idle is given by Np. Therefore, the strategy must ensure $Np < 1$ to avoid any collisions.

2.5.2.2 Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA).

The CSMA protocol is very simple and effective but is not a sufficient enough solution for a multi-hop wireless network. The hidden terminal problem that is not addressed in CSMA is discussed in (Tobagi and Kleinrock, 1975). Figure 2.3 illustrates this problem on a 2-hop network with 3 nodes.

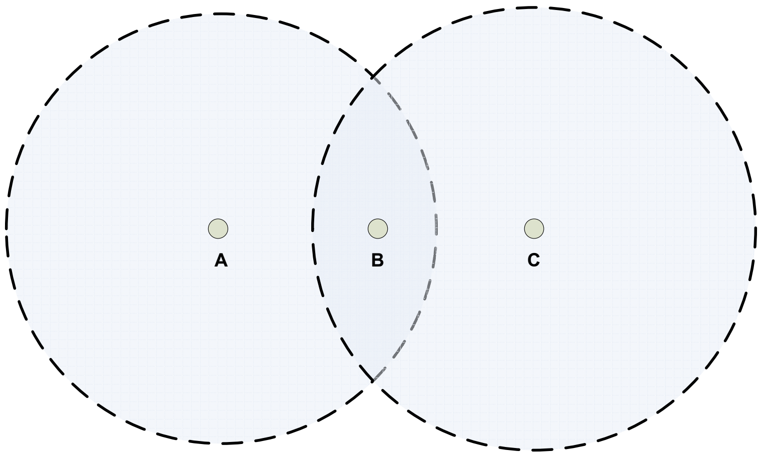


FIGURE 2.3: Hidden terminal problem.

The circles represent the *idealised* [1] transmission range of nodes **A** and **C**. Node **B** falls in the transmission range of **A** and **C** but the latter two do not know of each

[1]In reality, radio irregularity is a common phenomenon which arises from multiple factors, such as variance in RF sending power and different path losses depending on the direction of propagation. For this reason, the transmission range is not always a perfect circle of sphere for that matter. However, for the purpose of illustration, Figure 2.3 diagram has been simplified.

other's existence. Thus, if **A** is transmitting to **B**, node **C** will not be aware of this transmission because its own carrier sense would indicate that the medium is idle. If **C** starts transmitting then **B** will receive corrupted packets as a result of collision.

CSMA/CA was developed to counter this problem. This was adopted by wireless LAN standard IEEE 802.11. Collision avoidance strategy attempts to reserve the network for a single transmitter. The underlying concept involves establishing a brief handshake between the sender and the receiver before the sender commences transmission. The handshake entails the sender sending a short *Request-to-Send* (RTS) packet to the intended receiver. The receiver replies with an equally short *Clear-to-Send* (CTS) packet after which normal data transmission can begin. This handshake informs all the neighbouring nodes around both sender and neighbour to back off. The handshake limits the hidden terminal problem to RTS and CTS packets. However, these control packets are very small in size compared to data packets and thus the cost of collisions is reduced.

2.5.2.3 Multiple Access with Collision Avoidance (MACA

Whilst CSMA/CA contributes to avoiding collisions by ensuring that nodes back off when they overhear a handshake it does not specify how long the nodes should back off for. This issue is addressed in MACA (Karn, 1990) by extending CSMA/CA. In this strategy a duration field is added in both the control packets. This indicates to the neighbouring packets, the amount of data to be transmitted and thus helping them make a decision on how long to back-off for.

In an effort to further improve upon this Bharghavan et al. (1994) proposed MACAW. This strategy extends MACA by adding the acknowledgment (ACK) control packet. The ACK is sent after each data packet and allows the rapid link-layer recovery from transmission errors. The transmission between sender and receiver follows the sequence of RTS-CTS-DATA-ACK and this is depicted in Figure 2.4.

2.5.3 Hybrid-based Medium Access Control

We have so far discussed MAC strategies that are broadly divided into schedule-based and contention-based protocols. The former class of strategies work in a distributed way, have good flexibility, but channel detection and control packet overhead can consume a considerable amount of energy. Schedule-based protocols are energy efficient, but require extremely accurate time synchronisation between nodes and their performance degrades in the face of scalability. Recently, there have been several protocols proposed in literature that combine the strengths of schedule-based and contention-based strategies whilst at the same time off-setting their weaknesses. This section covers some well established protocols.

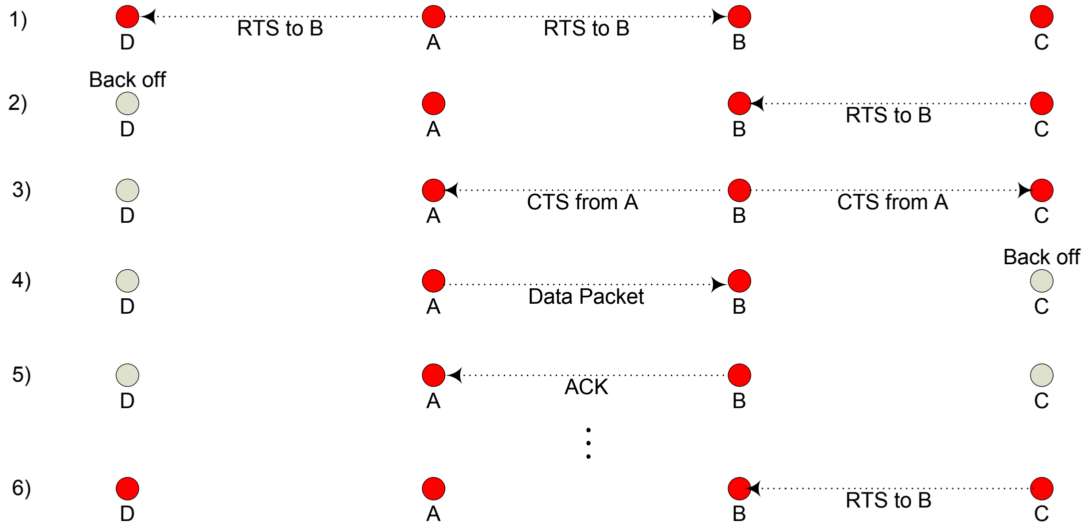


Figure 2.4: MACAW. The active nodes are coloured red. Neighbours of both transmitter (node A) and intended receiver (node B) are informed to back off through the use of control packets. Although node C also wants to transmit, it is unable to do so after failing to get a confirmatory CTS for its own transmission and therefore backs off. Later when A's transmission is over all nodes become active again and start competing for the medium.

2.5.3.1 Sensor-MAC (SMAC)

The basic idea behind the S-MAC protocol (Ye et al., 2002a) involves locally managed synchronisations and periodic sleep-listen schedules based on these synchronizations. Neighboring nodes form virtual clusters and share a common sleep schedule. In case two neighbouring nodes happen to reside in two different virtual clusters, they wake up at the listen periods of both clusters. Schedule exchanges are accomplished by periodic SYNC packet broadcasts to immediate neighbours. The listen period should be carefully chosen. If it is too long, too much energy would be wasted in idle listening. If it is too short, contention probability increases and hence more energy is wasted for the retransmission efforts.

S-MAC also encompasses the concept of *message passing*. Sending a long message as a single packet can incur a high cost of re-transmission in case of corruption of data. A long message in this context is application specific where multi-sensor nodes (nodes with several sensors) in place. A single transmission packet which comprises of data from all the on-board sensors may be several bytes long (eg. 64 bytes for 6 samples (Martinez et al., 2004)) as opposed to shorter packets for nodes with only one or two sensors (eg. 32 bytes for 25 samples (Werner-Allen et al., 2006)).

Also, sending a long packet as several smaller packets can incur a large cost of control overhead as each packet would have an RTS/CTS exchange. Message passing requires only one RTS and one CTS packet to reserve the medium which allows multiple fragments from the same message to be sent in a burst. Each packet is acknowledged separately and re-transmitted if necessary by extending the transmission duration. Each packet also includes the duration for the remaining transmission, allowing nodes that wake up in the middle of transmission to return to sleep. This helps reduce message level latency at the expense of unfairness in medium access.

Periodic sleep may result in high latency, especially for multi-hop routing algorithms, since all intermediate nodes have their own sleep schedules. The latency caused by periodic sleeping is called sleep delay. The adaptive listening technique is proposed to improve the sleep delay and thus the overall latency. In this technique the node that overhears its neighbours transmissions, wakes up for a short time at the end of the transmission. Hence, if the node is the next-hop node, its neighbor could pass data immediately. The end of the transmissions is known by the duration field of the RTS/CTS packets.

In addition to its implementation simplicity, time synchronization overhead may be prevented by sleep schedule announcements. The sleep schedules considerably reduce energy waste caused by idle listening but they are predefined and constant which decreases the efficiency of the protocol under variable traffic load. Broadcast data packets do not use the handshaking mechanism and contributes to a high probability of collisions. Also adaptive listening incurs overhearing or idle listening if the packet is not destined to the listening node.

2.5.3.2 Timeout MAC (T-MAC)

Dam and Langendoen (2003) propose T-MAC as extension to S-MAC in order to enhance its poor efficiency under variable traffic load. In this strategy, the listen period terminates if no activation event occurs for a time threshold TA. The challenge lies in selecting an appropriate value for TA. In a typical scenario, a node could end its listen period early and go to sleep while its neighbours are still communicating. This may not be the right thing to do since the node may be the receiver of a subsequent message. Receiving the start of the RTS or CTS packet from a neighbour is enough to trigger a renewed interval TA. Since a node may not hear, because it is not in range, the RTS that starts a communication with its neighbour, the interval TA must be long enough to receive at least the start of the CTS packet. This provides a lower limit of on the length of the interval TA given by

$$TA > Time_{contention} + RTS + Time_{turnaround} \tag{2.1}$$

where the turn-around time is the short time between the end of the RTS packet and the beginning of the CTS packet. A larger TA increases the energy used. Although T-MAC gives better results under these variable loads, the synchronization of the listen periods within virtual clusters is broken. This is one of the primary reasons for the early sleeping problem.

2.5.3.3 Dynamic Sensor MAC (DSMAC)

Another variant of SMAC is the DSMAC (Lin et al., 2004) which implements a dynamic duty-cycle feature in order to decrease the latency. Initially all nodes start with the same duty cycle. The nodes share their one-hop latency values (the time between the reception of a packet into the queue and its transmission). If a receiver node notices that the average one-hop latency value is high, it decides to shorten its sleep time by half and announces it within the SYNC period. Accordingly, nodes in the same cluster that receive this updated schedule, check their queue for packets destined to that receiver node. If there are packets to transmit, they decide to double their duty cycle provided their battery level is above a specified threshold. This is particularly useful for delay-sensitive applications and is also shown to have better average power consumption per packet. However, other inherent problems of SMAC still remain.

2.5.3.4 WiseMAC

The WiseMAC protocol (Enz et al., 2004) uses non-persistent CSMA in conjunction with synchronised preamble sampling to decrease idle listening. Each transmitting node precedes its data packet with preamble for alerting the receiving node. All nodes sample the medium (listen to the medium for a short duration) periodically with a common time interval, however, their relative schedule offsets are not common. If a node wakes up to find the medium busy after sampling, it continues to listen until it receives a data packet or the medium becomes idle again. To begin with, the size of the preamble is set to be equal to the sampling period. However, this may result in *overemitting*-type energy waste (transmitting a message when the destination is not ready) since there is no way for the transmitter to know if the receiver is ready at the end of the preamble. Moreover, overemitting increases as the length of the preamble and the data packet increase due to lack of any handshaking mechanism with the intended receiver.

To reduce the power consumption incurred by this predetermined fixed-length preamble, WiseMAC offers a method that dynamically determines its length. Nodes learn and refresh their neighbours' sleep schedule during every data exchange as part of the Acknowledgment (ACK) message. Thus, every node keeps a table of the sleep schedules of its neighbours and schedule transmissions so that the destination nodes sampling time corresponds to the middle of their preamble. Random wake-up preamble is used to decrease the possibility of collisions caused by the specific start time of the wake-up preamble. The choice of the wake-up preamble length is also affected by the potential clock drift between the source and the destination. A lower bound for the preamble length is calculated as the minimum of destinations sampling period, Tw, and the potential clock drift with the destination, which is a multiple of the time since the last ACK packet arrived. Considering this lower bound, a preamble length Tp is chosen randomly.

WiseMAC shifts the cost of communication from receivers to transmitters. The dynamic preamble length adjustment results in better performance under variable traffic conditions and the clock drifts are handled in the protocol definition itself mitigating the external time synchronization requirement. However, the different sleep and wake-up times for each node is problematic for broadcast-type communication. Broadcast packets are buffered for neighbours in sleep mode and delivered many times as each neighbour wakes up resulting in expensive redundant transmission and increased latency. Furthermore, the hidden terminal problem is not addressed in this model as the preamble might cause collisions on a node in the process of receiving data from another node.

2.5.3.5 Traffic Adaptive Medium Access (TRAMA)

TRAMA (Rajendran et al., 2003) increases the utilisation of classical TDMA in an energy-efficient manner. It uses a distributed election algorithm to select one transmitter within each two-hop neighborhood. This eliminates the hidden-terminal problem and ensures that all nodes in a one-hop neighborhood of the transmitter receive data without any collision. Time is divided into random-access and scheduled-access (transmission) periods. The random-access period involves contention based access to the channel in order to establish two-hop topology information.

Nodes are able calculate the number of slots for which they have the highest priority among two-hop neighbours within the period $[t, t+T_{interval}]$ where $T_{interval}$ is the transmission duration. They announce the slots that they will use as well as the intended receivers for these slots through a schedule packet. Additionally, the nodes also announce the slots for which they have the highest priority but will not use. Since the receivers of these schedule packets have the exact list and identities of the one-hop neighbours, they find out the intended receiver. When the vacant slots are announced, potential senders are evaluated for reuse of those slots. Priority of a node on a slot is calculated with a hash function of nodes and slots identities.

TRAMA nodes undergo higher delay as compared to CSMA-based protocols due to a higher percentage of sleep time. This also leads to less probability of collisions. The intended receivers are indicated by a bitmap resulting in less communication required during multi-cast and broadcast modes in comparison to other protocols. However, even though the transmission slots are set to be seven times longer than the random-access period, nodes are still either in receive or transmit states during the latter for exchange of each other's schedules. This implies that the duty cycle is at considerably high value of 12.5% without even taking into account the transmissions and receptions of actual data.

2.5.3.6 EMACS

The scheduling principle in the EMACS protocol (Nieberg et al., 2003) developed by the EYES project is very simple in that every node gets to control one time slot in every frame. After the frame length, which consists of several time slots, the node again has the same slot reserved for it. The time slot is further divided in three sections: *Communication Request* (CR), *Traffic Control* (TC) and the *data* section. In the CR section nodes intending to transmit make requests to the node that is controlling the current time slot. Nodes that have a request, will pick a random start time in the short CR section to make their request. This is similar to an RTS message and communication in this section is not guaranteed collision-free. Nodes that do not have a request for the current slot owner, keep their transceiver turned off during this slot. The time slot controller responds with a TC message in the next sub-period of the slot granting permission to the sender to transmit. This is comparable to the CTS message. The time slot controller also indicates in its TC message what communication will take place in the data section. If a node is not addressed in the TC section nor its request was approved, then the node will resume in standby state during the entire data section. After the TC section the actual data transfer takes place. When a time slot is not controlled by any node, all nodes sleep during that slot.

2.5.3.7 Sift

In event driven sensor networks, when an event is sensed, the first R of N potential reports are extremely crucial and have to be relayed immediately with low latency. The Sift algorithm (Jamieson et al., 2003) was proposed to address this issue. The algorithm uses a nonuniform distribution function of picking a slot within a slotted contention window. Assuming that the number of competing nodes is small, if no node begins to transmit in the first slot of the window, then each node increases its transmission probability exponentially for the next slot. Since Sift is a contention slot assignment algorithm, it is proposed to coexist with other MAC protocols like S-MAC. It achieves very low latency for many traffic sources at the cost of energy consumption. The high energy consumption are caused due to increased idle listening (listening to all slots before sending) and increased overhearing (nodes must listen until the end of an ongoing transmission in order to contend for the next transmission.

2.5.4 General Discussion of MAC

-	**Time Synch.**	**Type**	Advantages	Issues
S-MAC	No	CSMA	Adaptive Listening, Message passing	Not adaptive to variable load
T-MAC	No	CSMA	Adaptive Listening, Adaptive to variable load	Early sleep, Broken synchronisation between virtual clusters
DSMAC	No	CSMA	Adaptive Listening, Decreased Latency	Not adaptive to variable load
WiseMAC	No	np-CSMA	Adaptive to variable load, Energy consumption reduced in receivers	Problematic for broadcast type, Hidden terminal problem
TRAMA	Yes	TDMA/ CSMA	Increases utilisation of TDMA slots	Higher delay
EMACS	Yes	TDMA/ CSMA	Receiver controls time slot	Increased latency
SIFT	No	CSMA/CA	Very low latency	Increased collisions, Idle listening

Table 2.1: Comparison of MAC protocols

Table 2.1 summarises the various MAC layer protocols discussed in this section, with each having its advantages and disadvantages. There is no protocol accepted as a standard. This is because the choice of MAC protocol, in general, application-dependent. Furthermore, lack of standardization at lower layers (physical layer) and the (physical) sensor hardware in various applications makes it harder to devise a standard MAC protocol. TDMA protocols have the natural advantage of collision-free medium access. However, they introduce the problem of clock drifts and decreased throughput at low traffic loads due to idle slots. In addition, TDMA systems require synchronization of the nodes and it is difficult for nodes to adapt to topology changes caused by insertion of new nodes, exhaustion of battery capacities, broken links because of interference, sleep schedules of relay nodes, scheduling caused by clustering algorithms. Slot assignment can be a challenging task within a decentralized environment since all nodes must agree on the slot assignments. On the other hand, CSMA methods have a lower delay and promising throughput potential at lower traffic loads, which generally happens to be the case in wireless sensor networks. However, additional collision avoidance or collision detection methods should be employed to handle the collision possibilities. FDMA can offer a collision-free medium. However, it brings an additional circuitry requirement to dynamically communicate with different radio channels. This increases the cost of the sensor nodes, which is contrary to the objective of the sensor network systems. CDMA also offers collision-free medium, but its high computational requirement is a major obstacle for less energy consumption objective of the sensor networks.

This section has extensively covered various MAC protocols that have been developed for sensor networks. However, there is no protocol accepted as a standard. One of the reasons for this is that the choice of MAC protocol, in general, is application dependent. This means that there is no standard MAC for sensor networks. Another reason is the lack of standardization at the lower physical layer where hardware platforms vary immensely. Common wireless networking experience also suggests that the data link-level performance alone may provide misleading conclusions about the system performance. Hence, the system can be more efficient by contribution of higher layers in decision making. For instance, the routing path (networking layer) could be chosen depending on the collision information from the medium access layer. Therefore, the study proceeds further to review work done in the networking layer in the next section.

2.6 Networking Layer: Routing

We have mentioned that it may be necessary for nodes in a sensor network to transmit their data in short hops where intermediate nodes help relay data forward on to the base station. Thus, it is important to recognise that these intermediate nodes not only consume energy in transmitting their own data but also in receiving and relaying data on behalf of other nodes. In such a setting, those nodes that are closer to the base station are depleted of their energy much faster than those further away. This can result in data routes to change over a period of time as increasing number of intermediate nodes die leaving nodes that are further away in a state of isolation (too far to transmit to base station). Thus, it is important to devise energy efficient routes such that networks do not reach such a state too quickly and data gathered from the sensor nodes is maximised. The networking layer aims to address this issue. The design of a good networking layer takes into consideration the following 3 important principles:

1. **Nature of sensor network.** Most sensor networks can be divided into three categories based on their canonical methods of storage (Shenker et al., 2002). These are

    - *External storage.* Upon detection of events, nodes send their sensed data to an external storage where it can be further processed and entails no external queries from the user.
    - *Local storage.* Here, the sensed data is stored locally in each node and has to be queried.

- *Data-centric storage*. Here, all events are classified under a particular type and stored in a different node depending on their type. Querying is done by directing the query to that particular node.

The cost of each method in terms of routing differs from situation to situation. However, data-centric storage is particularly helpful if there are large numbers of static nodes allowing users to query classes of events and provides helpful global context for evaluating data.

2. **Useful Data aggregation**. We have already mentioned in Section 2.2 that processing consumes far less power than transmission. Therefore, nodes in a sensor network should take advantage of this characteristic to convert the raw data into something meaningful before transmitting. More so, intermediate nodes in a multi-hop environment should be able to aggregate data collected from different nodes before forwarding them further. This is essential to overcome implosion and overlap of data.

3. **Location Awareness.** In a typical sensor network, nodes may want to have an some knowledge of the neighbouring nodes in terms of the approximate radio transmission power required to facilitate communication. Based on this knowledge, the networking layer can then find energy efficient routes for a packet from a source to a destination. Some common approaches include *maximum available energy* route, *minimum energy consumption* route and *minimum hop* route.

Based on the above three principles, the network layer formulates the backbone of sensor network connecting all the nodes together. Several distributed routing protocols have been studied in literature and these are discussed in this section.

2.6.1 Introduction to Agents

As aforementioned, a typical sensor network is one in which the constituent nodes are spread over a large area and are subject to various restrictions such as fixed bandwidth allocation, limited battery lifetime, low processing speed and finite memory storage. Such networks are being deployed in a wide variety of application areas. These include environmental monitoring (Britton and Sacks, 2004), traffic control management (Cheung et al., 2005), real-time health monitoring (Otto et al., 2006) and surveillance purposes (He et al., 2004). With the cost of nodes being extremely low, they are expendable. However, the cost of deployment is usually much higher and therefore nodes are deployed in large numbers ranging from a few hundred to millions. Now due to the growing size of sensor networks, there is a strong need for decentralisation and autonomous components that act and interact in flexible ways in order to achieve the overall objectives of the system in uncertain and dynamic environments. Given this, *agent-based computing* (Wooldridge, 1998) has been advocated as the natural model for sensor networks. For this reason it is necessary to explore the basis of agents.

In the software domain, the term *agent* has many definitions. (Smith et al., 1994) define agents to be "computer programs that simulates a human relationship by doing something that another person could do for you". (Russell and Norvig, 2003) provide a more simple view of an agent by defining it to be anything that perceives its environments, and acts accordingly upon that environment through effectors. Although there is no general consensus on the definition of an agent, a widely accepted description of an agent provided by Wooldridge and Jennings (1995) defines an agent to be a software-based computer system that has the following 4 properties:

1. **Autonomy.** Ability to function without intervention and have control over its own actions and internal state.
2. **Social-ability.** Ability to interact with other agents and humans using a defined communication language.
3. **Reactivity.** Ability to perceive its environment and act in response to the changes in that environment.
4. **Pro-activity.** Goal-based behaviour that enables it to be proactive and not just react to external stimuli. The agent's goal should drive its actions.

In a distributed system it is common to have more than one agent and such a system is known as a *multi-agent* system. In such a system, several distinct components, each of which is an independent problem solving agent, come together to form some coherent whole (Luck and d'Inverno, 2004). There is no goal that drives the entire system with many agents; instead each agent entity has its own goal. This implies that agents need to cooperate and coordinate with one another to form one coherent system that serves the end user. The coordination and cooperation is a necessity to avoid duplication of effort, unwittingly hindering other agents in achieving goals and to exploit other agents' capabilities.

Today like most sensor networks, multi-agent systems are also open i.e. they allow agents to enter and leave when they want, creating ever changing topologies. In these systems agents typically represent different competing organisations and it is normal to assume that due to the different ownership of these agents, they could be self-interested. Nodes in a small sized sensor network are usually owned by a single application. However, a large open network may contain tens of thousands of heterogeneous nodes having been developed by various stakeholders and different times. The presence of such nodes at any given time in such an open system cannot always be determined due to the continual change of the environment. In spite of this, the nodes are expected to be able to operate efficiently without much interference from users or system designers. A multi-agent system offers the ability to model these nodes as autonomous agents, capable of negotiating and communicating on their own without user intervention.

Agent-based systems are inherently modular and this is particularly useful in complex systems where each smaller problem can be solved by a single agent. The overall solution can be produced by the interaction of these autonomous problem solving agents. This model of problem solving also provides the system designers with an abstraction. It enables them to view the complex software as a virtual community of interacting problem solvers. Therefore, the four properties that an agent must possess i.e. proactive, reactive, social-ability and autonomy; strongly apply to nodes of a decentralised sensor network.

Against this background, a number of agent-based routing protocols have been investigated to enhance the performance of wireless sensor networks. Although these protocols have not been discussed explicitly in terms of agent terminology, the nodes employing these protocols exhibit agent characteristics. These protocols are generally classified under one of the following three categories. Namely, *data-centric*, *hierarchical* or *location-based*.

Location-based protocols utilise the position information to relay the data to the desired regions rather than the whole network. Such protocols require each node to maintain a constantly updated map of several nodes if not all at any time. This introduces more overhead in terms of exchange of control packets. Hence these protocols are mainly limited to mobile ad-hoc networks (Yu et al., 2001; Li and Halpern, 2001). For this reason, we can discount a review of location-based protocols. The remainder of this section focuses on data-centric and hierarchical routing protocols.

2.6.2 Data-centric Routing Protocols

Traditionally, nodes were queried based on their pre-assigned unique address. However, this is not feasible for networks with a large number of randomly placed nodes, because it is likely to result in redundant data which is highly energy inefficient. In order to address this shortcoming, data-centric protocols, in which sensors are identified based on the data they sense rather than their address, were developed. These protocols are thus query-based and depend on attribute-based naming of desired data (to specify particular properties of data), in order to eliminate redundant transmissions.

2.6.2.1 Flooding

One of the oldest and most abstract routing algorithms in sensor networks is flooding. As the name suggests, the protocol works in such a way that each agent node receiving a data packet repeats or retransmits the packet by broadcasting it. In this way, the data

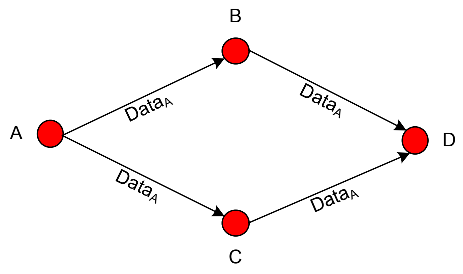


FIGURE 2.5: Data Implosion. Agent A starts by flooding its data to all of its neighbors. D gets two same copies of data eventually, which is not necessary.

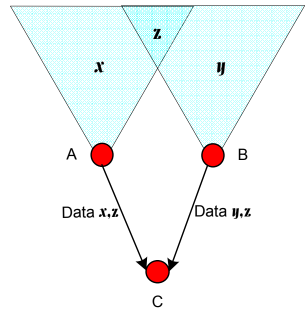


FIGURE 2.6: Data overlap. Two sensors cover an overlapping geographic region and C gets the same copy of data form these sensors.

floods its way through the network and reaches the required destination in the fastest time possible. This reactive techniques is extremely simple and does not require costly topology maintenance through complex route discovery algorithms. It does however, have many disadvantages. Flooding tends to continue in other areas of the network even after the packet reaches its intended destination thus wasting power unnecessarily. In addition, it has three major deficiencies which are listed below.

1. **Implosion.** This occurs when several copies of the same message are received by a single node. For example, if agent **A** shares **N** neighbours with agent **B**, the latter may get **N** copies of the same message originated by **A**. This results in a lot of unnecessary transmission. See Figure 2.5.
2. **Overlap.** This phenomenon occurs when two nodes located near each other share the same observation region and sense the same stimuli at the same time. Although this might be useful in some applications for confirming events, it results in neighbouring agents receiving duplicated messages.
3. **Resource Blindness.** Agents employing this protocol are not energy resource aware because they do not take into consideration the amount of energy available to them.

2.6.2.2 Gossiping

Gossiping (Hedetniemi et al., 1988) is a derivation of flooding and is aimed to address some of its problems mentioned above. It works in a way that each agent receiving a message decides to randomly select one of its neighbours. It then forwards the packet to this neighbour. This process repeats until the receiving agent turns out to be the intended destination of the packet itself. In this way, the problem of implosion is eliminated almost completely. However, one its biggest drawbacks is that it takes a long time for the message to propagate through the network till it reaches its intended destination. Furthermore, the problem of data overlap still exists.

2.6.2.3 Sensor Protocols for Information via Negotiation (SPIN)

In an attempt to refine the flooding and gossiping methodology, the SPIN protocol (Heinzelman et al., 1999) considers data negotiation between agents to disseminate information in the network and eliminate redundant data. It does so by adopting a publish-subscribe approach where agent nodes operate efficiently and conserve energy by only sending meta-data (i.e. data describing the data), instead of sending all of the actual data.

Agents running SPIN perform meta-data negotiations via a data advertisement mechanism before any data is transmitted. Each agent upon receiving new data, advertises it to its neighbours. Those who do not have the data but are interested in it retrieve the data by sending a request message. This solves the classic problems of flooding such as redundant information passing, overlapping of sensing areas and resource blindness thus, achieving a lot of energy efficiency. There are three messages defined in SPIN to exchange data between nodes. The **ADV** message allows an agent to advertise a particular meta-data. **REQ** message is used to request the specific data. Finally, **DATA** message is used to carry the actual data itself.

One of the advantages of SPIN is that topological changes are localised since each node needs to know only its single-hop neighbours. Thus, this model is useful for those agents interested in the data advertised and is an effective protocol to minimise energy spent in consumption until the actual data is transmitted. However, its data advertisement mechanism fails to guarantee the delivery of data. For instance, if the agents that are interested in the data are far away from the source and the intermediate agents between the source and destination are not interested in that data, such data will not be delivered to the destination at all. Therefore, SPIN is not suitable for applications, such as intrusion detection, which require reliable delivery of data packets over regular intervals.

2.6.2.4 Directed Diffusion

Directed diffusion (Intanagonwiwat et al., 2000) is a conceptually converse approach to SPIN. The idea aims at diffusing data through sensor nodes by using a *naming* scheme. It suggests the use of attribute-value pairs for the data and queries the agents in an on demand basis by using those pairs. In order to create a query, an *interest* is defined using a list of attribute-value pairs such as name of objects, interval, duration, geographical area and so on. The interest is then broadcasted by the base station through its neighbors. Each agent receiving the interest caches uses it to compare the received data with the values in the interests. The interest entry also contains several *gradient* fields. A gradient is simply a reply link to a neighbor from which the interest was received. It is characterized by the data rate, duration and expiration time derived from the received interests fields. Hence, by utilizing interest and gradients, paths are established between the base station and the data sources. Several paths can be established so that one of them is selected by *reinforcement*. The base station resends the original interest message through the selected path with a smaller interval hence reinforcing the source node on that path to send data more frequently. Figure 2.7 illustrates the directed diffusion protocol.

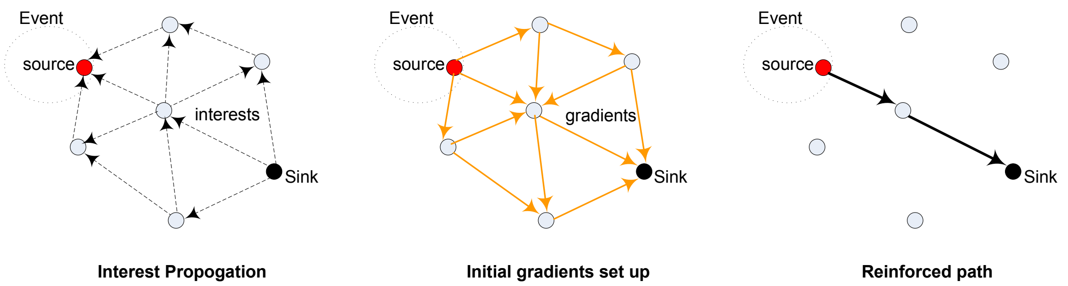


FIGURE 2.7: Directed Diffusion

Path repairs are also possible in Directed Diffusion. When a path between a source and the sink fails, a new or alternative path should be identified. For this, Directed Diffusion basically re-initiates reinforcement by searching among other paths, which are sending data in lower rates. There is an extra overhead of keeping the alternative paths alive by using low data rate, and this requires extra energy. However, more energy can be saved when a path fails and a new path needs to be chosen. Directed Diffusion differs from SPIN in terms of its demand data querying mechanism. In SPIN, agents advertise the availability of data allowing interested agents to query that data. In Directed Diffusion, however, the sink queries the agent nodes if a specific data is available by flooding some tasks. This has many advantages. Since it is data centric, all communication is neighbor-to-neighbor with no need for an ID addressing mechanism. Each agent can do aggregation and caching, in addition to sensing and this is advantageous in terms of energy efficiency and delay. In addition, it is highly energy efficient since it is on demand and there is no need for maintaining global network topology. However, because of its query-driven data model, it can not be employed in those sensor networks that require continuous data delivery to the sink. Therefore, it is not a feasible routing protocol for the applications such as environmental monitoring.

2.6.2.5 Minimum Cost Forwarding Algorithm

In most sensor networks, the direction of routing (towards a fixed base station) is always known. This characteristic is exploited by the Minimum Cost Forwarding Algorithm

(MCFA) (Ye et al., 2001). This protocol argues that an agent node need not have a unique ID nor maintain a routing table. Instead, it should maintain the least transmission cost estimate from itself to the base-station. A message to be forwarded by the agent node is broadcast to its neighbours. Each agent receiving this message, verifies whether it exists on the least cost path between the source and the base-station. If so, it re-broadcasts the message to its neighbors and the process repeats until the message reaches the base station. This protocol requires each agent to know the least cost path estimate from itself to the base station. To determine this estimate the base station begins by broadcasting a message with the cost set to zero. Initially, every agent in the network has its least cost estimate value set to infinity by default. Upon receiving the broadcast message that originated at the base station, each agent updates its current estimate to the sum of the estimate in the message and the link on which it is received the message if it is less than the current estimate. The message is then re-broadcasted to neighbours only if the current estimate is updated.

2.6.2.6 Energy Aware Routing

Shah and Rabaey (2002) argue that using the minimum energy path all the time depletes the energy of agents on that path. Therefore, in order to increase the lifetime of the network as a whole, one of the multiple sub-optimal paths available should be used. These paths are chosen by means of a probability function, which depends on the energy consumption of each path. The primary concern of this approach is the metric of network survivability and it assumes that each agent is addressable through a class-based addressing scheme that includes the location and types of agents. The protocol involves 3 main phases:

1. **Setup phase**. During this phase localised flooding occurs in order to establish communication routes and routing tables are created. In doing so, the total energy cost is calculated for each node. For example, if the request is sent from node N_i to node N_j , N_j then calculates the cost of the path as follows:

   $$C_{N_j,N_i} = Cost(N_i) + Metric(N_j, N_i)$$

   The energy metric captures the cost of transmission and reception along with the residual energy of the nodes. Paths with a very high cost are ignored. Nodes are selected according to a measure of their closeness to the destination. A probability is assigned to each of neighbour in routing table (also known as forwarding table (FT)) corresponding to the formed paths. The value of each probability is inversely proportional to the cost and is given by:

   $$P_{N_j,N_i} = \frac{1/C_{N_j,N_i}}{\sum_{k \in FT_j} 1/C_{N_j,N_k}}$$

N_j then calculates the average cost for reaching the destination using the neighbors in the forwarding fable (FT_j) using the formula:

$$Cost(N_j) = \sum_{i \in FT_j} P_{N_j,N_i} C_{N_j,N_i}$$

2. **Data Communication Phase.** Each node forwards the packet by randomly choosing a node from its forwarding table using the probabilities.

3. **Route maintenance phase.** In order to keep all the paths alive, localised flooding is performed on a regular basis.

This approach is similar to Directed Diffusion in the way potential paths from data sources to the sink are discovered. Whilst in Directed Diffusion, data is sent through multiple paths and only one is reinforced to send at higher rates, this approach selects a single path randomly from the multiple alternatives. The advantage here is that it provides a small improvement energy saving and increase in network lifetime. The disadvantage of this approach is that it requires gathering information about the location of agents for setting up the addressing mechanism for nodes which is not feasible in many wireless networks.

2.6.2.7 Gradient-based Routing

Gradient-Based Routing (GBR) (Schurgers and Srivastava, 2001) is a variant of directed diffusion where the agents memorize the number of hops at the same time when interests are diffused through the network. This eliminates the need for the *gradients set-up* phase and allows the agents to calculate their *height*, which is the minimum number of hops for their packet to reach the base station. The difference between the height of two agents is considered the gradient on the link between them. Packets are forwarded on a link with the largest gradient. In addition, GBR uses auxiliary techniques such as data aggregation and traffic spreading in order to uniformly divide the traffic over the network. For example, if multiple paths pass through relaying agent, it may combine data according to a certain function and then use one of the following three dissemination techniques:

1. **Stochastic Scheme**. Here, the agent picks one gradient at random when there are two or more next hop links that have the same gradient.

2. **Energy-based scheme**. Here, if the energy of an agent decreases below a certain threshold, it decreases its height to discourage others from sending data to it.

3. **Stream-based scheme**. Here, new streams are not routed through agents that are already part of the path of other streams.

2.6.2.8 Constrained Anisotropic Diffusion Routing (CADR) and Information Driven Sensor Querying (IDSQ)

A more general form of Directed Diffusion is discussed by Chu et al. (2002). The idea behind their work is to query agents and route data in a network such that information gain is maximised whilst at the same time bandwidth and latency is minimised. This is achieved by activating only those agents that are close to a particular event and then dynamically adjusting data routes. It differs from Directed Diffusion by taking into consideration the information gain in addition to the communication cost. They do so by proposing two techniques namely constrained anisotropic diffusion routing (CADR) and information-driven sensor querying (IDSQ).

In CADR, each agent evaluates an information/cost objective and routes data based on the local information/cost gradient and end-user requirements. The information utility measure is modeled using standard estimation theory. IDSQ, on the other hand, is based on a mechanism in which the querying agent can determine which agent can provide the most useful information whilst balancing the energy cost. While IDSQ provides a way of selecting the optimal order of sensing agents for maximum incremental information gain, it does not specifically define how the query and the information are routed between the agents and the base station. Therefore, it is seen as a complementary optimization procedure to CADR.

2.6.2.9 COUGAR

A radically different approach to traditional data-centric protocols is proposed in the COUGAR protocol (Yao and Gehrke, 2002). This protocol views the network as a huge distributed database system and uses declarative queries. The COUGAR approach abstracts query processing from the network layer functions such as selection of relevant agents and utilises in-network data aggregation to save energy. The abstraction is supported by introducing a new query layer between the network and application layers. In this architecture for the sensor database system the agents select a leader to perform aggregation and transmit data to the base station. The base station generates a query plan by specifying the necessary information about data flow and in-network computation for the incoming query and sends it to the relevant agents. The query plan also describes how to select a leader for the query.

Thus, the architecture plan of the protocol provides in-network computation ability for all the agents ensuring energy efficiency, especially when there is a huge number of agents interacting with the leader. However, this network-layer independent solution for querying the agents, has certain drawbacks. Introducing an additional query layer on each agent brings extra overhead in terms of energy consumption and storage. Furthermore, in-network data computation from several agents requires synchronisation amongst agents. In other words, a relaying agent does not receive all data at the same time from incoming sources and has to wait for every packet, before forwarding it to the leader agent. Finally, the leader agents need to be dynamically maintained in order to prevent them failure prone.

2.6.2.10 ACQUIRE

Another data-centric approach that also views the sensor network as a distributed database is *ACtive QUery forwarding* (ACQUIRE) (Sadagopan et al., 2005). It is particularly suited for complex queries which consist of several sub queries. The query is initiated by the base station and each agent receiving the query, tries to respond partially by using its pre-cached information and forwards it to another agent. If the pre-cached information is not up-to-date, the agents gather information from their neighbors within a look-ahead of d hops. Once the query is resolved completely, it is sent back either through the reverse or the shortest path to the base station. ACQUIRE's primary motivation is to deal with one-shot, complex queries for data where a response can be provided by many agents.

Since, the data-centric approaches such as Directed Diffusion use flooding-based query mechanism for continuous and aggregate queries; it would not make sense to use the same mechanism for one-shot complex queries due to energy considerations. The mechanism behind ACQUIRE provides efficient querying by adjusting the value of parameter d. If the value of d is equal to to the size of the network, then the protocol behaves similar to flooding. On the other hand, the query has to travel more hops if d is too small.

2.6.3 Hierarchical Routing Protocols

Most sensor networks are single-tier in nature (i.e. there is only one base station that acts as a gateway to many agents around it. With increase in the number of agents in the network, there is a high chance that the gateway may be overloaded with traffic and cause undesired latency in communication and inadequate tracking of events. Furthermore, a single-tier architecture is not scalable for a larger set of agents covering a wider area of interest since the agents are typically not capable of long-haul communication. In other words, the low power transceivers installed on the agents may only allow it to communicate over short distances (a few metres), say a gateway node, which in turn may want to forward the data to a base station a several kilometres away using a much more powerful radio transceiver.

Against this background, networking clustering has been pursued in some routing approaches. In such approaches, agents are involved in multi-hop communication within a particular cluster and perform data aggregation and fusion to decrease the number of transmitted messages to the base station. Cluster formation is typically based on the energy reserves of the agents and their proximity to a cluster head. This forms the basis of hierarchical routing protocols. We now explore some well-known hierarchical protocols in this sub-section.

2.6.3.1 Low Energy Adaptive Clustering Hierarchy (LEACH)

One of the first hierarchical routing approach for sensor networks was proposed in the LEACH protocol (Heinzelman et al., 2000). The aim of this cluster based protocol is to minimise energy dissipation in the agents and is designed for remote environment monitoring applications. The protocol binds the concept of distributed cluster formation in the network, local processing in the cluster heads to reduce global communication and randomised rotation of the cluster head. The whole operation of this protocol is divided over two phases:

1. **Setup Phase.** This phase begins with the random selection of a cluster head based on probabilistic model. Once the cluster heads are selected, they advertise their new status to the rest of the group. The remaining agents in the network then determine their affiliation with a cluster based on the signal strength of the advertisement packets they received from the cluster heads. Once a cluster is formed the cluster heads assign communication time slots to the agents.
2. **Steady Phase.** This phase lasts longer than the setup phase and agents transmit their data to the cluster head which in turn aggregate all the data they collect before forwarding it to the base station. This phase lasts a set period of time after which the system re-enters the *setup phase* and new cluster heads are selected.

This algorithm exhibits energy-efficiency because only the cluster heads communicate with the base station. The rest of the agents save energy by communicating with the cluster heads at a much smaller distance. More so, cluster heads aggregate data collected from the cluster agents minimising communication. Also the rotation of cluster heads ensures that energy consumption is shared out amongst all agents extending network lifetime. LEACH is completely distributed and requires no global knowledge of network. However, it uses single-hop routing where each agent can only transmit directly to the cluster-head and the base station. Therefore, it is not applicable to networks deployed in large regions. Furthermore, the idea of dynamic clustering brings extra overhead such as head changes and advertisements which may diminish the gain in energy consumption.

2.6.3.2 Power Efficient Gathering in Sensor Information Systems (PEGASIS)

In order to enhance the functionalities of LEACH, a near optimal chain-based protocol called PEGASIS (Lindsey and Raghavendra, 2002) was proposed. This protocol is based on the idea that in order to extend network lifetime, agents only need to communicate with their closest neighbours. These neighbours can then take turns in communicating with the base station. In this manner, the power draining is spread uniformly over all agents. The local coordination between only the agents close together significantly reduces bandwidth used in communication. It avoids cluster formation and uses only one agent in a chain to transmit to the base station instead of using multiple agents. The chain consists of those nodes that are closest to each other and form a path to the base-station. Signal strength of the radio is used to identify the closest neighbour. This signal strength is then adjusted so that only one node can be heard. The aggregated form of the data is sent to the base-station by any one node in the chain. The nodes, however, take turns in sending to the base-station.

PEGASIS is able to double the network lifetime in comparison to LEACH by eliminating the overhead caused due to dynamic cluster formation and by decreasing the number of transmissions and reception by using data aggregation. However, it still requires dynamic topology adjustment because an agent requires knowledge about the energy status of its neighbors to know where to route its data. This can introduce significant overhead especially for highly utilized networks. Moreover, all agents are assumed to be capable of directly communicating with the base station which is not feasible for many networks. Finally, the single leader in the PEGASIS protocol can be a major cause for a bottleneck.

2.6.3.3 Threshold Sensitive Energy Efficient Sensor Network (TEEN)

Another hierarchical protocol designed to be responsive to sudden changes in the sensed attributes such as temperature has been proposed in TEEN (Manjeshwar and Agrawal, 2001). Responsiveness is particularly important for time-critical applications where the deployed agents operate in a reactive mode. TEEN pursues a hierarchical approach along with the use of a data-centric mechanism. After the clusters are formed, the cluster heads broadcast a hard threshold and a soft threshold to their cluster agents for the sensed attributes. The hard threshold signifies the *minimum possible* value of an attribute that should trigger an agent to switch on its transmitter and transmit to the cluster head. This allows the agents to transmit only when the sensed attribute is in the range of interest, thus reducing the number of transmissions. The soft threshold signifies the *change in value* of the attribute (once the value goes above the hard threshold) that should trigger the agent to switch on its transmitter and transmit to the cluster head.

As a consequence, this further reduces the number of transmissions if there is little or no change in the value of sensed attribute. However, TEEN is not good for applications where periodic reports are needed since the end-user may not get any data from the network if the thresholds do not reach the agents.

To avoid this from happening, work on TEEN was extended to propose a new protocol called Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network (APTEEN) (Manjeshwar and Agrawal, 2002). APTEEN is a hybrid protocol that changes the periodicity or threshold values used in the TEEN protocol according to the user needs and the type of the application. In APTEEN, the cluster-heads broadcasts a parameter called *Count Time* in addition to the threshold values. This parameter indicates the maximum time period between two successive reports sent by an agent. If an agent does not send data for a time period equal to the count time, it is forced to sense and retransmit the data. Thus it combines both proactive and reactive policies. The main drawbacks of the two approaches are the overhead and complexity associated with forming clusters at multiple levels, the method of implementing threshold-based functions, and how to deal with attribute-based naming of queries.

2.6.3.4 Sensor Aggregates Routing

In work done by Fang et al. (2003), a set of algorithms for constructing and maintaining sensor aggregates were proposed. The objective is to collectively monitor activity in target tracking applications. An aggregate comprises of those agents in a network that satisfy a grouping predicate for a collaborative processing task. The parameters of the predicate depend on the task and its resource requirements. The formation of appropriate aggregates are discussed in terms of allocating resources to sensing and communication tasks. Agents in the sensor field are divided into clusters according to their sensed signal strength, so that there is only one peak per cluster. One peak may represent one target, multiple targets, or no target in case the peak is generated by noise sources. This is followed by electing cluster leaders. To elect a leader, information exchange between neighbouring agents is necessary. If an agent finds that it is higher than all its one-hop neighbors on the signal field landscape, it declares itself a leader. This leader-based tracking algorithm assumes the unique leader knows the geographical region of the collaboration.

A lightweight protocol called Distributed Aggregate Management (DAM) is used for forming agent aggregates to monitor a target. The protocol comprises of a decision predicate P for each agent to decide if it should participate in an aggregate and a message exchange scheme M about how the grouping predicate should be applied to the agents. An agent is able to determine if it belongs to an aggregate by simply applying the predicate to its data as well as information gathered from agents. Aggregates are formed when the process eventually converges. An Energy-Based Activity Monitoring

(EBAM) algorithm is used to estimate the energy level of each agent by computing the signal impact area and combining a weighted form of the detected target energy at each impacted agent. The underlying assumption here is that each target agent has equal or constant energy level. A third algorithm, Expectation-Maximization Like Activity Monitoring (EMLAM), is used to estimate the target positions and signal energy using received signals. This is combined with the resulting estimates to predict how signals from the targets may be mixed at each agent. This process is iterated, until the estimate is sufficiently good. These three protocols combine to form a scalable system and works well in tracking multiple targets when the targets are not interfering.

2.6.3.5 Hierarchical Power-aware Routing

Another protocol called hierarchical power-aware routing protocol (Li et al., 2001), divides the network into groups of agents. Each group in geographic proximity are clustered together as a zone and each zone is treated as an entity. To perform routing, each zone is allowed to decide how it will route a message hierarchically across the other zones such that the battery lives of the agents in the system are maximized. Messages are routed along the path which has the maximum over all the minimum of the remaining power (max-min path). The drawback is that using agents with high residual power may be expensive as compared to the path with the minimal power consumption. Thus the authors propose an approximation algorithm, called the $max - min\ zP_{min}$, which is based on the trade off between minimizing the total power consumption and maximising the minimal residual power of the network. Hence, the algorithm tries to enhance a max-min path by limiting its power consumption by doing the following:

1. It finds the path with the least power consumption P_{min} by using the Dijkstra's algorithm.
2. It then finds a path that maximizes the minimal residual power in the network.
3. It optimises both the solution criteria by relaxing the minimal power consumption for the message to be equal to zP_{min} with parameter $z \geq 1$ to restrict the power consumption for sending one message to zP_{min}.

The algorithm consumes at most zP_{min} while maximizing the minimal residual power fraction. This is followed up by zone-base routing that relies $max - min\ zP_{min}$. Zone-base routing is a hierarchical approach where the area covered by the network is divided into a small number of zones. To send a message across the entire area, a global path from zone to zone is found. The agents in a zone autonomously direct local routing and participate in estimating the zone power level. Each message is routed across the zones using information about the zone power estimates. A global controller for message routing is assigned the role of managing the zones. This may be an agent with the highest power. If the network can be divided into a relatively small number of zones, the scale for the global routing algorithm is reduced. The global information required to send each message across is summarized by the power level estimate of each zone. A zone graph is used to represent connected neighbouring zone vertices if the current zone can go to the next neighboring zone in that direction. Each zone vertex has a power level of 1. Each zone direction vertex is labelled by its estimated power level computed using Djikstra's algorithm.

We have extensively reviewed various routing protocols that have been presented in literature. They all have the common objective of trying to extend the lifetime of the sensor network, while not compromising data delivery. Routing techniques can be classified into three categories: data-centric, hierarchical and location-based. For the purpose of this research we have focused this review on the first two categories because the protocols falling in the third category are not energy efficient for a non-mobile sensor network. We shall now proceed to discuss the next layer of abstraction in the protocol stack (i.e. the application layer).

2.7 Application Layer

Although, this thesis has mentioned the various application areas for sensor networks, the research recognises that there are several potential application layer protocols for sensor networks that are yet to be realised. In general, the application layer can be divided into 3 types of protocols. These are:

1. **Node Management.** As the name suggests, these protocols are aimed at making the hardware and software of the lower layers of the protocol stack transparent to the management applications of the entire sensor network. For example, these protocols allow remote access of sensor networks for application users to directly interact with the sensor nodes to perform administrative tasks such as
    - Introducing the rules related to data aggregation, attribute-based naming, and clustering to the sensor nodes.
    - Exchanging data related to the location finding algorithms
    - Time synchronization of the sensor nodes
    - Moving sensor nodes
    - Turning sensor nodes on and off
    - Querying the sensor network configuration and the status of nodes, and re-configuring the sensor network
    - Authentication, key distribution, and security in data communications

2. **Task Assignment and Data Advertisement.** Protocols falling under this category are mainly utilised for interest dissemination of interests where users can send their interest to a node, a subset of the nodes, or the entire network. This interest may be about a certain attribute of the phenomenon or a triggering event. Another approach is the advertisement of available data in which the sensor nodes advertise the available data to the users, and the users query the data in which they are interested. An application layer protocol that provides the user software with efficient interfaces for interest dissemination is useful for lower-layer operations such as routing.

3. **Querying and Dissemination of Data.** These protocols provides the applications with interfaces to issue queries, respond to queries and collect incoming replies from nodes. Although queries to particular nodes can be issued, in general these queries are addressed to nodes based on attribute or location. For example, "the locations of the nodes that have detected the movement of Mercedes cars in the last 2 days" is an attribute-based query. Similarly, "Number of Mercedes cars spotted in region A" is an example of location-based naming.

Having had a brief overview of the application layer. We now proceed to discuss the task management plane.

2.8 Task Management Plane

The task management plane is responsible for helping the sensor nodes coordinate the task of sensing. In doing so it inherently becomes responsible for managing the node's power usage. For example, a node low on power may inform its neighbours that it might not participate in routing data and therefore reserve its power for sensing. The plane is also responsible for detecting and registering the movement of neighbouring nodes so that it can consistently maintain a transmission route to the base station. This is also very important because knowledge of neighbours can help the node balance its power and task usage. For example, it may not be necessary for all nodes in a particular region to perform sensing, unless otherwise specified, when it is sufficient for only one to carry out the task. As a result, some nodes may perform more sensing than others.

2.8.1 Adaptive Sampling

In a sensor network, the nodes themselves might need to adjust their sensing rates. Typically, decisions such as how often to sense are fixed in advance. In a less dynamic environment (one where the variables to be measured do not change frequently) it would be feasible for each node to observe these variables at regular intervals of time. On the other hand, in a dynamic environment (one where the variables to measured change frequently) the rate of making observations will need to be adequate enough so that any significant events are not missed. Consider the following scenario:

A museum building is installed with a sensor network comprising of sensor nodes deployed in each room. The nodes are programmed to take temperature readings of each room every 30 minutes to ensure that the paintings are all maintained at a room temperature of 22 degrees. Now suppose the temperature in one of the rooms starts to rise at a rate of 3 degree celsius every minute. This means that by the time the next reading is taken the room could be well above 100 degrees indicating possibly an event of fire and too late to rescue any paintings in that room!

In the above scenario, if temperature observations were taken in an adaptive fashion, readings can be used as they become available to adjust future observation times. For example, if an upward trend is noticed in the temperature, then more frequent readings could be made in a quicker space of time enabling to understand the rate at which temperature is rising and cautioning the system to take remedial action against the fire. This is known as adaptive sampling where where the accruing data (i.e., the observations) are used to adjust the sensing rate.

Unfortunately, adaptive sampling procedures are complicated to design and to analyze, and in sensor networks are more difficult to implement. Most sensor networks using formal experimental designs use fixed sampling procedures which decide in advance how many samples they are going to collect. Although these fixed procedures are easier to design and analyze, because they don't have as many alternatives, they are less efficient and sometimes significantly so. However, to this end, there are a few sensing protocols that have been proposed in literature that enable nodes to observe the environment in an efficient manner. These are discussed in this section.

2.8.1.1 Inter-node Adaptive Sensing

Inter-node modeling, where correlations among the readings of same type sensors on different but spatially close-by nodes are proposed in works done by Willett et al. (2004), Ye et al. (2003) and Rahimi et al. (2004). These works discuss an adaptive sampling approach for energy conservation in sensor networks that have a hierarchical structure such as LEACH (Heinzelman et al., 2000). In each approach, small subsets of sensor nodes communicate their observations to a fusion centre that in turn provides an initial estimate of the environment being sensed. Based on this coarse estimate, the centre then determines which regions of the field may contain boundaries or sharply varying behaviour and activates additional sensor nodes in those regions. These additional nodes in turn provide finer resolution estimates and the refined estimates are communicated to the fusion centre. The key idea is that the initial estimate detects correlations in the environment, indicating that most of the sensors may not need to be activated by the fusion centre. Thus, adaptive sampling can save energy compared to dense, non-adaptive sampling.

ASAP (Gedik et al., 2007) is a more recent adaptive sampling approach to periodic data sampling and builds on Backcasting and PEAS where only a subset of the nodes are made active, while preserving the network connectivity. ASAP uses a similar logic, but in a different context and for a different purpose: only a subset of the nodes are used to actively sense, while the quality of the collected data is kept high using locally constructed probabilistic models to predict the values of the non-sampler nodes. It makes use of a sensing-driven cluster construction algorithm to create clusters within the network such that nodes with close sensor readings are assigned to the same clusters. Correlation-based sampler selection and model derivation are used by elected cluster heads in order to determine the sampler nodes within the clusters and to calculate the parameters of the probabilistic models that capture the spatial and temporal correlations among the sensor readings. Finally, adaptive data collection and model-based prediction are used to minimize the number of messages used to extract data from the network.

However, these approaches are utilised in the context of field estimation and does not discuss adaptive sampling from the individual sensor's "rate of sampling" point of view. In a large distributed sensor network large numbers of nodes deliver continuous data to the base station. The rate at which each data is sampled at each sensor affects the computational load at the central server.

2.8.1.2 Query Based Adaptive Sampling

Previously, Section 2.6.2 discussed declarative querying as one of the key programming paradigms for data gathering in sensor networks providing an unusual opportunity for database researchers to apply their expertise in the field.

Han et al. (2004) propose a way to minimize the sensor node energy consumption in answering a set of user supplied queries with specified error thresholds. The queries are answered using uncertainty intervals which are cached at the server and updated using an optimized schedule of server-initiated and sensor-initiated updates.

Snapshot Queries (Kotidis, 2005) is another query based approach where each sensor node is either represented by one of its neighbors or it is a representative node. This division is similar to the analogy of sampler and non-sampler nodes discussed in Section 2.8.1.1. The representative nodes predict the values of their dependent neighbors by employing a binary linear regression model. Query evaluation can cut down the energy consumption dramatically for aggregate queries, since a single value will be produced as an aggregate from the value of the representative node and the predicted values of the dependent neighbors. However this local prediction does not support savings for applications that require collection of readings from all nodes.

Declarative querying is powerful in the sense that it allows programmers to task an entire network of sensor nodes rather than requiring to program individual nodes. However applying this mindset to sensor networks results in two problems:

1. **Misrepresentations of data.** Data retrieved by the base station may not reflect the true nature of the environment. This could be attributed to several factors that may include non-uniform placement of the nodes, faulty sensors and high packet loss rates. Thus a straightforward interpretation of the nodes' readings as a "database" may not be a very reliable representation of the real world.

2. **Inefficient approximate queries.** Sensor readings are only a representation of the true state of the world at discrete instants and locations. Therefore, it is fair to say that these readings are approximate. Whilst some approaches to query processing tend to acquire as much data as possible even though most of that data provides little benefit in approximate answer quality.

This problem is tackled by the BBQ model (Deshpande et al., 2005) where the sensor network architecture is incorporated with statistical models of real-world processes to provide robust interpretations of sensor readings by accounting for biases in spatial sampling, identifying faulty nodes and extrapolating the values of node readings that have gone missing or are no longer operational. This provides a framework for optimising the acquisition of sensor readings and nodes are only used to acquire data only when the model itself is not sufficiently rich to answer the query with acceptable confidence.

Such a mechanism, however, is sufficient only when there is prior knowledge about the real-world processes and fails if the nodes are deployed in an alien territory. This highlights a strong need for prediction mechanisms used for forecasting future observations such that the rate of sampling may be adjusted and the least number of observation samples would provide a good approximation of the actual model of the environment.

2.8.1.3 Kalman-Filter based Estimation Technique

The Kalman-Filter (KF) - based estimation technique (Jain and Chang, 2004) is one of the very few adaptive sampling techniques where each node adapts to the streaming-data characteristics. In this approach, the nodes use the KF estimation error to adaptively adjust their sampling rates within a given range, autonomously. When the desired sampling rate violates the range, a new sampling rate is requested from the base station. The base station allocates new sampling rates under the constraint of available resources such that the KF estimation error over the entire set of the active streaming nodes is minimised.

2.8.1.4 Load Shedding

Traditional methods such as load shedding (Tatbul et al., 2003) and adaptive precision setting (Olston et al., 2001) are used in some sensor networks to collect data at peak sampling rate and then determine whether the collected data should be dropped to conserve during communication. However, the task of sensing itself consumes energy (although not as much as communication) and excessive sampling can incur high cost in data collection and processing. In addition, these methods do not use prediction/estimation models and are activated only when the load on the system increases beyond what it can handle. Thus there is a strong need for adaptive sampling modules that are executed during the lifetime of a (data) stream.

Whilst there are several more variants of the different adaptive sensing techniques discussed above, none of them place an emphasis on the importance of the sensed data in relation to the cost of communicating it. Statistical models in minimising the task of sensing alone is not good enough if in the end the data gathered does not add a high enough value to the research for the amount of energy that was compensated in both sensing and transmitting it. This is an important aspect that has been overlooked in research till date and this thesis will attempt to address this issue further in Chapter 5.

The management plane is as important as the layers in the stack because it ensures that the nodes can work together in a power efficient way, route data and share resources between them.

2.9 Summary

In summary, this chapters has reviewed the different layers of abstraction for protocols in a sensor network. In detail, it has focused on the review of protocols implemented in the data-link layer (Section 2.5), the networking layer (Section 2.6) and the task management plane (Section 2.8). It has also highlighted the main areas of research that need to be addressed in each layer and reviewed the strengths and weaknesses of the different protocols. In light of this, we shall proceed to present a short summary of the general requirements of any model/protocol developed for a typical sensor network.

General Requirement 1 (Openness and Scalability) – The model should be scalable and its performance should not be affected by the number of nodes added to it. Furthermore, overall functionality of the network should not be affected greatly if nodes fail.

General Requirement 2 (Decentralisation) – The model should be robust, and continue functioning even if there are problems with parts of the network on which it has to operate.

General Requirement 3 (Communication Robustness) – The model should be able to overcome unreliable communication links between nodes and work with low bandwidth.

General Requirement 4 (Intelligent Routing) – The model should manage the task of processing and communication in an intelligent manner. Nodes should communicate in a manner that expends the least amount of energy.

General Requirement 5 (Intelligent Sensing) – The model should coordinate activities of the physically distributed sensor nodes to collect data in an efficient manner.

The following chapters present the work done specifically for this thesis. More specifically, Chapter 3 provides a real life implementation of a sensor network that has been currently deployed as part of on going research. Here, the strengths and weaknesses of this application are detailed and the list of refined general requirements are highlighted. Chapter 4 presents a data-link layer model for this application. Furthermore, Chapter 5 presents a networking layer model that also incorporates the task of sensing.

Chapter 3

GlacsWeb: A sensor network to monitor sub-glacial environment

Chapter 2 provided a general overview of the relevant research that has taken place in the various aspects of sensor networks. Each of these research studies has evolved around a particular sensor network with its own unique specifications and characteristics. Emphasis on these unique individual specifications is dependent on the sensor network's application and end user. Furthermore, some of these studies are based on theoretical sensor networks whilst others have resulted in sensor network designs catered only for specific environments in which they are deployed.

Although this implies that the same design for a sensor network is not appropriate for all sensor network applications, it does not render all research useless. Certain segments of research studies can still be reused and moulded to address the aims of a new sensor network. However, a refined statement of the problem specific to the new sensor network application has to be formulated. Furthermore, existing research has to be used as a set of building blocks to extend the work in order to meet the demands of the new sensor network.

To this end, this chapter presents a real life sensor network called GlacsWeb which is deployed in a glacier in Norway. It highlights real-world experiences from a sensor network that was developed for operation in the hostile conditions underneath a glacier. Later chapters (Chapters 4 and 5) will discuss the relevant work carried out to make GlacsWeb a more efficient sensor network. In more detail, this chapter describes GlacsWeb, comprising of many sensor nodes placed inside the en-glacial ice and on the glacier's bed (located at the bottom) to take measurements including temperature, pressure, stress and movement. The data gathered is important in understanding the sub-glacial dynamics of glaciers as well as global warming.

3.1 GlacsWeb Motivation

An important challenge today is to understand climate change and its effect on sea level rise due to global warming. Glaciers are a key element in this because they contribute by releasing a lot of fresh water into the sea severely disturbing the thermohaline circulation of the sea and affecting marine life therein. According to glaciologists (Blakenship et al., 1986; Alley et al., 1986), 90% of the melting of West Antarctica's ice is controlled not only by snow fall and surface melting but also by processes under the ice. The behaviour of these processes, however, are very poorly understood. Both sea level rise and climate change directly affect society so any monitoring and understanding of these processes will help provide solutions to these far reaching effects. Furthermore, these processes determine the overall movement of the glacier and it is vital to understand them in order to predict future changes. Therefore, the main motivation behind GlacsWeb is to use technological advances to understand what happens beneath glaciers and how they are affected by climate.

The key to understanding these sub-glacial processes requires measuring the relative motion of the sub-glacial bed to the surface and study of the motion of small rocks in the bed (See Appendix A). Therefore in order to mimic rocks, sensor nodes are required to be placed in and under the glacier which contain appropriate sensors but must communicate their data to the surface via radio communications.

These requirements led to the following technical objectives:

- miniaturisation
- low power design
- ad-hoc networking
- autonomous and adaptive behaviour

The GlacsWeb team were particularly interested in how the hostile environment would influence the design of the sensor network and hardware. Having identified the key motivation factors and the overall objectives of GlacsWeb, we move onto the next section which presents the basic architecture of the network.

3.2 System Architecture

The purpose of the GlacsWeb sensor network is to collect data from sensor nodes (Probes) within the ice and the till (sub-glacial sediment) without the use of wires which could disturb the environment. The system is also designed to collect data about the weather and position of the base station from the surface of the glacier. The final aspect of the network is to combine all the data in a database on the Sensor Network Server (SNS) together with large scale data from maps and satellites. Figure 3.1 shows a simple overview of the system. The system is composed of Probes embedded within the ice and the sediment till at the bottom of the glacier, a Base Station positioned on the surface of the ice, a Reference Station located 2.5 km from the glacier so that it has access to mains electricity and the Internet) and a sensor network server based in Southampton (United Kingdom).

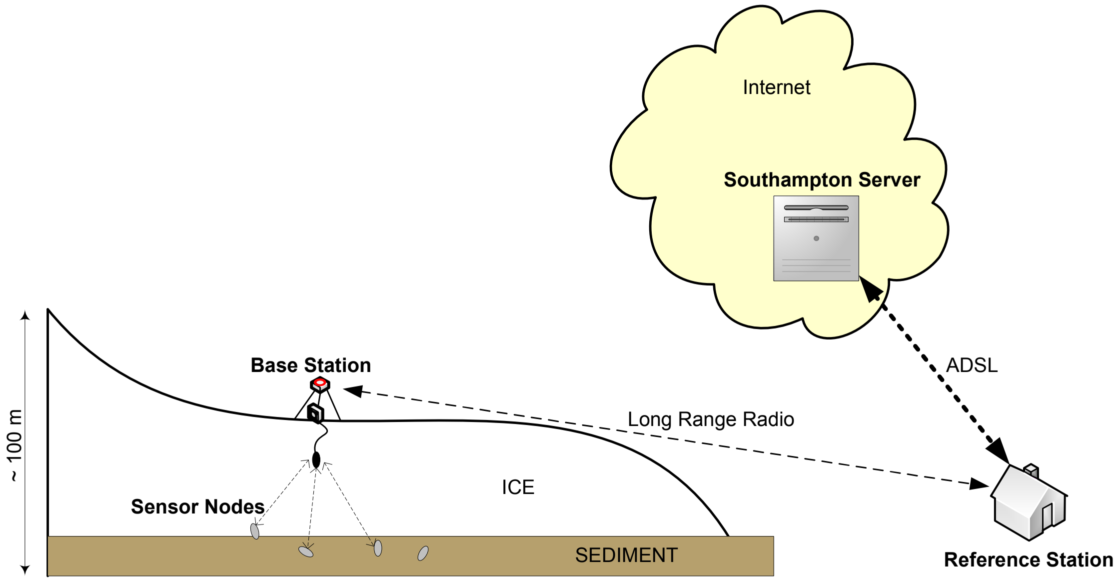


FIGURE 3.1: Overview of the GlacsWeb network. The system is composed of sensor nodes embedded in the ice and the sub-glacial sediment to monitor data and transmit it to the base station positioned on the surface of the ice. The base station in turn accumulates additional information about the weather and sends it to a Reference Station (approximately 2.5km away) that has access to mains electricity and a phone connection. The data is finally uploaded to a Southampton-based server through the Internet to be accessed by glaciologists for analysis.

Before deployment, the probes are programmed to wake up every 4 hours and record various measurements that include, temperature, strain (caused due to stress from the ice), the pressure (to see if the probes are immersed in water or not), orientation or tilt (in the 3 dimensions), resistivity and light (to determine if probes are sitting on sediment till, water or ice). This method provides 6 sets of readings from each sensor of each probe daily.

The base station is programmed to communicate to the probes once a day during a set time window. It is powered up from its standby state for everyday, during which, it collects data from the probes and reads the weather station measurements. It also takes 10 minutes once every week to record its location using the differential GPS installed on it. This 10 minute window is also often used to remotely login from the Southampton server for maintenance purposes. All data collected by the base station is then sent to the reference station PC via a long range radio modem. Figure 3.2 shows the sequence of events occurring during the communication window describing the communication process between probes, base, reference station and the Southampton Server.

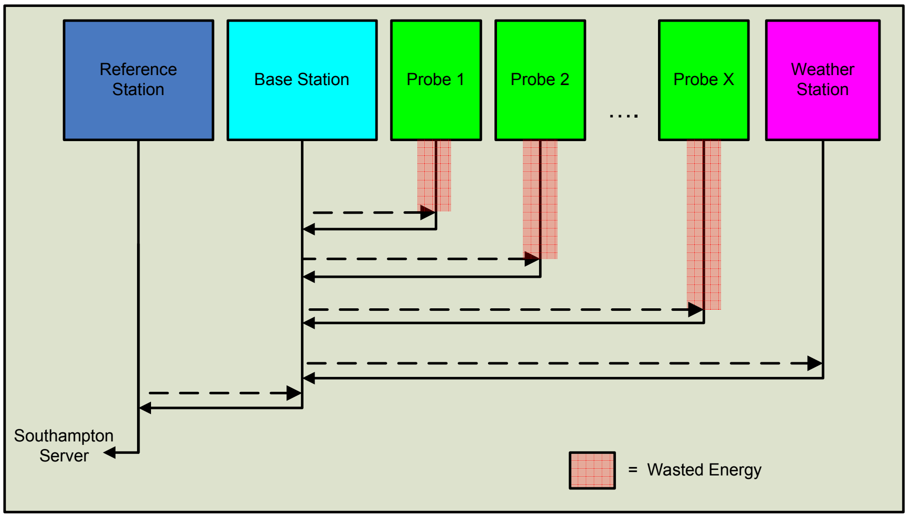


FIGURE 3.2: Sequence of Events during Communication

The reference station is configured to transfer all the data to the sensor network server via an ISDN dial-up or broadband every evening. This data is stored in a database to be used by glaciologists to interactively plot graphs in order to interpret sub-glacial events.

Having now described the overall skeleton of the GlacsWeb system it is necessary to understand in more detail its finer aspects. Furthermore, it is also essential to understand what were the key reasons behind the design of these finer aspects. The next section provides information about the factors affecting the overall design of GlacsWeb. In doing so it also provides insight of the system in detail.

3.3 Design of GlacsWeb and factors affecting it

In a sub-glacial environment, nodes can be subject to constant immense strain and pressure from the moving ice. Therefore, a robust sensor design, integrated with high levels of fault tolerance and network reliability was developed. The design of the system was influenced by a comprehensive list of factors including scalability, power consumption, production costs and hardware constraints. These factors served as essential guidelines for the design structure of the network and the chosen protocol for communication. The rest of the sub-sections discuss the impact of each factor on the overall design.

3.3.1 Production Cost

Most sensor networks consist of a large number of sensor nodes. More often than not if the cost of the network is more expensive than the cost of deployment, the sensor network is not cost-justified. Taking into consideration, however, the hostile environment of the glacier and the hazards that the nodes were expected to face without failing over a long duration of time, it was a pragmatic decision to invest substantially in the development of the nodes. The final cost of each probe came to an estimated £177. Owing to the specialised nature of the sensor network, the cost of deployment was significantly higher as it involved travel to Norway with scientific equipment, accommodation for several nights and other expenses (large amounts of diesel and physical effort to drill holes in the ice). Unlike traditional sensor networks where nodes are easily placed in the observation environment, deploying the probes in the glacier is a laborious process. It could take up to a week to deploy only 20 nodes at the bottom of the glacier. Furthermore, an additional week was required to choose an appropriate location for the base station and its setup. For these reasons, cost was a significant aspect in limiting the number of probes being deployed to approximately 20 every year.

3.3.2 Power Consumption

Probe

Figure 3.3 shows the design architecture of a GlacsWeb probe. Each probe is powered with six 3.6V Lithium Thionyl Chloride cells providing 21.6Wh of energy. The cells were chosen due to their high energy density and good low temperature characteristics. The probes are designed to consume only 1μA in their sleep mode, where only the real time clock and voltage regulators are powered. Table 3.1 shows the power consumption of the probe in its various functioning states.

A probe is installed with a PIC18F4320 micro controller which is programmed to activate the sensors every 4 hours to take measurements and then go back in to sleep mode. The sensing activity lasts for about 50ms. It is also programmed to communicate with the base station once a day when it is powers up its transceiver for a maximum of 3 minutes. During this window it attempts to send its data readings directly to the base station. An approximate calculation of a probe's daily power consumption is 5.8mWh.

Base Station

The Base station is powered with lead-acid gel batteries with a total capacity of 96Ah (1152Wh). These batteries feed power to a Strong ARM-based embedded computer (BitsyX), Global Position System (GPS), Global System for Mobile Communication (GSM) and long range communication modules and a weather station. The BitsyX consumes 120mW in sleep mode and 1.45W when operating. The base station is powered

Power mode	Current consumption (across 5V)
Sleep	$\approx 1\ \mu A$
10 Sensors switched on for about 50ms	60mA
Idle Transceiver	10mA
Transmitting at 10 mW (Tx-10)	35mA
Transmitting at 100 mW (Tx-100)	90mA
Receiving (Rx)	10mA

TABLE 3.1: Probe consumption in different states

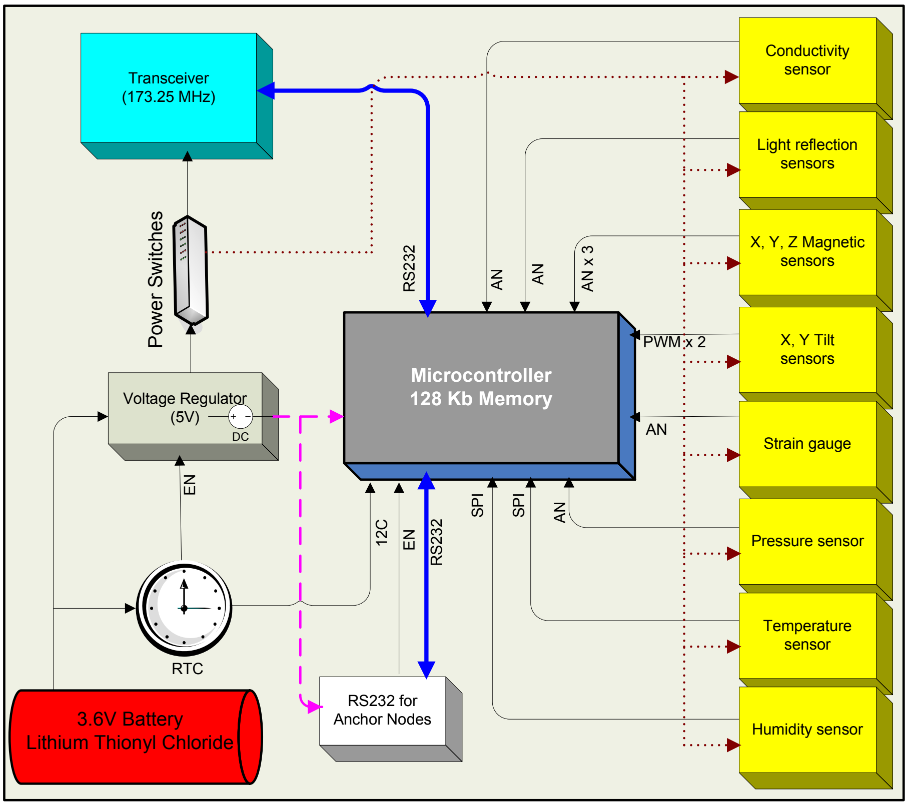


FIGURE 3.3: GlacsWeb Probe Design

up for a maximum of 15 minutes a day during which it communicates with the probes, takes measurements, monitors the weather station and sends its data to the reference station via the long range communication modem. The estimated power consumption during this job is approximately 1W (4Wh per day). This combined with a consumption of 170mW (120mW BitsyX + 50mW Weather Station average) in sleep mode results in a total estimated daily consumption of 5Wh. The batteries are connected in parallel with two solar panels (15W in total) to produce 15Wh per day during summer. In addition a wind generator is also in place to effectively make use of the wind power during winter months when there is no sunlight. This means that the batteries should never run out of energy as they are constantly recharged. This does not, however, imply that the base station has an indefinite lifetime and is discussed further in Section 3.3.5. The schematic of the base station is illustrated in Figure 3.4.

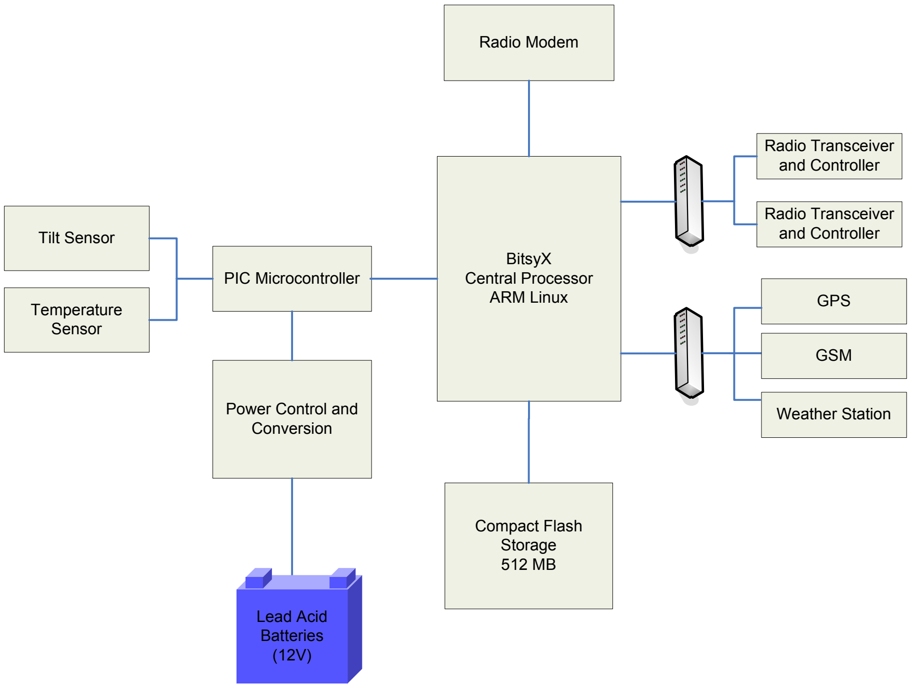


FIGURE 3.4: Base station: schematic

3.3.3 Transmission Media

The communication module for the probes like most other sensor networks is also based on radio frequency (RF) circuit design. There have, however, been a few developments to the design over the years in order to accommodate better transmission through ice. The presence of liquid water is a major problem when trying to use radio waves underneath a glaciers. This problem is especially exaggerated during the warmer months of the year when large portions of the glacial structure melts forming sub-glacial rivers and puddles. These forms of the en-glacial water bodies scatter and absorb the radio signals making it difficult to receive coherent transmissions (Gades et al., 2000).

The GlacsWeb team has concluded that this is part of the reason why there was such a high failure rate of communication with the first batch of probes deployed in the summers of 2003 and 2004 (Martinez et al., 2004). Therefore, the communication frequency has been reduced (halved) in every deployment phase since then. Thus, by reducing the frequency and hence increasing the wavelength, the radio waves skips majority of the water bodies that would usually impede the signal of smaller wave. Table 3.2 indicates the communication frequency employed by the nodes in each year of deployment. This progressive reduction in the frequency has helped lower signal losses significantly. However, as a direct result of this adjustment, the size of the radio antenna on the probe has also increased each year. Figure 3.5 illustrates how the antenna size has been modified with each deployment.

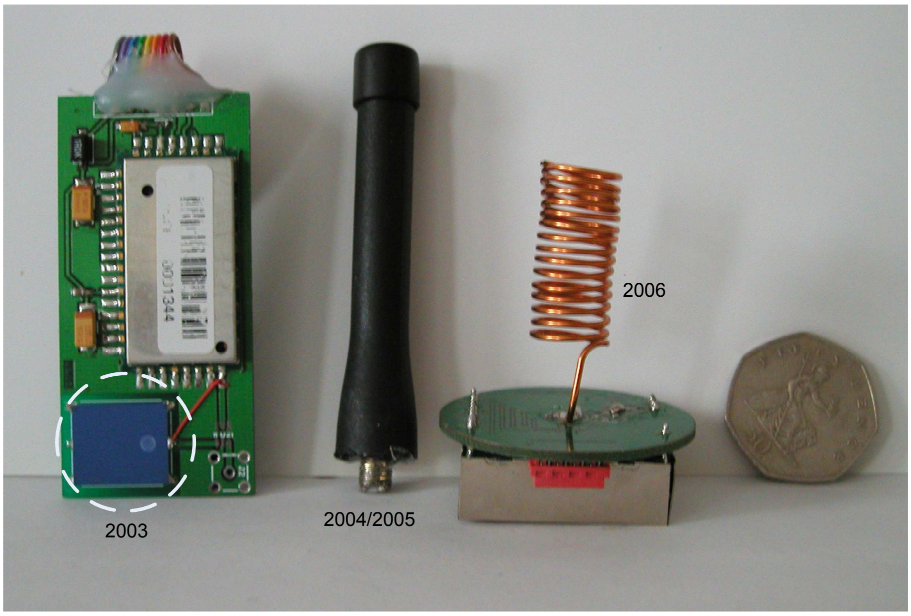


Figure 3.5: Varying antenna sizes with each deployment

In addition, the radiated RF power has also been increased significantly in successive deployments. This has been done using transceiver modules that incorporate a programmable RF power amplifier that boosts the transmission power to over 100mW to improve the signal penetration through ice. To further improve communication, base station transceivers (connected via serial cables) are suspended 30-40m under the surface of the ice so that they are closer to the ice-embedded probes.

Year of Deployment	Frequency Employed	RF Power
2003	868MHz	10mW
2004	868MHz	100mW
2005	433MHz	100mW
2006	173MHz	100mW

Table 3.2: Communication frequency and RF power of the nodes in each deployment

3.3.4 Scalability

During its first three phases the GlacsWeb network was infrastructure based, i.e. all nodes were only one hop away from the base station. The polling mechanism initially used for communication between the probes and the base station, has a natural advantage over other contention based protocols due to reduced duty cycles, no overhead of control packets and no collisions. This fixed scheduling mechanism, exposes the problem arising with the deployment of additional new nodes and the openness of the system. This means that the polling window at the base station has to be adjusted as and when nodes enter and leave the system. Nevertheless, due to the relatively small size of the GlacsWeb network (at most 20 nodes at any time), this is not seen as a major problem.

The base station runs on a Linux operating system, which executes a sequence of shell scripts and a custom *cron*-like scheduler to complete its daily jobs. This provides the designer to assume full control of the system and reconfigure the shell scripts to update the communication schedule in order to incorporate new probes without hampering the overall network's operation.

3.3.5 Fault tolerance

Most sensor networks are designed to manage multiple sensor node failures without upsetting the functionality of the entire network. In a system like GlacsWeb where only a limited number of nodes are deployed, it is crucial that all aspects of the system are robust. The glacier's environment is nevertheless very hostile to allow smooth operation of the system including communication. Therefore some vital measures were taken in order to sustain the network functionalities, even at the cost of time delay, during breakdown of certain aspects of the system. These are discussed below.

Probe Failure.

The probe's firmware was designed to have an alterable 3k word segment called *user space*. It holds programs that are autonomously executed whenever the probe awakens. Programs can be loaded or removed from the user space providing flexibility to modify the probes' functionalities from the Internet. A watchdog timer also placed on the firmware ensures that any rogue programs loaded into the user space are terminated if they exceed some preset timeout. A simple function that records this failure in the probe's cache ensures that the same rogue program is not automatically executed next time.

Base Station Failure.

In an event where the base loses communication with the reference station over the long range modem, the GSM modem is activated. This allows data to be sent directly to the UK server via short messaging service (SMS). The probes house a 64Kb Flash ROM which is organized as a ring buffer. The six sets of measurements recorded by the probe over one day use 96 bytes. These are time stamped and stored in the Flash ROM. This allows the probe to store up to 682 days worth of data in the event of a short range link failure where the base fails to communicate with the probe.

Communication Failure.

An custom communication packet has been developed to specially cater for the system due to the limited resources provided by the PIC micro-controller embedded in the probes. The packet size varies between 5 and 20 bytes. The gap between each transmitted byte was set to a maximum of 3ms to ensure spurious data did not inhibit valid communication. The packet incorporates a checksum byte. If a communication error is detected, the receiver (i.e. the base) retries establishing communication. Such events are logged and can be analysed to investigate communication behaviour. The limit on the number of retries during failures was set to 3 as a compromise between reliability and power consumption. This decision proved to be generally correct because in practice few retries are ever seen. In the months after deployment, when the probes are still near the bore-hole and lots of water, communication is less reliable. However, there is rarely a need for more than three retries on a packet. Figure 3.6 shows the retry behaviour of the system, averaged across all nodes. The quality measure Q has been computed as follows:

$$Q = 100 * \frac{N_{successful}}{N_{total}} \tag{3.1}$$

where $N_{successful}$ is the number of error-free packets and N_{total} is the total number of packets transmitted.

It can be seen that there are more retries in the settling-in period after August (month 8) than in the winter months when there is less liquid water around. The graph does not show anything from January and February. This is because the base station suffered a break-down during this period as a result of a severe winter. The absence of much sun during this period ensured that the solar panels could not charge the on-board batteries and the base station failed to establish communication with both en-glacial nodes and the Southampton based server.

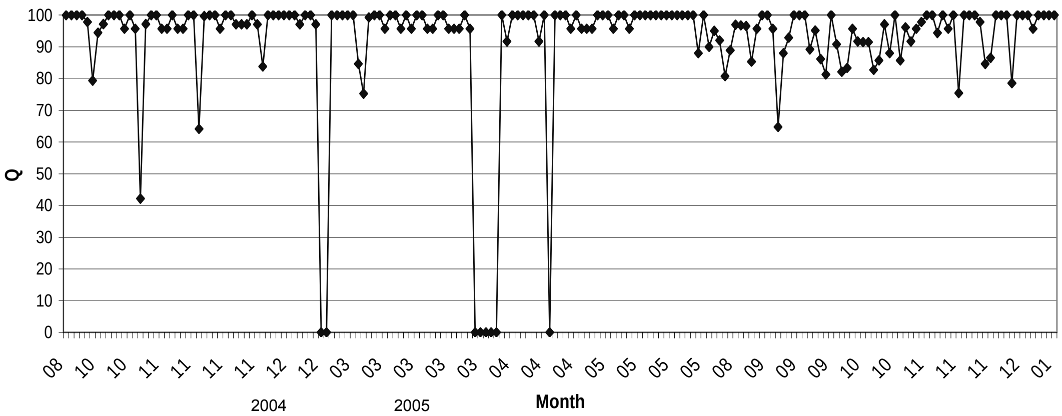


FIGURE 3.6: Percentage of good probe packets over 16 months (10000 packets)

3.3.6 Hardware Constraints

Probe constraints.

A typical sensor node comprises of 4 basic modules. These are a power module, a sensing module, a processing module and a transceiver module. All these units need to fit into a palm-sized module that can be easily dispatched into the glacier's bed through holes up to 70m long and 20cm wide. As shown in Figure 3.7, all the electronics are enclosed in a polyester egg-shape capsule measuring 14.8cm x 6.8cm. The round shape simplified insertion into the drilled holes.

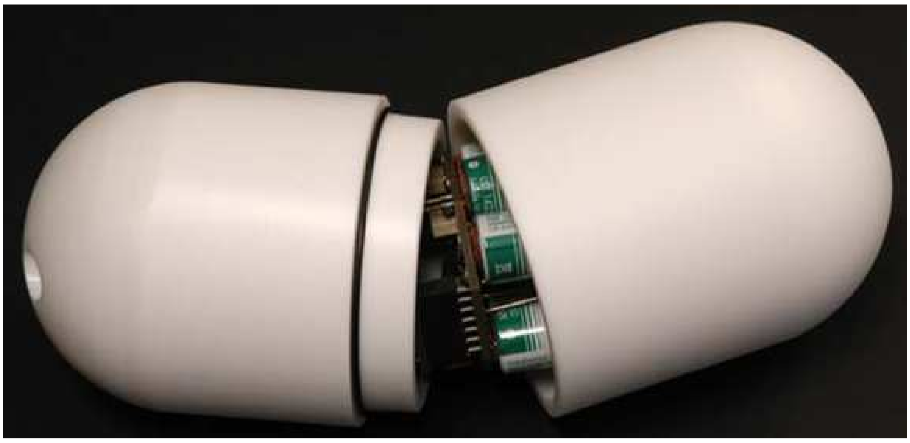


FIGURE 3.7: Probe shown open

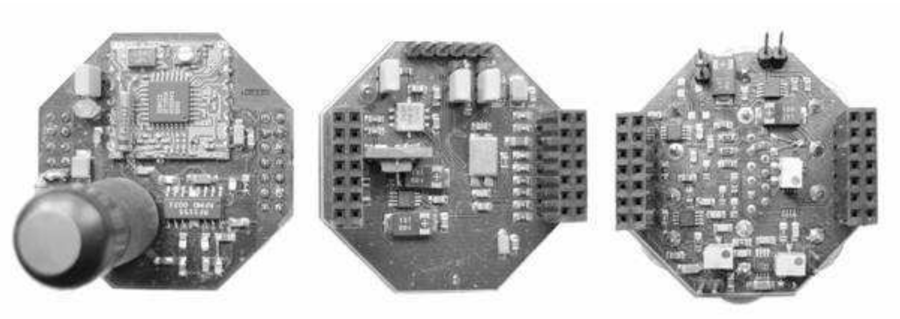


FIGURE 3.8: Radio, digital and analogue sub systems

The probe electronics was initially divided into 3 sub-systems: digital, analogue and radio each of which was mounted on separate octagonal PCBs as shown in Figure 3.8 and stacked together as shown in Figure 3.9. This efficiently utilized the available volume and modularized the design. However, Figure 3.10 shows the 2006-07 design of the probe which uses a single PCB to save further space and a Radiometrix radio module to reduce radio frequency to 173MHz and increase communication range. Figure 3.11 gives an idea about the reduction in size of the new probes which are much easier to fit down the deep holes in the glacier.

PIC micro controllers are low-cost, small sized RISC computers with low power consumption. The probes use these embedded PIC processors to configure, read and store the attached sensors at user-specified times, handle power management and communicate with the base station. The length of the capsule is designed so that it can also accommodate a conventional 1/4 wavelength "stubby" helical antenna fixated on to the radio module.

Base Station constraints

The base station is a very critical aspect of the network as the entire operation of

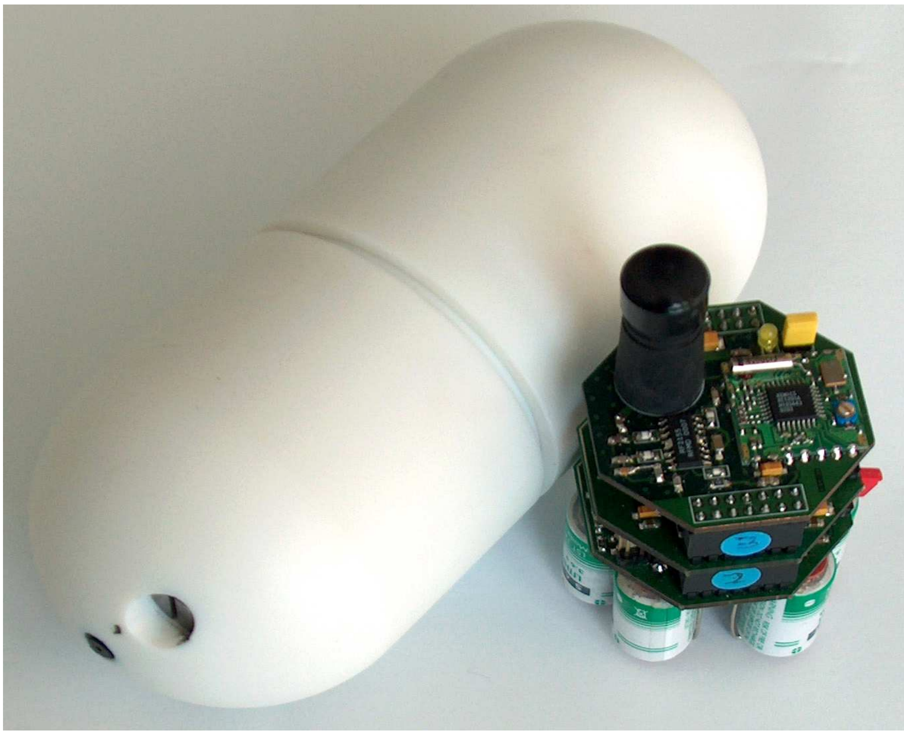


FIGURE 3.9: Pre-2006 probe which used a stack of 3 PCBs shown next to its capsule

FIGURE 3.10: 2006-07 design of the probe that uses a single PCB

the system depends on it. Due to its location on top of the surface of the glacier, several measures are taken in order to ensure safety and efficiency. The base station is held together with the help of a permanent weather and movement tolerant pyramid structure as seen in Figure 3.12. The electronics and the batteries are housed in two separate sealed boxes. Their weight in total stabilizes the entire base station by creating a flat even surface as they melt the ice beneath. The long pole in the middle of the

FIGURE 3.11: Size difference: The new smaller probes fit down the holes easier

pyramid is used to mount the GPS antenna, the wind generator, the long range modem antenna to communicate with the reference station and the anemometer connected to the weather station in the box. The solar panels are attached directly on top of the boxes in order to minimise wind-drag.

3.3.7 Topology

Unlike most sensor networks, GlacsWeb probes are not deployed in an arbitrary fashion. The deployment site of the glacier is surveyed before hand using Ground Penetrating Radar (GPR) to determine any geophysical anomalies (e.g. a sub-glacial river or a puddle) which could hamper communication between probes and the base station. Based on this survey, probes are deployed in holes around a wired probe (attached to the base station directly) suspended approximately 25m into the ice. This is done to enhance communication between probes and base station in ice, a medium where radio propagation range of probes is restricted to 30m compared to 0.5km in air. Radio-wave propagation in pure ice depends on relative permittivity and dielectric loss factor. The dielectric constant of ice at $0\,^{\circ}$C is approximately 3.17 (Glen and Paren, 1975) and the absorption of radio over 100m of ice at $-1\,^{\circ}$C at $100-1000$MHz is less than 10 dB (Evans and Smith, 1969). Budd et al. (1970) and Dowdeswell and Evans (2004) argue there is no significant loss by absorption up to frequencies of 800 MHz. However, glaciers contain sediment, water and air bubbles which may act as communication barriers and

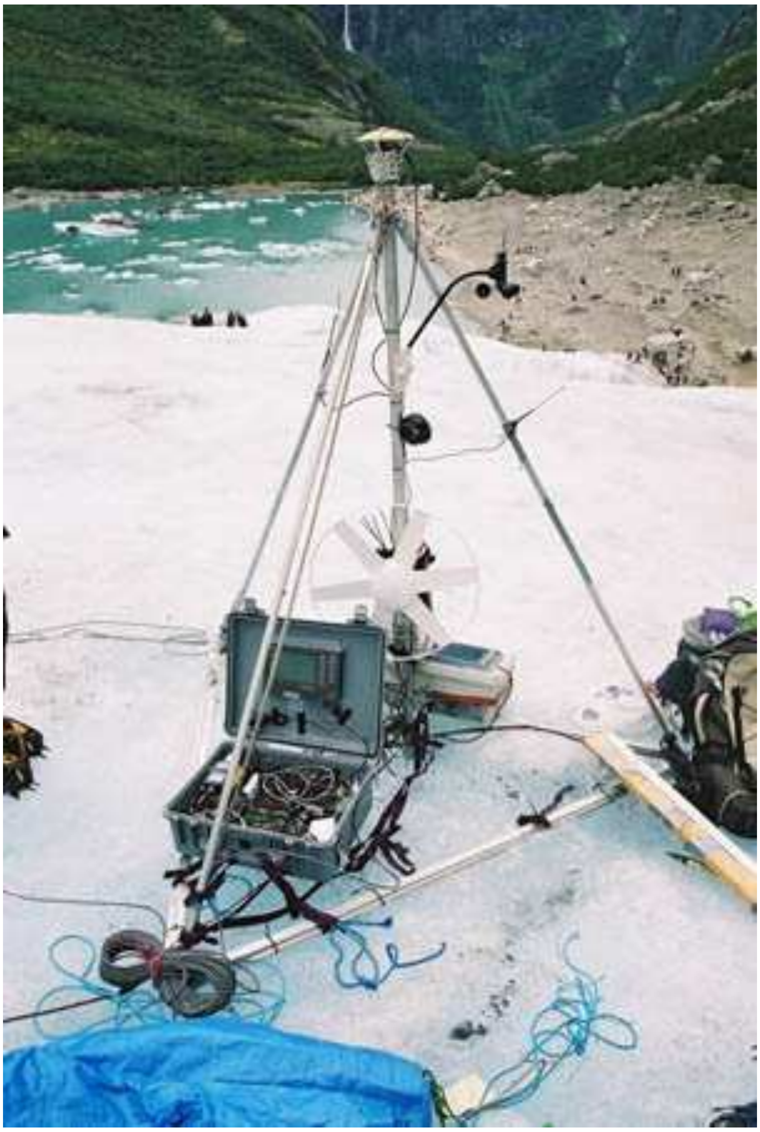


Figure 3.12: Base Station and the pyramid, showing solar panels, battery box, antennas and weather station

sever radio links. Furthermore, the movement of the ice and the continuous change in the sub-glacial environment implies that that the topology of the GlacsWeb network does not remain constant. Some areas of the glacier move differently than other areas resulting in certain probes getting isolated indefinitely from the rest of the network.

3.4 Discussion of GlacsWeb Performance

The hostile and dynamic nature of the environment resulted in a high failure rate of the probes. For example, 8 probes were deployed in August 2004 and although initially the base station managed to collect data from 7 probes, communication was reduced to only 3 probes during the course of the next one year. Another 16 probes were deployed in

August 2005 but only a meagre 3 managed to establish communication for more than 3 months. The following August in 2006 saw deployment of a further 20 nodes but only 5 communicated with the base station before the latter fell into a crevasse disengaging the entire network (See Figure 3.13).

Thus, failure is something that the GlacsWeb network is well accustomed to and the GlacsWeb team have been involved in the administration of the system long enough to assess the reasons behind it. The remainder of this section discusses five important weaknesses for the high failure rate of the network and the remedial actions required to improve upon them.

3.4.1 Range of Probe Transceivers

As aforementioned in Section 3.3.7, the radio signal is attenuated severely in ice and this reduces the range of the probe transceivers to under 30m. Now although the base station is attached to wired-transceivers inserted in the ice to improve data gathering, the loss of communication with the probes implies that the sub-glacial movement of the ice carries the probes so far away that they move out of transmission range.

Furthermore, the movement of ice may carry the probes into englacial water bodies such as a puddle or a river. Also, the holes in which the probes are buried may fill up with rain water resulting in the probes sitting at the bottom of long column of water. Whilst immersed in water, the transmission range of the probes is further reduced secluding them from the network even if they haven't moved from their original deployment position.

Such circumstances, in the highly dynamic nature of the glacier, have called for appropriate measures to intensely modify GlacsWeb's networking protocol. The infrastructure based protocol implies that the probes can only communicate with the base station *directly*. This is not ideal when probes move out of range and therefore it is important to develop a multi-hop ad-hoc network of probes. Whilst multi-hopping does not necessarily make significant transmission energy savings (see Section 4.3) it has an obvious advantage in that probes would cooperate amongst each other to help transmit data from *out-of-range* probes.

3.4.2 Probe Breakdown

The deployed probes can undergo large amounts of stress from the large mass of compact and mobile ice. Although the probes are encapsulated in a polyester shell for protection, they are designed to undergo a maximum stress of 250 psi. Stress beyond this level can result in probes being damaged and losing all communication with the base station. Whilst physical deformation is the most anticipated form of damage, probes can also be damaged as a result of water leaking into the polyester casing and causing a short circuit to the electronics. These causes of failure are very hard to avoid and the only way to overcome them is to deploy more probes that could increase the chances of data gathering. Therefore, it is vital that a decentralised system is in place to cater for the openness and scalability of the system. The failure rates of the probes have been decreasing with successive deployment phases. For example the first deployment phase saw only 1 probe operating for a maximum period of 14 days. However, later phases have seen a few probes last for up to 18 months.

3.4.3 Base Station Breakdown and Time Synchronisation

The base station sitting on the surface of the ice is in as hazardous an environment as the probes inside the ice. The GlacsWeb base station has had its fair share of problems ranging from power failure to physical damage caused by wind and surface movement. On all such occasions, a "dead" base station meant that although probes in the ice were still functioning, data could not be retrieved from them. Figure 3.13 shows a snapshot taken by one of the local climbers after the base station fell into a crevasse and was seriously damaged. On times like these, an expedition team had to be dispatched to attend the glacier and physically reinstate the normal operation of the base station. However, the reinstatement of the base station is not immediate and can take several days due to the remote location of the deployment site. This has a detrimental effect on the probes even after the base station is fixed as the real time clock (RTC) of the probes is heavily dependent on the base station. It is estimated that the probes' RTC may drift up to 2 seconds daily. In order to keep all probes synchronised, the base station updates them every day during its communication window using broadcast packets. The base station RTC itself is set by the GPS once every week. Failure of the base station for long periods, however, suggests probe RTC may drift too much outside the base station's polling window. This could be one of the reasons explaining why many probes went *missing* after the base station was restored.

3.4.4 Polling Mechanism

The hostile nature of the glacier connotes that communication with probes can be very intermittent, i.e. at times radio links between the probes and base station is seamless and on other occasions there is a complete communication loss due to the presence of englacial water bodies. In such circumstances, the polling mechanism is not best suited. This is because unnecessary power is wasted frequently by the base station polling for *lost* probes. Furthermore, probes polled last are depleted of more energy as they have to wait longer before they can turn off their transceiver and go back to sleep. This can be seen in Figure 3.2. This research envisages a much larger GlacsWeb network in the future for which it will be extremely important to manage power and communication

FIGURE 3.13: Base station, heavily damaged after falling into a crevasse in August 2006 (Photo by Valentin Burki)

in a more efficient manner. Therefore a shift from this more centralised approach to a more ad-hoc approach is required.

3.4.5 Set Sensing Rate

Currently, the probes sense and observe the environment at a regular time period of every 4 hours. This time interval has been set by the glaciologists. Although the task of sensing itself is not as power hungry as the task of communication, more sensing implies more data to transmit and hence resulting in more power consumption. Figure 3.14 shows a sample of data sensed by probe 8. All three graphs in the figure have periods where the data is linearly represented. This means that certain aspects of the environment are predictable and that it is possible to infer future data samples from the past observed samples. This implies less sensing and therefore less transmission. Therefore, there is a strong need for an adaptive sampling mechanism installed on the probes that would enable them to sample the environment selectively. The pre-set sampling rate of the sensors (4 hours) is not theoretically justified. Furthermore, this traditional notion of transmitting every single reading is flawed and does not take into consideration the non-transmission of unnecessary data. Chapter 5 discusses in greater detail a mechanism where an adaptive sampling developed for GlacsWeb is combined with a feature that disregards data that is deemed too insignificant or unimportant to transmit to the base station.

Fact File 1. Figure 3.14 shows a sample of data gathered by probe 8 which survived for more than 14 months. The data can be interpreted as follows. The settling-in period of around a month after deployment shows variations in water pressure as the hole closed-up, together with a drop in resistivity when water was present (event A). The readings are more stable by November and there is no relationship between resistance and pressure, indicating the holes have closed. From January 2005 the pressure steadily increased until it was higher than that possible from a column of water filling the depth of the hole: this is over pressure from the ice. The tilt shows the probe moves a lot in the first month then settles into one position. When spring arrives around March 2005 it moves more rapidly showing that the glacier is waking up. There is also a wet event (event B) around that time presumably from increased melting of ice and snow. Mid July (event C) sees a movement change, rapid pressure decrease and less electrical resistance consistent with the presence of lots of liquid water. This is when the glacier moves the most.

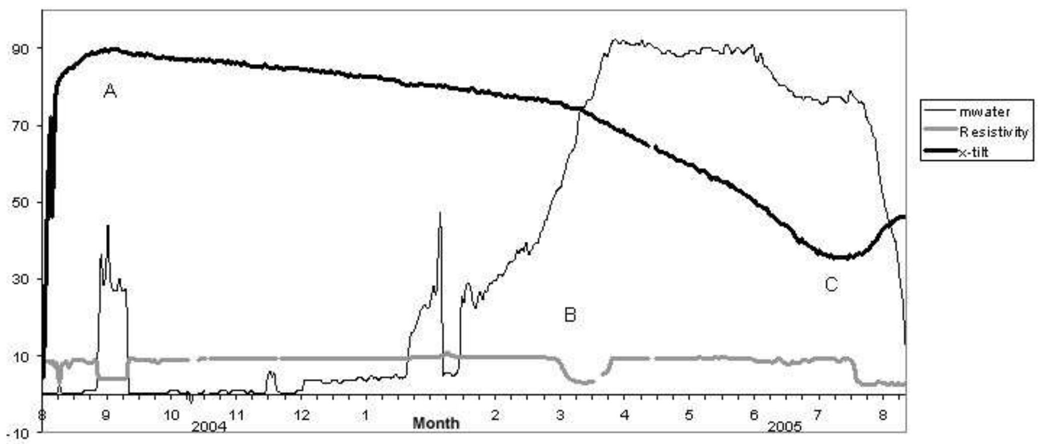


FIGURE 3.14: One years readings from Probe 8 showing water pressure as depth of water (m), resistance (M Ω) and X-tilt in degrees

Having now explained the salient features of GlacsWeb and the design factors affecting its overall architecture, the next section provides a discussion of other similar real life environmental sensor networks and how their success or failure can beneficial to the improvement of GlacsWeb.

3.5 GlacsWeb Vs Other Environmental Sensor Networks

Sensor network research seems like a highly application-specific field where the requirements and constraints of various applications are not yet fully understood. However, deployed applications do share some common characteristics such as raw sensor data transmission over wireless connection, centralized data processing, simple routing scheme and best-effort data transport design. This is a good point of departure to provide a comparative study in order to understand the parallels between various applications. This section aims to provide a discussion of environmental sensor networks similar to GlacsWeb and learn from the experience and shortcomings of other deployments so that the GlacsWeb architecture can be improved further. The applications discussed in this section have served as testbeds or prototypes to identify research challenges for GlacsWeb and verify proposed methods for GlacsWeb. Section 3.5.1 begins with the discussion of the Volcan Tungurahua project used a wireless ESN to monitor volcanic activity by specially-constructed microphones to monitor infrasonic (low-frequency acoustic) signals emanating from the volcanic vent during eruptions. Section 3.5.2 discusses the Gread Duck Island project that installed wireless sensor networks to monitor the nesting habits of Storm Petrels.

3.5.1 Volcan Tungurahua Project

The Volcan Tungurahua project (Werner-Allen et al., 2006) was a similar application to GlacsWeb in that it was also deployed in an extreme environment. It used a wireless sensor network to monitor volcanic activity by specially-constructed microphones to monitor infrasonic (low-frequency acoustic) signals emanating from the volcanic vent during eruptions at Volcan Tungurahua, an active volcano in central Ecuador. This network consisted of five tiny, low-power wireless sensor nodes, three of which were equipped with a specially-constructed microphone to monitor infrasonic (low-frequency acoustic) signals emanating from the volcanic vent during eruptions. The network gathered approximately 54 hours of continuous infrasound data, transmitting signals over a 9 km wireless link to a base station at the volcano observatory.

Volcanos emit powerful seismic waves during eruption which are carefully studied by volcanologists. However, seismic data can also be induced by earthquakes, minining operations and ambulating quadripeds which can complicate observations. For this reason, it is believed that correlating additional observations of infrasonic signals (emmitted near moment of eruption) with the seismic readings can provide better detection of eruption.

The sensor nodes used in this project comprised of five off-the-shelf Mica2 devices developed by Crossbow Technology[1]. Only three of these nodes were assigned the task of

[1]http://www.xbow.com/Products/wproductsoverview.aspx

collecting data using custom-built infrasonic sensors. Sampling at approximately 102Hz, they transmitted data packets containing multiple readings to one receiver mote at approximately 4Hz. This receiver mote connected to a MIB600 interface board forwarded data packets along a long-range serial point-to-point link, provided by a pair of FreeWave modems fitted with 9dBI directional Yagi antennas, back to the observatory 9 km away. To provide the required common time base for the data-collection motes, the fifth Mica2 mote was interfaced with an off-the-shelf Garmin GPS receiver. The time synchronization mote received a time pulse every second from the GPS unit and relayed the pulse to the infrasound motes via radio. Each mote marked the infra sound sample taken when each GPS time pulse was received, allowing the signals from each mote to be synchronized across time.

The Volcan Tungurahua Project demonstrated the feasibility of using wireless sensors for volcanic studies even though the network lasted for a little more than two days. Whilst this seems like a great failure at first glance, it provides a very good case study to highlight two very crucial learning outcomes for the deployment of other similar sensor networks such as GlacsWeb. These outcomes are discussed below.

1. **Unnecessary Sensing and Continuous Transmissions.** The number of nodes deployed at Tungurahua is extremely small (only 3 data gathering nodes). Although this allows for transmission of continuous signals from each of the nodes, such an approach would not be feasible for a larger network such as GlacsWeb (20 nodes) where each node is to required function for at least a year (since new nodes are deployed annually during summer). To save bandwidth and energy, it would be desirable to adopt an approach where data transmission could be ceased during periods of quiescence. The project does attempt to tackle this issue to an extent by designing a mechanism that couples distributed voting amongst the nodes with local detection of events.

   The distributed event detector attempts to transmit only well-correlated signals to the base station. This is achieved by a decentralized voting process to measure signal correlation among a group of nodes where each node sampling data continuously, buffers a window of acquired data while simultaneously running a local event detection algorithm. When the local event detector triggers, the node broadcasts a vote message. Any node receiving a pre-defined number of votes from other nodes during a pre-set time window initiates a global data collection by flooding a message to all nodes in the network.

   The local event detector is based on a based on a pre-defined threshold which is triggered whenever a signal rises above a certain threshold υ and falls below another τ during a time window ω. This kind of a detector relies on absolute thresholds which is sensitive to the particular signal gains on each sensor node. Therefore, it is susceptible to false triggering due to spurious signals, such as wind noise, although the voting scheme mitigates this effect to an extent. Pre-defined thresholds is not a very grounded approach as it may result false positives (which may trigger data collection for uncorrelated signals) and false negatives (which may cause true explosions to be missed). Therefore, a much more structured adaptive sampling needs to be employed which would help reduce unnecessary sensing and help reduce the number of transmitted packets.

2. **Multi-hop Routing** The voting process employed in the network uses a local radio broadcast, while data collection is initiated using a global flood. The expectation is that in a typical deployment each node will have multiple neighbors within radio range with which it can compare votes using local broadcast only. However, to reduce radio contention during data collection, the network uses a TDMA scheme for scheduling transmissions to the collector mote, which in essence is a base station. This is again similar to GlacsWeb in that it is infrastructure based and the problem can be very quickly escalated when nodes either move out of range of the sink node or transmission becomes lossy unexpectedly. This was clearly evident during deployment when a large number of packets (5%) went missing from the recorded dataset. This is a very high number for a network comprising of only three nodes. Furthermore, duplicate packets were recorded on a number of occassions due to lost acknowledgment packets and this caused redundant retransmissions. This further reinforces this thesis's claim that outdoor networks like this are susceptible to lossy communication and that a robust multi-hop network is required to tackle this issue.

3.5.2 Habitat Monitoring

(Mainwaring et al., 2002) deployed a sensor network on a remote island 15Km of the coast of Maine, USA in order to monitor the nesting habits of Leach's Storm Petrel's inhabiting there. The close of integration of wireless sensor networks in this environment provided biological data at densities previously impossible whilst at the same time demonstrating how sensor data can be useful for predicting system operation and network failures. The node and network analysis was based on over one million data packets to develop network reliability profiles and failure models. In total, 32 nodes were deployed in underground nesting burrows as well as well as entrances to these burrows overground. the network logged data from July 18, 2002 through November 18, 2002 (4 months). Nodes were installed with sensors to monitor humidity, pressure, temperature, ambient light and infrared radiation to detect the presence of a petrel.

Engineering and development efforts were minimised by using an off-the-shelf product: the MICA mote (incidentally developed at University of California, Berkeley) which ran on the TinyOS operating system. The sensors took data readings once every 70 seconds which were time stamped with 32-bit sequence numbers and kept in flash memory.

Each reading was transmitted in a single 36-byte data packet using a single channel, 916MHz radio from RF Monolithics to provide bidirectional communication at 40kbps. The nodes were run by an 4MHz Atmel ATmega128 micro controller and were equipped with 512KB of non-volatile storage. A pair of conventional AA batteries and a DC boost converter provided a stable voltage source. The nodes were extremely small in size (approximately 2.0 x 1.5 x 0.5 inches).

The Great Duck Island project is a representative of many applications in this domain and its practical experience with sensor network deployment is a guide to the creation of other sensor networks. However, there are some important challenges that need to be addressed so that users will be able to tailor the network's operation to a variety of experimental setups and allow scientists to reliably collect data from locations previously unaccessible on a micro-measurement scale. These are discussed below.

1. **Need for Autonomous Re-tasking.** As initial data is analyzed, life scientists may be interested in monitoring certain sensors more closely than others. For example, after examining raw thermopile occupancy data, the node could be re-tasked to report only the entrance and exit of the animal. Whilst the project addresses the concern of network retasking by discussing how appropriate changes could be made to simple scalar parameters such the sampling rates, duty-cyle and filters running on each node, performing these particular refinement tasks has to be done manually by completely re-programming the nodes. This process is quite costly it involves reliably transmitting the binary image of the code (approximately 10kB) to all motes that need to be reprogrammed, and invoking a reprogramming application which runs the motes for 2 minutes while drawing about 10 mA. The authors point out that their energy budget only allows them to reprogram the motes only once a day during the 9 month life cycle. As a result, there is a strong need for autonomous distributed algorithms (localised adaptive sampling) to be deployed on the nodes to alter these changes. The authors admit to this and state that the pay-off is huge in terms of increase in life time. This is crucial to an application such as GlacsWeb where nodes can be very expensive to replace.

2. **Need for an Intelligent Routing Protocol.** The communication protocols in this project are simplified as much as possible in order to meet the data delivery requirements. The routing is based on a hierarchical model. Each sensor node acts as a transmit-only device in a single-hop broadcast network. The data is received by the gateway node that operates with significantly more energy capacity than the small sensor nodes and relays packets to the base station. In order to extend the patch to more burrows beyond the single-hop broadcast range, the nodes may form a multi hop wireless network by forwarding each others messages. The project employs coordination algorithms such as SPAN (Chen et al., 2002) GAF (Xu et al., 2001) in order to extend the longevity of the network by selecting representative

nodes to forward data. However, GAF and SPAN do not account for infrequent sampling. Instead they focus on continuous network connectivity and operation. GAF and SPAN are independent of communication frequency, whereas such an application requires increased power savings that may be achieved by adjusting the communication frequency. This is a problem that could be envisaged within the GlacsWeb architecture as well and for this reason an intelligent routing mechanism is required which will consider both communication frequency and the transmission load intermediate nodes are under.

The Volcan Tungurahua and the Great Duck Island projects can be viewed as exciting opportunities to take ideas and solutions developed for the sensor networking community and test their mettle outside of the artificial laboratory environments in which they were created. The GlacsWeb project can learn from the success and failures of such projects. In particular, the 4 specific learning outcomes discussed above are a good point of departure. The long-term plans for GlacsWeb are to provide a long-standing, autonomous sensor array to monitor glacial movement and by addressing these challenges, GlacsWeb can improved further. In this light, the next section summarises the work done within GlacsWeb and identifies key specification requirements that need to be fulfilled to improve it further.

3.6 Summary and Requirements Specification

This chapter has described some of the fundamental elements of a real deployed sensor network called GlacsWeb. The motivation behind the deployment of this sensor network is to monitor a sub-glacial environment as part of an overall research behind understanding climate change. However, the sub-glacial ice harbouring the sensor nodes is extremely hostile and can be detrimental to communication between the nodes.

Risks	**Preventive Measures**
Shorts or damage to sensors	Care in sealing probes
Water ingress	Test seal technique, care when sealing
Communications breakdown	Store and forward protocols
Shorts/Connector problems	Glue, sealant, testing
Faulty software	Testing, dual coding
Wired transceiver damage	Spare transceiver
Fails to go to sleep	Testing. Next day would sleep
Swept away by water	Daily data gathering, more TXR, Ad-hoc network

Table 3.3: Risks faced by probes and their preventive measures

The risks to the probes are summarised in Table 3.3. Part of the system design has been to minimise those risks but it can be seen that many risks are actually managed by construction and management issues. Implementing enough diagnostics inside the probes so that failures can be identified was not possible. Therefore, when probes are "lost" completely it is difficult to know exactly why. Having reduced the margin of improvement in the physical construction of the probes, the focus now must be shifted on its actions: radio communication and sensing. GlacsWeb requires a multiple hop, self-organising ad-hoc network that would improve data collection as well as reduce power consumption. Ideally, the probes would need to be completely autonomous and independent of any manual intervention to perform both reactive tasks (adjusting sensor sampling rates according to change in the sub-glacial environment) and proactive tasks (selecting the most efficient multi-hop route to transmit data to the base station).

Thus, this thesis is now in a position to refine the general requirements that were mentioned in Section 2.9 and present a short summary of detailed requirements of the model/protocol that needs to be developed for the GlacsWeb application in order to make it a more efficient sensor network.

GlacsWeb Requirement 1 (Openness and Scalability) – The high failure rate of nodes calls for a new model that would not be affected in its overall functionality if some nodes fail. Furthermore, the model should be able to automatically incorporate new nodes in the network allowing scientists the freedom to deploy nodes at will.

GlacsWeb Requirement 2 (Decentralisation) – The nodes, once deployed, cannot be retrieved and therefore are required to be completely autonomous and independent of any manual intervention to perform both reactive tasks (such as adjusting sensing rate) and proactive tasks (such as selecting an efficient route to base station). The model should ensure that nodes are able to make independent decisions when facing problems with parts of the network in which they operate.

GlacsWeb Requirement 3 (Communication Robustness) – The model should ensure that the nodes are able to overcome unreliable communication links between nodes and work with low bandwidth. This follows requirement 2 that the nodes should interact and adopt a multi-hop approach in transmitting data to the base station. Furthermore, the model should synchronise communication efficiently between nodes throughout the network.

GlacsWeb Requirement 4 (Intelligent Routing) – The model should manage the task of processing and communication in an intelligent manner. Nodes should communicate in a manner that expends the least amount of energy. The model should also dynamically adapt new efficient routes in the face of changing network topology. It should also implement a fail-safe mechanism to ensure that data reaches the intended destination when communication fails between nodes in the routing path.

GlacsWeb Requirement 5 (Intelligent Sensing) – The model should coordinate activities of the physically distributed sensor nodes to collect data in an efficient manner. In the absence of statistical models of real world-processes the model should have the capability to predict future samples based on the historical observations and be able to distinguish the important (informative) observations with the not so important ones.

Chapter 4 addresses requirements 1 and 3 by proposing a protocol for GlacsWeb at a much lower level of abstraction of the sensor network protocol stack (see Figure 2.1). This includes focusing the research on the physical and data-link layers of the GlacsWeb sensor network. In particular, Chapter 4 discusses a GlacsWeb tailored 2-tier network architecture where TDMA based MAC protocol is proposed in order to improve upon the existing polling mechanism.

Chapter 5 further consolidates on requirement 1 in addition to addressing requirements 2, 4 and 5 by focusing the research on the networking layer of the protocol stack. Here, multi-agent techniques are described to design a new protocol tailored for GlacsWeb that combines the tasks of both sensing sensing and communication through a single utility function.

Chapter 4

GWMAC - A MAC Protocol for GlacsWeb

Chapter 3 discussed the overall architecture of the deployed GlacsWeb sensor network and the challenges it had to encounter whilst in operation. It also highlighted some important model requirements (see Section 3.6)that need to be fulfilled in order to enhance the performance of the overall system. Of these requirements, one in particular, asserted the importance of introducing multi-hopping and ad-hoc networking within GlacsWeb. Like most sensor networks, however, GlacsWeb is also a shared-medium network where nodes have to gain access to the medium in turn before they can transmit their packets. Therefore, in such a network, where several nodes are required to independently sense, transmit and relay each other's data, it is extremely important to have a data-link infrastructure to ensure that node-to-node communication is carried out in a reliable, fair and power-efficient manner.

Chapter 2 discussed the four fundamental tasks of a good Medium Access Control (MAC) protocol. These being,

- Avoid collisions so that two interfering nodes do not transmit at the same time.
- Eliminate overhearing so nodes don't receive packets that are not intended for them.
- Minimise the transmission of control packets to reduce transmission overhead.
- Reduce idle listening so nodes don't waste energy in having their transceivers on without good reason.

To this end, in this chapter presents an efficient MAC protocol called GlacsWeb-MAC (GWMAC) for the ad-hoc probes to be deployed in the GlacsWeb network. This protocol is designed to replace the original and abstract polling mechanism used by the base station to communicate with the probes. The primary aims of GWMAC are to reduce energy consumption and achieve multi-hopping with collision avoidance capability. It achieves these aims by completely eliminating contention between probes and utilising a centralised scheduling algorithm.

Whilst the importance of decentralisation within sensor networks has been explained previously, this chapter argues that at this level of abstraction, it is better to implement a centralised scheme in a mobile and dynamic network. The topology of a network such as GlacsWeb may change frequently requiring the network to re-establish neighbours and re-organise communication links and time slots. Intuitively, a decentralised scheduling approach results in contention and hence collisions in the network configuration stage itself. This phenomenon is demonstrated later on in this chapter in Section 4.3. Therefore, the role of decentralising GlacsWeb (or a sensor network) is assigned to a more higher level of abstraction which includes the task of adaptive sampling and routing. This layer of abstraction is discussed in detail in Chapter 5.

The remainder of this chapter provides a detailed description of GWMAC. The chapter begins by highlighting some of the GlacsWeb attributes that need to be taken into consideration for a suitable MAC design. Having described these key characteristics, the chapter proceeds to the design GWMAC in Section 4.2. Given that the design of any MAC depends on the data-link layer, a brief overview of key physical layer issues that influenced GWMAC is provided in (Section 4.2.1). Sections 4.2.2 and 4.2.3 present the detailed architecture of GWMAC. Following this, in Section 4.3 evaluates the performance of GWMAC in comparison to some well received MAC protocols in literature. The chapter concludes with a summary of GWMAC and its advantages.

4.1 GlacsWeb Attributes to Consider

The study has already identified the main sources that cause inefficient use of energy within a sensor network. In addition, however, it also needs to identify the specific trade offs that would have to be made whilst designing a new MAC protocol. Chapter 2 discussed several MAC protocols designed and developed for various types of communication networks. These included TDMA-based protocols, CDMA-based protocols, contention-based protocols and hybrid protocols such as SMAC and LMAC. All these protocols achieved the primary goal of energy efficiency. Nevertheless, these protocols are tailored for highly specialised networks with specific attributes. Therefore, before we proceed further, the main attributes of GlacsWeb that were taken into account need to be discussed before designing a any specific MAC protocol. These attributes are discussed below.

1. **Energy Efficiency.** The GlacsWeb nodes are battery powered and once they are deployed there is no possible way of retrieving them. Thus, there is no way of recharging or changing the batteries. In addition, each node costs up to £170 to make. This cost is significantly high in comparison to most of the off-the-shelf nodes in the market such as Mica motes (Technology, 2008) and Chipcon nodes (Incorporated, 2008) that may cost as low as £20. This cost is almost 9 times more than the market nodes and therefore prolonging network lifetime for these nodes is critical.

2. **Scalability.** The glacier is a highly hostile environment. The GlacsWeb network experiences a high failure rate. Some nodes fail over time and new nodes are deployed (added) every summer. Furthermore, nodes are continually mobile resulting in a continuous change in the size, density and topology of the network. Therefore, a new protocol had to be designed which would be scalable enough to easily accommodate such network changes. However, the network of earlier version of probes still had to be maintained separately as the technology of the new probes had changed considerably.

3. **Fairness.** In many traditional networks, each node desires equal opportunity and time to access the medium for their own application. In GlacsWeb, however, all nodes are expected to cooperate for a single common task as there is only one application at stake. With adaptive sampling Chapter 5, some nodes may have dramatically more data to send than others during certain times. Therefore, fairness was not deemed to be an important issue so long as application-level performance was not degraded.

4. **Latency.** GlacsWeb is not a real-time system and therefore its application is expected to have long disconnected periods that can tolerate high degrees of latency. The end users of the system, i.e. the glaciologists, attach high importance to the data gathered itself rather than how quick they receive it. To them it does not matter whether they receive data gathered by nodes within one week or one day as long as they receive it. Therefore, latency was regarded as a secondary attribute during the design of GWMAC.

5. **Throughput.** Applications demanding a longer lifetime usually tend to accept lower throughput. This may either be due to hardware constraints or the power required in dealing with high data rates and errors. Low throughput can have a detrimental effect on the performance of schedule-based protocols since longer time slots have to be used. GlacsWeb's consistent improvement in its radio transmission range to reduce errors meant that the nodes had to constantly thrive for an ideal throughput value limited by the selected radio transceiver's capabilities.

4.2 GWMAC Protocol Design

Taking into consideration, the several characteristics of GlacsWeb that make it different from other sensor networks and in addition to the attributes discussed above, this study developed GWMAC, a protocol consisting of three major components. Namely, these are *Single Communication Window*, *Collision Elimination*, *Overhearing Avoidance* and *Acknowledgment Omission*. In the sub-sections that follow, each is explained in greater detail. However, before this study proceeds further, it is important to briefly describe some important details of the physical layer which will enables better understanding of the protocol.

4.2.1 Physical Layer

Chapter 3 discussed the importance of the choosing the appropriate radio infrastructure. In fact specifying the radio frequency, choice of transceiver and size of antenna have been long-standing challenges in the GlacsWeb project. Earlier deployments of GlacsWeb network used radio frequencies of 968MHz and 433 MHz that resulted in very lossy communication due to presence of en-glacial water bodies. This prompted the research to use a transceiver with a lower frequency to enhance the communication signal. Unfortunately, due to legal requirements[1], the GlacsWeb team could only use license-exempted channels and this restricted the choice of transceivers for lower frequencies. In addition, commercially available wireless sensor network hardware platforms do not support frequencies lower than 433MHz. Therefore, a standalone transceiver module had to be obtained that could be incorporated into the node design.

After a thorough investigation a single channel transceiver module manufactured by Radiometrix called BiM1 was chosen. This transceiver module operates in the 173.25 MHz band and is license-free for general applications. A major challenge faced with this module was to find a suitable antenna because it was discovered that most 173MHz commercial antennas are designed for applications in air only. It was found that their performance degrades severely inside a glacier because of the different dielectric properties of ice. Furthermore, they were too long (75 mm) to fit inside the polyester probe casing. Therefore, a compact 173 MHz helical antenna was designed which had to be manually tuned to the ice through the laborious process of trial and error.

The BiM1 module offers a maximum bit rate is 10Kbps. However, Manchester encoding was used to reduce the error rate and this reduced the throughput to 5Kbps providing a byte time of 1.6ms. Although this is very slow compared to other conventional

[1] The Brikdalsbreen glacier is located inside Jostaladsbreen National Park in Norway and is a major tourist attraction. The research carried out in this area had to be cleared by the Norwegian government and the GlacsWeb team were unable to obtain necessary permission to use a wide variety of radio frequency

transceivers working at higher frequencies, this disadvantage was compensated by an important advantage. BiM1 module is capable of measuring the received signal strength (RSSI) over a range of 60dB or more and this is extremely useful for the purpose of designing a routing algorithm (at higher level of abstraction of the protocol stack) in order to decide the best transmission paths and gateways for each node in the network. The BiM1 module has three operational states: transmit (TX), receive (RX) and standby (sleep). The performance parameters of the module are shown in Table 3.1.

A custom packet structure was used as shown in Figure 4.1. Each packet is made up of 64 bytes. The header comprises of 9 bytes leaving the payload to be 55 bytes. This structure allows a complete data sample to be stored in one packet whilst at the same time it is small enough to reduce errors. Figure 4.1 figure also shows the default time slot to be 130ms. With a maximum packet size being 102.4ms (64 byte x 1.6ms) it allows 27.6ms for the preamble and guard band to switch ON and stabilize the transceiver. It also compensates for unexpected time drift.

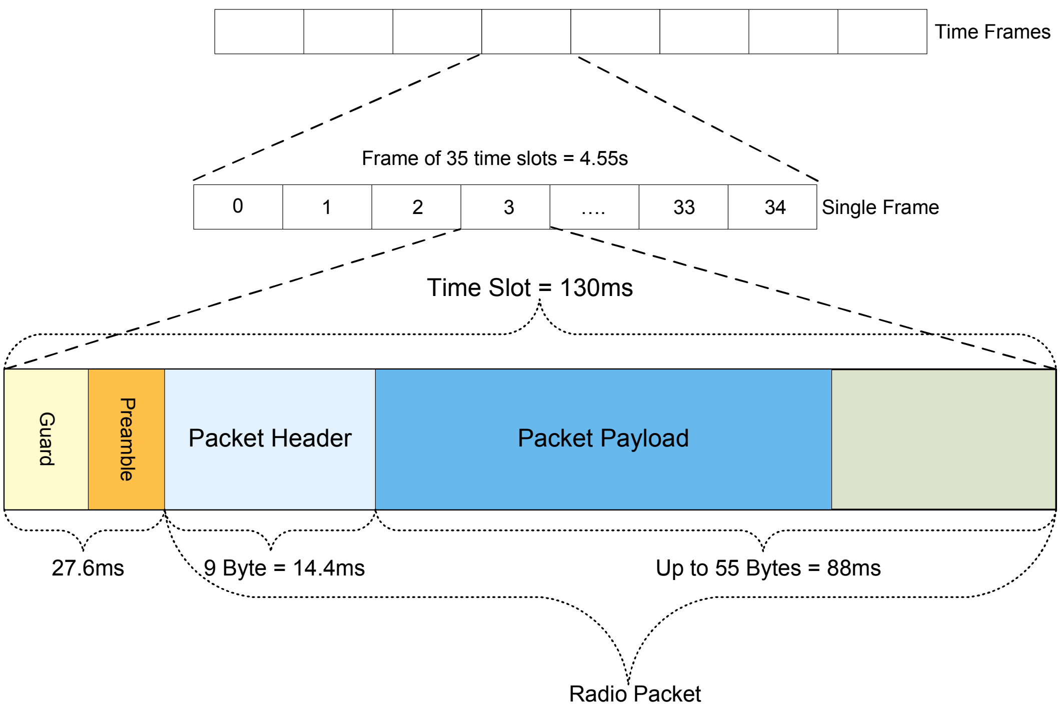


FIGURE 4.1: Times Slots, Frames and Packet Structure

4.2.2 Limited Communication Window

In many sensor network applications, there are several periods when no sensing occurs. This results in a low data rate during such periods. Switching on the node transceivers at these times causes a great amount power wastage through idle listening. Benchmark protocols such as S-MAC, L-MAC and T-MAC reduce energy consumption through idle listening by letting the nodes go into periodic sleep mode. For example, if in each second a node sleeps for half a second and becomes active for the other half, its duty cycle (and hence energy consumption) is reduced by 50%. However, these protocols are catered for sensor networks with near-real time use where the environment or target requires to be monitored very frequently (every few seconds). This has to be complimented with frequent radio communication to transfer all the data.

In contrast, GlacsWeb is not a real time system. The glacier is a slowly changing environment that does not require monitoring every few seconds like in the case of a surveillance network. Therefore, it requires less sensing and hence less communication. The GW-MAC protocol reduces the duty cycle of the probes to almost zero by circumscribing only one small communication window per day. In this manner, the probes activate their radio transceivers to communicate for a maximum of 1 minute daily and sleep for the remainder of the time.

4.2.2.1 Basic Scheme

The basic scheme is shown in Figure 4.2. The timeline shows a node frequently sensing (downward arrows) and communicating only during the communication window. Nodes sleep for the majority of the time and wake up only during the communication window daily to transmit (or relay) data packets they observed (or received from other nodes) in the period prior to this window. Even during the communication window, the nodes have their transceivers turned off for the major part (Idle state) and only turn them on during time slots in which they are expected to receive data or slots uniquely assigned to them for transmission. Prior to going to sleep, the nodes set a timer to wake up for the next communication window. The number of communication windows can be varied according to different applications. For GlacsWeb, one communication window per day was found to be sufficient enough to allow all nodes to transmit the maximum permissible data collected by them. This cap on the maximum permissible data was defined by the glaciologists who were interested in the final data.

4.2.3 Collision Elimination and Overhearing Avoidance

Like most TDMA based protocols, GWMAC also divides time into frames. These frames are further divided into slots as shown in Figure 4.1. The number of slots in a frame is determined by the number of nodes present within the network. In other words, each node is assigned a slot for transmission. If nodes enter or leave the network, the total number of slots are dynamically increased or decreased respectively upon the *network discovery phase* (See Section 4.2.3.2 and Section 4.2.3.3). Furthermore, nodes can be reassigned different time slots depending on the topology of the network so that each communication frame is used in the most efficient manner.

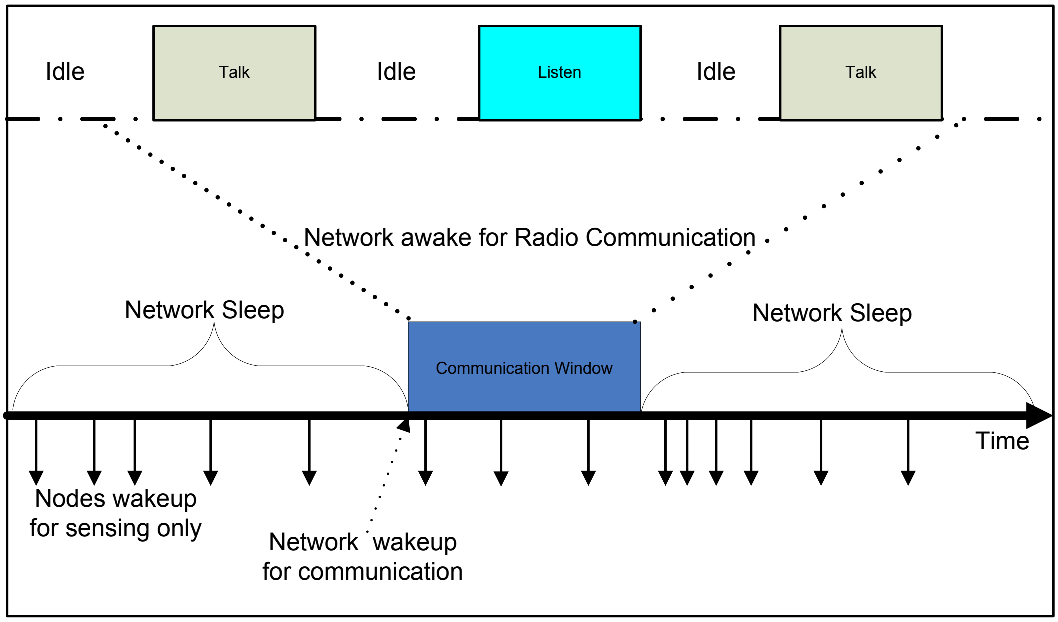


FIGURE 4.2: Single Communication Window

4.2.3.1 Base Station and Anchor Nodes

The task of assigning a time slot to a node is delegated to a scheduling algorithm on the base station. The algorithm assigns time slots to nodes in sequential order according to their hop level. In other words, nodes further hops away from the base station are assigned earlier time slots compared to the nodes fewer hops away. The assignment algorithm makes sure that each time frame is used uniformly with no empty slots. It also guarantees the delivery of any data packet from the base station to any node or vice versa in just one super frame. This information about nodes and their newly assigned time slots along with a list of their possible next hop destination nodes is then broadcasted over the network.

In order to enhance communication between the base station and the ice-embedded probes, the former is connected to n wired probes called *anchor* nodes as shown in Figure 4.3. These anchor nodes, although fully controlled by the base station, are also embedded in the ice and communicate with the remaining wireless probes using the GWMAC protocol. The network hops are organised around the anchor nodes based on RSSI values between probes.

Network discovery and configuration are only initiated during the communication window. The initiation is carried out time to time (every 1-7 days) depending on the system behaviour and time of the year (rainy periods may require more frequent initiations). This is to ensure nodes are not lost and multi-hopping routing is always optimised in the face topology change. This following sections describes the phases through which the network nodes are discovered and configured to carry out the ad-hoc network activities.

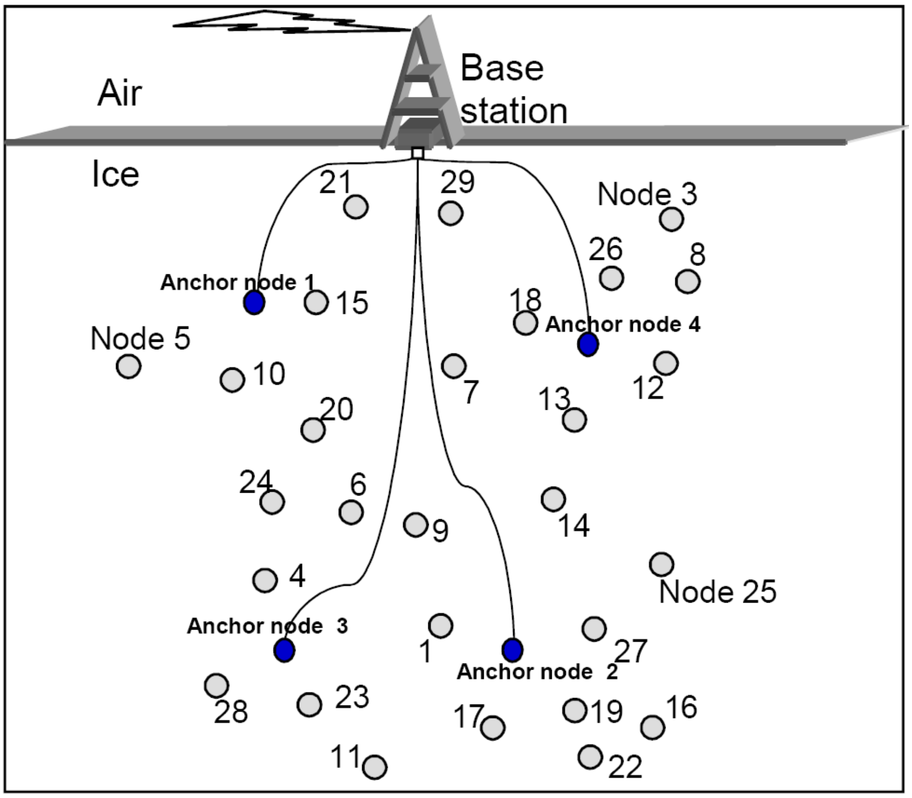


Figure 4.3: Network Setup with GWMAC

4.2.3.2 Network Discovery

This phase combines the use of TDMA with an optimised ad-hoc flooding technique to broadcast messages and retrieve logistics information from the probes. This combination overcomes the potential problems of message redundancy and packet collisions as a result of flooding (Ni et al., 2002). Network discovery takes advantage of a *default* network schedule where each probe is allocated a time slot equal to its own ID. For example, in Figure 4.3 anchor node 1's default slot is 1, node 15's default slot is 15. This ensures that even before the network can start organising, each time slot is unique to any node thereby avoiding any collisions. The base station initialises the process synchronising all the *active* anchor nodes[2]. At most, 3 different command messages are traversed through the network to collect information about the network structure. These are the following:

1. *Direct Echo (DE)* The anchor nodes commence network discovery by broadcasting a DE command in their respective slots as shown in Figure 4.4. Each non-anchor node that receives this command replies to the corresponding anchor node in its own default time slot by transmitting a *direct echo reply* (DER). The base station holds a list of all deployed nodes and is able to establish all nodes within one-hop range along with those that are not (missing probes i.e. probes that could not send communicate directly with the anchor nodes).

[2]some anchor nodes can be damaged or disconnected through wire cuts

2. *Spread Echo. (SE)* If nodes are deemed missing, the next step involves the anchor nodes transmitting a second discovery message called spread echo. Each node receiving this command records the ID of the transmitter along with its RSSI and retransmits / broadcasts the same command in its own default time slot, even if it has to wait for the next frame to do so. This flooding technique makes sure that the spread echo command is disseminated through the entire network. The total number of frames or depth of exploration is controlled by a *time-to-live* parameter embedded within the spread echo command packet and decremented with each forward transmission. This flooding stops when the parameter decrements to zero. The time-to-live parameter is very useful in providing the flexibility to reduce the depth of network during periods of good communication and increase the depth of the network during rainy periods that lead to poor communication.

3. *Spread Echo Reply.* This command is initiated by those nodes that when receive an SE command and realise that the time-to-live parameter has decremented to zero. Each node receiving an SER adds its own recorded list of received IDs and RSSIs to forward it back to the node it first heard the SE command from. The aim of this scheme is to make sure that node IDs and RSSIs from the entire network reach the base station (anchor nodes) in the quickest manner. This is also disseminated through the network in the same manner as the spread echo command.

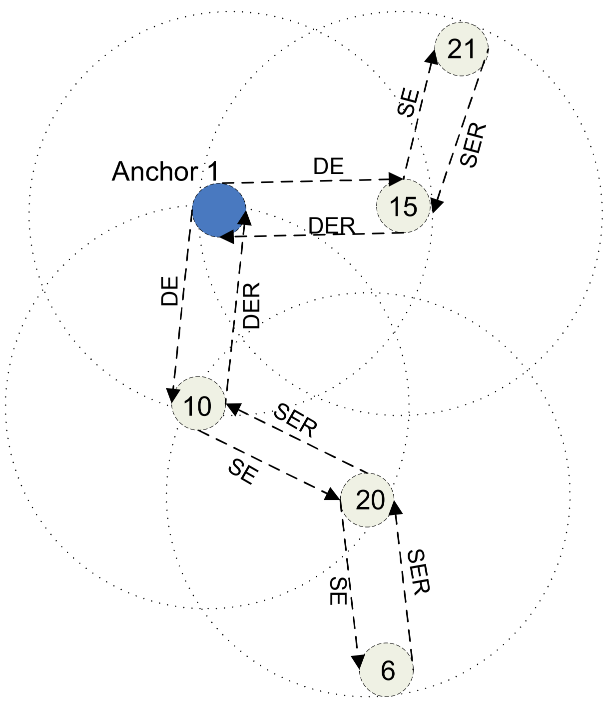


FIGURE 4.4: Network Discovery Messages

4.2.3.3 Network Configuration Phase.

The base station analyses the RSSIs and IDs it receives from the network discovery phase and uses them to perform the following tasks:

- *Assign optimised time slots.* The scheduling algorithm running on the base station assigns new time slots to nodes in sequential order in the time frame. The assignment is based on the network hop level of the nodes. Figure 4.3 shows how nodes in nearer hops occupy the earlier time slots than nodes further hops away. This assignment algorithm makes sure that each frame is used uniformly with no empty time slots. It also guarantees the delivery of any message from the base station to the nodes or vice versa in just one super frame.
- *Assign node gateways.* Based on RSSI values, each node is assigned one parent to communicate through. The algorithm also attempts not to overload any of the parent nodes with too many children so long as there are alternative routes.

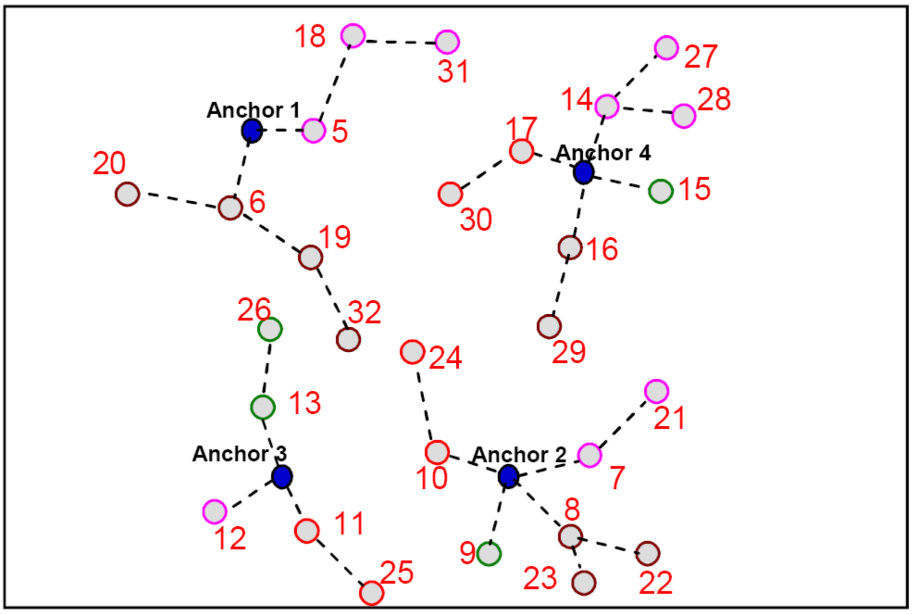


FIGURE 4.5: Network shown in Figure 4.3 after configuration

The above configuration information is then broadcast by the base station to the entire network in a sequence of four different command messages as follows:

1. *Assign slots.* The list of newly assigned time slots is sent to the entire network in this command message. This is broadcast in the same way as the network discovery commands. All commands here on are broadcast to the entire network in one super frame making use of the newly assigned time slots.

2. *Assign gateway.* A list of nodes with newly assigned parents and/or children are sent to the network in this command.

3. *Reply frames.* Information about which hop level each node lies in, is set in this command.

4. *Set parameters.* This command message is loaded with additional changes in the network parameters such as number of nodes, number of time slots, number of frames, current date and time, next time of wakeup, in activity time out and time of sleep.

At the end of this phase the probes in the network have all the information they need to know and who to route data through. The section now proceeds to detail the data acquisition phase.

4.2.3.4 Data Acquisition Phase.

Data acquisition is automatically scheduled immediately after the network configuration phase. However, if the base station believes that the network topology might not have changed since it last communicated to the network and therefore does not require configuring the network, it may choose to commence data acquisition at the start of the communication window itself. The process is initiated by sending a *Get Data* command. This is referred to as the *down-link* mode in which all nodes in the network use their recorded configuration to forward this command, in their designated time slot $S_{assigned}$, to their children. The entire command down-link lasts for one super frame. This is followed by the *up-link* mode where each node i transmits data (if there is any) to its parent in time slot S^i_{uplink} such that

$$S^i_{uplink} = S_{frame} - S^i_{assigned} \tag{4.1}$$

where S_{frame} is the total number of slots in a frame. Again, this ensures that each packet is transmitted from the originator to its intended recipient (usually the base station) in one super frame. In their reply packets, along with the data, the probes indicate how much more data needs to be sent or forwarded from others so that the number of up-link frames are adjusted accordingly for all those probes that fall in that multi-hop path. Only the probes that are assigned to more up-link frames stay awake for consequent data packets whilst the rest of the network is put to sleep.

4.2.3.5 Energy Savings

The TDMA scheme is adopted during both network configuration phase as well as data acquisition phase thereby eliminating any form of collisions. One may argue that collision may occur when a long lost node suddenly returns to the network expecting to transmit in its originally assigned time slot, which may or may not have been assigned to another probe since. This problem is rectified by programming the nodes in such a way that they are forced to switch to a *listen only* mode if they do not (or cannot) participate in any form of communication within the network for a minimum threshold time Φ. This mechanism forces a *lost* node to seek the network rather than waste energy in advertising its existence. Furthermore, additional energy is conserved as the TDMA schedule used completely eliminates overhearing in the nodes. This is because nodes only turn on their receivers during their parents' and children's time slots and sleep the rest of the time.

Section 4.2.1 mentioned how the use of the BiM1 radio transceiver significantly reduces the bit rate. This had a significant impact on the use of control packets in GWMAC. The size of time slots, 130ms, used in this TDMA scheme is very large compared to slot sizes used in other protocols - 5ms (Dam and Langendoen, 2003; Ye et al., 2002a). This value is 26 times greater and justifiably so because size of larger packets by the GlacsWeb nodes. However, incorporating acknowledgments packets within the scheme to confirm receipt of data or command packets would further increase the size of the slots. Therefore, it was decided the use of acknowledgment packets should be omitted all together. This decision was made keeping in mind that the communication paths inside the network were RSSI dependent and that nodes would only communicate if the RSSI links would guarantee successful transmission. If, however, for any reason a node fails to transmit successfully then the base station can identify this and request data from the node again.

4.2.3.6 Custom Network Commands

The base station or the network administrator can schedule some additional network commands when the network is awake. These commands may include changing certain network configuration parameters such as the number of time slots, number of nodes, inactivity timeout and the next wake up and sleep times. It can also include commands to force nodes to take a set number of sensor readings over a period of time. There are also commands to update node firmware.

4.2.3.7 Maintaining Synchronisation

Each node has a real-time clock (RTC) that keeps the time (hours, minutes, seconds, milliseconds) and date (day, month, year). In addition each node also has a millisecond timer inside the microcontroller which synchronises with the RTC millisecond timer every 1 second. A tight TDMA schedule requires synchronisation amongst all the nodes. For this, a unique algorithm is used to synchronise the entire network at start up and, additionally, each time a command packet is received. All nodes are assumed to be unsynchronised when they wake up to communicate. In order to get them synchronised, the base station takes the first initiative by synchronising its own RTC with the average of the closest RTCs of the anchor nodes. The remaining nodes are synchronised through the diffusion of any message packet initiated by the anchor nodes. When a node receives a message packet it can uniquely determine its clock by considering the following

- **Time at Transmission**. This millisecond time stamp value is denoted as $T_{transmit}$ and is embedded within the packet.
- **Time of Flight** or the time taken for the packet to reach the receiver from the transmitting node and is denoted as $T_{preamble}$. Calculating its value is not easy. However it is easily compensated for by the available guard time or time of switching (see next).
- **Time of Switching** or the time taken for the transceiver to switch on and stabilise represented as T_{switch}. This is also illustrated as the guard time in Figure 4.1
- **Time of Processing** which is represented as ι and is so small that it may be considered negligible.

Thus, a node can be synchronised to the current time $T_{current}$ using the following equation

$$T_{current} = T_{transmit} + T_{preamble} + T_{switch} + \iota + (1.6ms * (H + P)) \tag{4.2}$$

where H is the length of the packet header and P is the payload.

4.3 Evaluation of GWMAC

GWMAC was implemented on a test bed of 10 GlacsWeb probes to validate its operation. However, it was difficult to evaluate its effectiveness and relative performance in comparison to existing protocols through the test bed alone. This is primarily due to the voltage profile of the battery cells (See Section 3.3.2) that were used for powering the nodes. As can be seen from Figure 4.6 these cells maintain the same voltage until they drain completely making it difficult to calculate energy consumption and assess network lifetime.

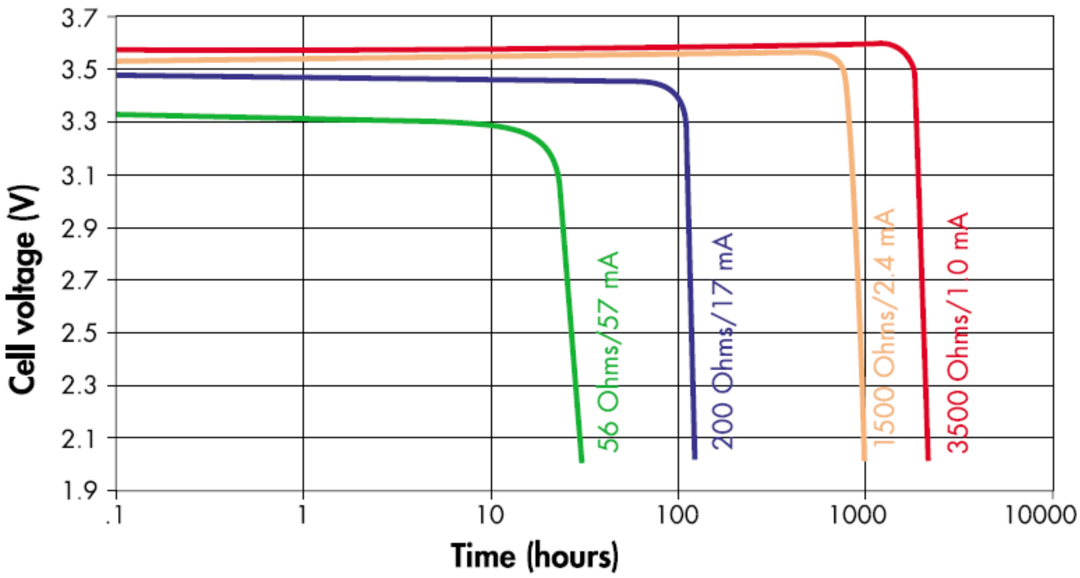


FIGURE 4.6: Voltage profile of the Lithium Thionyl Chloride cells

However, in order to gain some idea about the network lifetime of the node using GW-MAC, a simple accelerated experiment was conducted. A simple two-hop network was setup with only two nodes (A and B). Packets from source A flow through node B and end at the base station, while B originates its own packets and transmits it to the base station. Since the transmission range of the nodes was quite high, the base station was able to hear node A as well. However, for the purpose of this experiment, the base was configured to ignore packets received directly from A until B died (after which any packets from A would be picked up).

Node A was programmed to take sensor readings twice every second and transmitted each set of readings (64 bytes) to B immediately. Node B also took sensor readings twice every second and transmitted its own readings as well as the one received from A to the base station. The protocol selected the uplink frame size to be 440ms which included 50ms for sensing and three communication slots of 130ms (one for A and two for B). This meant that the A's duty cycle was 36% and B's was 88%. The power consumption of the each node is provided in Table 3.1. The experiment ran until the base station stopped receiving packets from either node.

Table 4.1 illustrates some of the results of this experiment. Node B exhausted its energy resources for approximately 68 hours. Node A continued transmitting directly to the base station for another 57 hours. The successful transmission for both nodes was approximately 95%.

The results of the experiment validates two things:

1. Multihopping does not necessarily help reduce power consumption where transmitting directly to the base station is possible. This is clearly evident from the

Table 4.1 as node A continued transmitting for almost twice as long as B. Node B, on the other hand, had to pay the price of receiving packets and retransmitting them on top of its own transmission cost. However,

2. Practical lifetime is less than theoretical lifetime. Theoretically, node A (equipped with 21Wh or energy) should have continued transmitting for another 22 hours. However, the battery voltage profile does not allow for this as it drops drastically after a certain time. See Figure 4.6.

	Node A	Node B	Base Station
Duty Cycle	36%	88%	100%
Total Packets Originated (estimate)	901440	493200	0
Total Packets Received	0	468540	1322459
Total Packets Forwarded	-	468540	-
Lifetime of the network	≈125 hours	≈68 hours	-

Table 4.1: Experiment Results

The results of this simple experiment are optimistic for GlacsWeb as the duty cycle of the nodes is significantly less in the actual deployment. In other words, each node would require to transmit at most 24 packets a day in contrast to 2 every second (or 172800 a day). This would mean that a deployed node should provide data for at least a few years. In order to get a more detailed evaluation of GWMAC, a customised sensor network simulator in Java was developed. This was done to compare the performance of GWMAC against other protocols. For this study it was decided to compare GWMAC against 3 other well known MAC protocols discussed in Section 2.5. These are namely S-MAC, T-MAC and L-MAC.

The network in the simulations was subjected to change in size (number of nodes) and change in average traffic (number of data packets originated per node) in order to plot performance graphs of all 4 protocols. These performance plots represent how network lifetime is affected by each protocol and illustrates the volume of data packets collected by the base station over network lifetime. Usually, network lifetime is defined as the time span from deployment to the instant when the network is considered non-functional. However, at what point in time should a network be considered non functional is application-specific. For the purpose of simulations and for the remainder of this thesis the following definition for network lifetime is adhered to.

Definition 4.1. Network Lifetime is defined as the time taken for 50% of the initial number of agent nodes in the network to die.[3]

[3]A formal definition of network lifetime is not straightforward and depends on application scenario in which the network is used. The simulations suggested that failure of approximately 50% of the nodes (almost all of them being intermediate i.e. closer to the base station) led to little or no connectivity with the network. Hence, the definition refers to the capability of the network to provide the services it was designed for (i.e gather data from the nodes).

In addition to analysing network lifetime and volume of data gathered by the base station, the simulation also compared the energy expenditure of the nodes and the time taken by the network during the network discovery phase. To plot the result graphs, 3 different sets of simulations were conducted. The first set of simulations was run to establish the average network lifetime of the network using each MAC protocol. The second set of simulations was conducted to demonstrate the total data collected by the base station for each MAC protocol over the lifetime of the network. The third set of simulation was run to demonstrate the total energy consumed and time taken by the entire network during the self-organisation phase. However, before the study proceed to the results of these simulations it is necessary to provide some background on the experimental setup and details of certain network parameters.

4.3.1 Simulation Setup

The wireless sensor network simulator was customised and designed independently in Java to provide more flexibility of use. It provides a virtual environment in which sensor nodes can either be scattered randomly or inserted at specific locations. The nodes take one of the following actions in a single time period: *sense* (sensor read), *idle listen* (where a node enables its radio antenna so that it is ready to receive data or for carrier sense), *transmit*, *receive*, and *sleep*. All actions have a set power consumption value affixed to them. The radio propagation model in the simulation was assumed to be symmetric. The study decided to ignore the processing action of the node due to its near negligible power consumption. Specifically, Table 4.2 shows the typical energy consumption of each action based on the values obtained from the designed probe platform. All nodes were initialised with an energy capacity of 1000 Joules. Four equidistant anchor nodes were placed roughly around the centre of environment.

The basic functionalities of S-MAC were incorporated in the simulation with the presence of both the message passing module and periodic listen and sleep. The S-MAC listen time and L-MAC slot time were set to 200ms to encompass 130ms for the GlacsWeb packet and 70ms to settle contention for transmission of the data. The sleep time for SMAC was set to 600ms. Finally the active time window (TA) for T-MAC was also set to 70ms making its minimum limit to be equal to the size of the contention window.

For a fair comparison between the three protocols the periodic listen and sleep was only activated during the same window of communication as GWMAC. In other words, probes sleep (and sense) for most of the time but only communicate during set predefined windows during which all three MAC protocols execute. The communication window interval was set to 24 hours and the simulations were conducted under 3 different data traffic loads. These are

- Low Traffic. Under this condition each node observes the environment every 12 hours such that it has 2 data packets to transmit during the communication window.
- Medium Traffic. Under this condition each node observes the environment every 3 hours such that it has 8 data packets to transmit during the communication window.
- High Traffic. Under this condition each node observes the environment every 1 hour such that it has 24 data packets to transmit during the communication window.

Finally, in order to obtain statistically significant results, average results of 100 simulations are reported in each of the experiments carried out.

Parameter	**Value**
Transmission per packet	0.0585J
Reception per packet	0.006435J
Idle transceiver per second	0.05J
Sense	0.015J
Sleep per second	105μJ
Communication Range	28m
Network Area	150m × 150m
Packet Size	64 bytes
Slot Duration	130ms

TABLE 4.2: Simulation Parameters

4.3.2 Network Lifetime

In this experiment, the four protocols were simulated in order to observe how the network lifetime is affected by each. Figure 4.7 shows the average network lifetime for each protocol as network size is varied under various traffic loads. The error bars are approximately the size of the plotted symbols. A gradual decline can be observed in lifetime as more number of nodes are introduced in the network. This result is contrary to what some might expect but it is important to understand that it is the network lifetime that is being referred to and not network coverage. Furthermore, as expected lifetime decreases as data traffic increases.

Although all four protocols perform similarly for a small network size of 5 nodes, GW-MAC outperforms the other protocols under all three traffic conditions. It is important to note, for each protocol, the rate at which lifetime decreases. L-MAC, S-MAC and T-MAC are all decentralised and distributed in nature. They involve a lot of idle listening for the purpose of carrier sense and this contributes to their energy depletion.

Furthermore, more nodes (in larger networks) competing for the same medium during the RTS phase results in an increase in the number of collisions. Consequently, some nodes are prevented from getting access to the medium but at the same time their energy reserves worn out. This is a typical scenario where nodes spend energy in competing for the medium instead of transmitting data itself. This characteristic explains the steeper gradient in their respective plots compared to GWMAC.

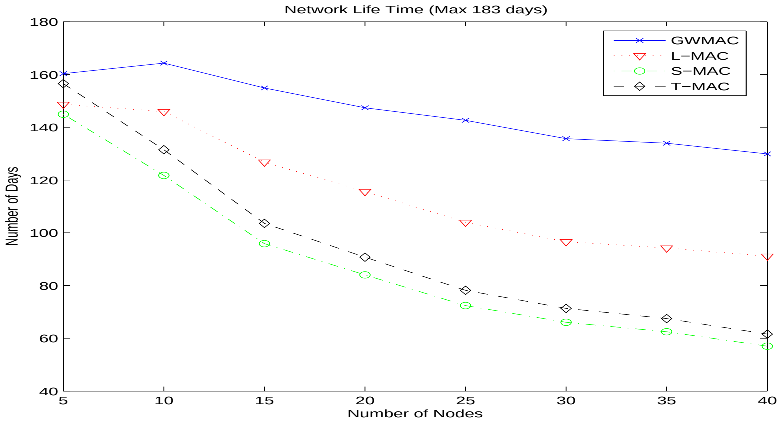
4.3.3 Data Collected Over Network Lifetime

In this experiment, the four protocols were simulated in order to observe how many data packets the *anchor* nodes could gather from the rest of the network over its lifetime. The results of these simulations are shown in Figure 4.8 under the three different traffic conditions. The error bars are approximately the size of the plotted symbols. The shorter network lifetime of the decentralised protocol ensues that less data is collected. T-MAC and S-MAC perform the worst because these protocols do not accommodate the presence of a slot controller like the receiving node in L-MAC or the transmitting node in GWMAC. This creates a *bottle neck* effect where intended receivers (relaying nodes) also compete for the medium to transmit their own data. The added functionality of the active time (TA) in T-MAC slightly improves performance over S-MAC where nodes tend to follow the same listen-sleep schedule. However this is not enough to better GWMAC or L-MAC.

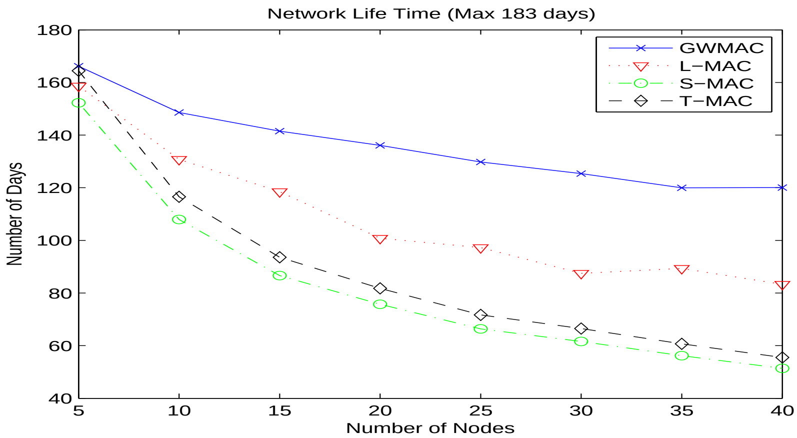
It can also be seen that the performance of the protocols are closer for small sized networks (the plots are bunched closer together). However, as the number of nodes in the network begin to increase, the gap between the plots becomes wider. Figure 4.8(a) and Figure 4.8(b) show GWMAC anchor nodes manage to collect almost twice as many data packets than their L-MAC counterparts and almost thrice as many as their T-MAC and S-MAC counter parts in large networks under low and medium traffic conditions respectively. However, the wide gap between S-MAC and T-MAC in larger networks tends to decrease under heavy traffic load. This is explained due to the bursty nature of their transmissions. In other words, once a node manages to reserve the medium for itself it is allowed to transmit all its data. Thus, it is more fruitful to compete for the medium under heavier traffic conditions than in low traffic because the reward (data transmission) is higher.

4.3.4 Energy Consumption and Time taken for Network Setup

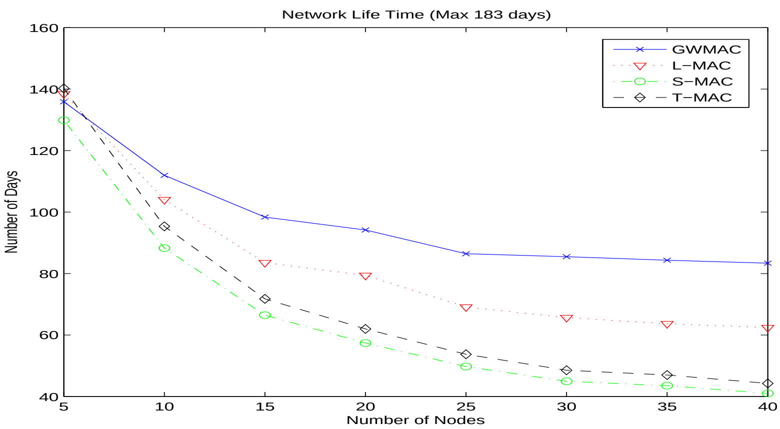
Network discovery and set up should be conducted on a regular basis, especially in a deployment field such as a glacier where the communication medium is frequently changing (ice to water and vice versa) and nodes are constantly on the move severely affecting network topology. In this part of the experiment, simulations were conducted



(a) Low Traffic

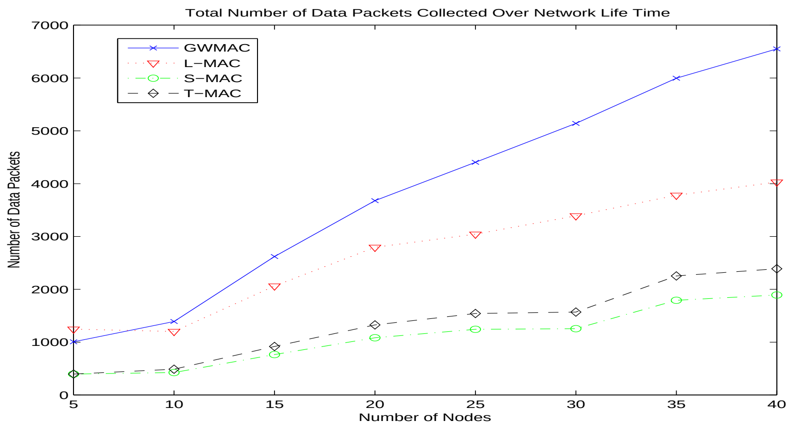


(b) Medium Traffic

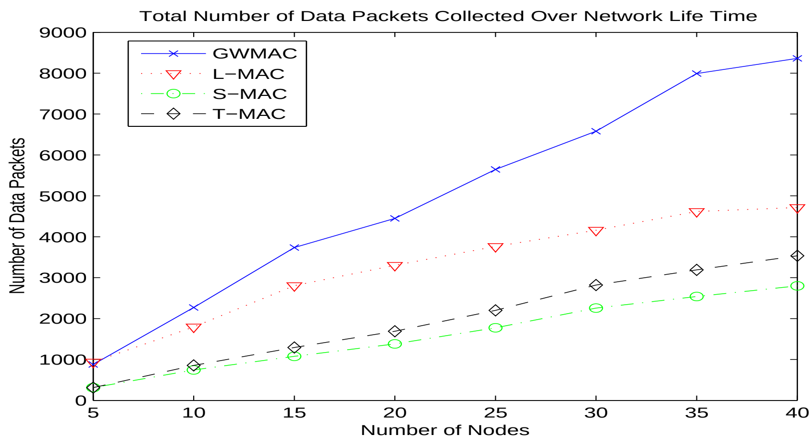


(c) High Traffic

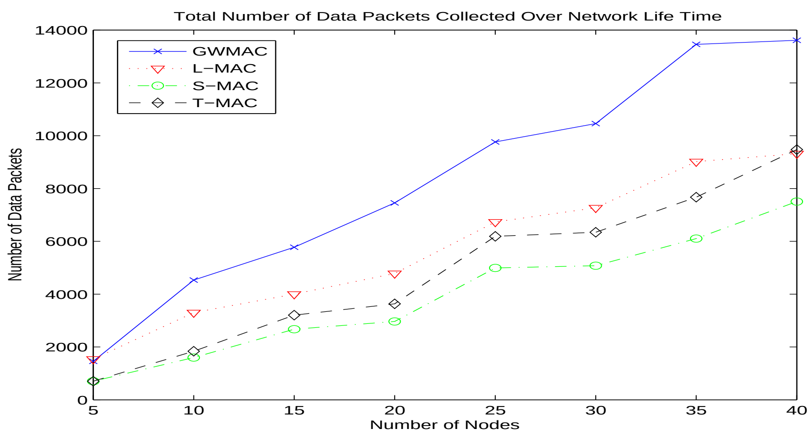
FIGURE 4.7: Network Lifetime plotted against number of nodes in the network.



(a) Low Traffic



(b) Medium Traffic



(c) High Traffic

FIGURE 4.8: Data Packets collected plotted against number of nodes in the network.

to analyse how much energy and time it takes for each protocol to organise the network. The results of these simulations can be seen in Figure 4.9. T-MAC has been deliberately left out in the graphs because it uses the same self-organising scheme as S-MAC and thus provides a very similar plot.

Figure 4.9(a) characterises GWMAC's linear relationship with the size of the network. In contrast both L-MAC and S-MAC plots illustrate that energy consumed increases exponentially as network size increases. Again, this is attributed to the decentralised nature of latter two. L-MAC nodes transmit their schedules to each other constantly until it is ensured that no two nodes within a 2-hop neighbourhood control the same time slot. S-MAC nodes continuously transmit each other's schedules until they are all synchronised to the same schedule. The time taken to complete these procedures increases with network size and this is verified in Section 4.9(b).

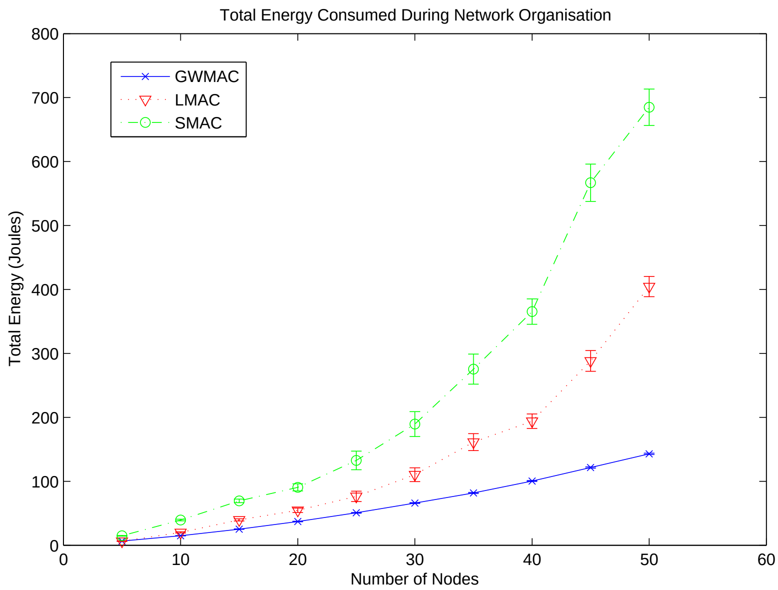
4.4 Summary

While many popular MAC protocols rely on idle listening (carrier sense multiple access), it is a waste of energy in sensor networks with low duty cycles. GlacsWeb is a typical example of such a sensor network. For this reason a better approach is to use TDMA-based protocols that allow the radio transceivers to switch off for the duration when no data must be exchanged. However, implementing such protocols present critical constraints that include

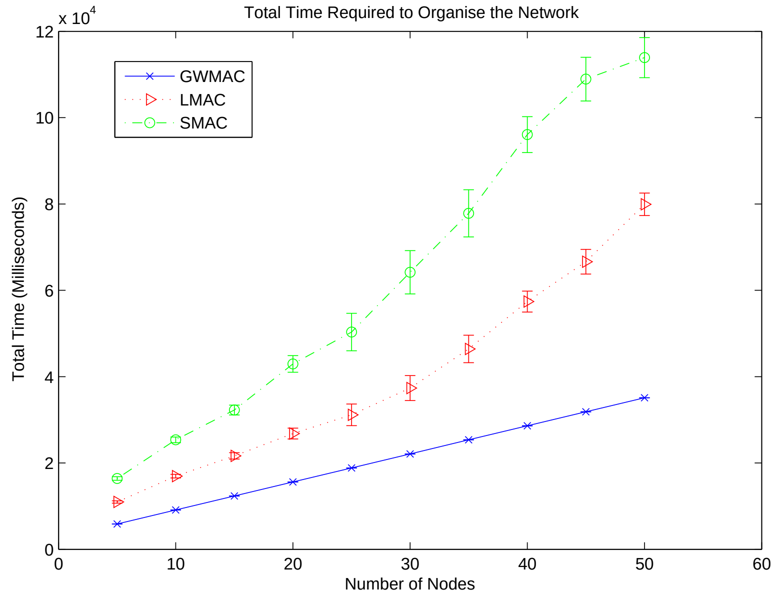
- Time Synchronisation.
- Scalability.

Although the protocol was tested in the lab, it could not be deployed and tested efficiently in the field due to logistical reasons. This is because the Brikdalsbreen glacier, in which the nodes were meant to be deployed, receded dramatically. Setting up the base station and the drilling equipment on surface ice became increasingly difficult (due to lack of much flat ice) and dangerous (due to development of deep crevasses). Deployment work had to be abandoned and the GlacsWeb team had to investigate an alternative glacier. Whilst, a new glacier (Skalafellsjokull in Iceland) has been tested and identified as a potentially new deployment site for the GlacsWeb team GWMAC will need to be deployed on the next generation of probes in order to understand the feasibility of its use in the actual hostile environment.

This chapter has presented GWMAC a TDMA-based MAC protocol that provides the functionality for node synchronisation and also caters for increasing network size so long as the base station is informed about new nodes entering the system. In addition



(a) Energy Consumption in Joules



(b) Time Taken in Milliseconds

FIGURE 4.9: Energy Consumption and Time Taken for Network Setup

GWMAC completely eliminates collisions and overhearing that providing huge energy savings compared to some of the other hybrid (schedule and contention based) MAC protocols. It has also verified this through evaluation in simulation and shown that GWMAC outperforms well known protocols such as S-MAC, T-MAC and L-MAC.

The evaluation chapter has shown many positive aspects of GWMAC, but it does not address some of the key aims of this thesis highlighted in Section 3.6. GWMAC does not factor into its model the concept adaptive sensing whereby nodes conserve energy. Furthermore, GWMAC does not address the issue of routing where nodes choose energy efficient routes to transmit their data to the base station. In the next chapter, the state of the art is extended by providing a network layer model that addresses these issues.

Chapter 5

USAC: Utility Based Sensing and Communication

This chapter considers how data transmission in a sensor network can be minimised whilst at the same time maximising the information gained about the environment sensed. Towards the end of Chapter 4 the aims of thesis achieved by the GWMAC model were highlighted. However, it was also identified that the GWMAC model lacked autonomy at node level where individual nodes are incapable of making independent decisions on sensing and transmission of data. This limitation is addressed in this chapter.

At present, sensing in GlacsWeb is carried out at a pre-determined constant rate which is blind to the actual variations in the environment. This decoupling results in unnecessary sampling because, given the same energy expenditure, the information gained by sensing a slowly varying environment is less than what could be gained in a more dynamic situation. Furthermore, as aforementioned in Section 2.8.1, several works have been carried that discuss novel adaptive sensing techniques. However, none of them place an emphasis on the importance of the sensed data in relation to the cost of communicating it. Statistical models alone are not good enough to minimise the task of sensing if, in the end, the data gathered does not add a high enough value to the research in comparison to the amount of energy compensated by the node in transmitting it. This is an important aspect that has been overlooked in literature till date and this chapter will attempt to address this issue by modelling GlacsWeb as a cooperative multi-agent system where all nodes (agents) work towards the pre-defined goal of maximising data collection. Specifically, this chapter aims to address GlacsWeb requirement 2 (decentralisation), GlacsWeb Requirement 4 (intelligent routing) and GlacsWeb Requirement 5 (intelligent sensing) that were discussed in Section 3.6.

Against this background, this chapter develops a *Utility-based Sensing And Communication protocol* (called USAC). This consists of a sensing and a routing protocol that uses the cost of transmission and the value of observed data as utility metrics in the agents' decision-making process. In developing the protocol, the state of the is advanced in the following ways :

- This chapter develops a novel, decentralised mechanism for *adaptive sampling*. In this, each agent locally adjusts its sensing rate depending on the valuation function that it uses to value the observed data. This valuation is based on the combination of Bayesian Linear Regression and the Kullback-Liebler Divergence (Kullback and Leibler, 1951) which gives it a sound information theoretic foundation.
- This chapter devises a new *multi-hop routing* protocol that finds the cheapest cost route from an agent to the centre. Here, the cost of a link from one agent to another is derived using the opportunity cost of the energy spent relaying the data (i.e. the value that a relay could have gained by using the energy in sensing instead of relaying).
- This evaluates the USAC protocol against four benchmark protocols; including a theoretically optimal protocol, a greedy protocol, GlacsWeb's original protocol and a protocol employing the adaptive sampling mechanism alone. The chapter demonstrates that against the latter three, USAC provides a significantly higher gain in information, whilst reducing power consumption. Furthermore, it compares favourably with the optimal protocol which is based on unrealistic assumptions such as the nodes having prior knowledge of their entire future observations and the best path to route data via over lifetime.

The chapter begins by discussing the sensing protocol in Section 5.1 by detailing the two main underlying decision making processes of forecasting data (Section 5.1.1) and assigning value to the sensed data (Section 5.1.2). It follows this up by explaining how these two decision making processes can be applied to GlacsWeb data in particular (Section 5.1.3). The chapter then presents a routing protocol where each agent node evaluates its own cost of transmission by incorporating the value of the data it sensed (Section 5.3). Having firmly established the theory behind USAC it then proceeds to demonstrate its superiority through a series of simulations in a wide range of scenarios in Section 5.4.

5.1 USAC's Sensing Protocol

The sensing protocol dictates how an agent should schedule its future sensing actions based on its current knowledge. If the protocol is adaptive, the agent only needs to decide *when* to next sense data. This is because, given the next sensed data, it may then change its future sensing times. Therfore, this section discusses a generic framework for this decision-making process within the agent, and explain how this can then be used in relation to the GlacsWeb data.

In this context, the *optimal* time at which the next sampling should occur can only be derived if the agent has knowledge about the future data. However, this requirement is contradictory since in the case that the agent knows the future data, it does not need to sense the environment. As a result, an agent can only find an optimal sampling rate based on its *forecast* of the future data. Then, upon observing previously predicted data, the agent gains information by reducing its uncertainty about its model of the environment.

Thus, in order to decide when to sample, a metric is required to determine how well a particular future sampling time is likely to do compared to another. The metric used in this case is derived from *information-theory* because this enables this research to have a principled means of obtaining the maximum information from the environment under certain constraints as imposed by the application scenario (e.g. power , bandwidth or other operational constraints). Such a principled approach is important because it can help provide a generic framework for other applications of sensor networks.

In more detail, the sampling protocol proposed can be described by Figure 5.1. The sensor first samples at some point and acquires the data x_n (`Sample`). This data is then used to update its existing model (`UpdateModel`) of the environment which, in turn, is employed to forecast data in the future. The magnitude of the change in the updated model then determines the value of x_n (`Evaluate` $Value(x_n)$). This value, along with the updated model, is then stored in the model history which is then passed on to the communication protocol discussed in section 5.3.

Furthermore, the sensing protocol needs to determine the next time the node should sample from the environment. In order to do so, it runs an iterative algorithm that compares the predicted value at future time steps[1] against a threshold value. This threshold value is important since without it, the sampling would occur at each available opportunity[2]. It is determined by the model history and the problem constraints, in conjunction with the domain knowledge (if available and relevant). If the predicted value of the data at a certain future time step is lower than the threshold value, then the algorithm prevents the sensor from sampling at that point and computes the value at the following time step. Note that as a result of basic information theory, the value of data at successive time steps increases since it is less predictable. Thus, the iterative algorithm will continue calculating the future predicted value until the time step where it surpasses the threshold. The sensor is then instructed to sample at this point in the future.

We can thus observe that the crucial decisions when using this framework are:

[1]The time step is determined by the maximum sampling frequency available to the sensor node.
[2]This is the information maximising sampling rate when no constraints are present.

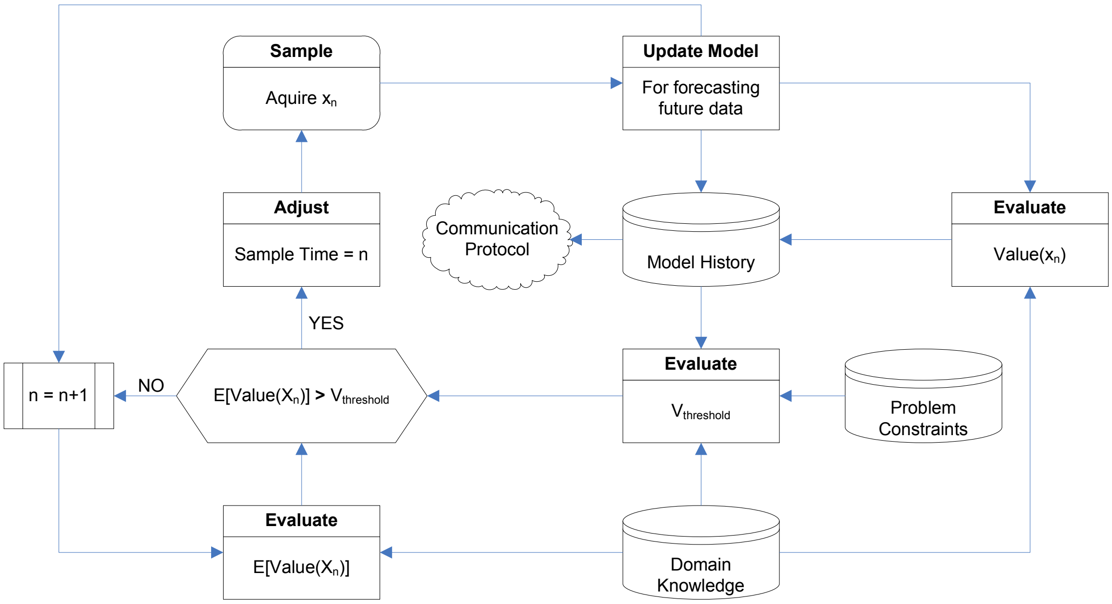


FIGURE 5.1: Decision making process for sensing within an agent. The domain knowledge part may not be present dependent on the application. When a sample is acquired, it is used to update the existing model for forecasting future data. The updated model is then in turn used to place a value on the acquired sample. $V_{threshold}$ is calculated using the history of the model, domain knowledge and other constraints. An iterative algorithm is then executed to compare the values of successive predicted samples in the future against the threshold in order to decide the next sampling time.

1. Choosing the framework that will be used to update the model which is then employed for predicting future data.

2. Deciding on the way of calculating the value of the next possible sampling of data, x_{n+1}.

3. Setting the threshold value. This is primarily caused by the constraints in the problem and is also dependent on the history of the sensing protocol, as well as the domain knowledge available.

As can be observed from figure 5.1, the forecasting method chosen will impact on the value of the data being measured by the sensor. This work focuses on a Bayesian model for updating the model held by the sensor. However, this model can be generalised to Gaussian processes as explained by (Williams, 1998). In GlacsWeb's particular case, this generalisation is not required since a fair amount of prior knowledge about the data from the glaciers is assumed (see Appendix D). Thus, the additional computational load of Gaussian Process Regression is not warranted in this case. The next section provides details of the framework of Bayesian regression which used in this chapter.

5.1.1 Forecasting Data

This section describes a generic forecasting method that also provides an agent with a means of rating the reliability of its forecast. This rating is important within the framework since the agent has to value the data it measures as well as predict the value of future data. It presents the Bayesian linear regression analysis framework, which can be extended in a straightforward manner to non-linear regression analysis by modifying the input vector as described in Box and Tiao (1992).

Within this context, a standard linear regression model with Gaussian noise can be represented as:

$$x_i = \boldsymbol{t}^{\mathrm{T}}\boldsymbol{w} + \epsilon \tag{5.1}$$

where x_i is an observed value, $\boldsymbol{t} = \{t_1, \ldots, t_j, \ldots, t_M\}$ is an input vector consisting of M variables (e.g. time and location at which readings are taken), $\boldsymbol{w}$ represents the weights assigned to each input variable t_j within the input vector and ϵ is additive noise drawn from an independent and identically distributed Gaussian distribution with zero mean and variance σ^2:

$$\epsilon \sim \mathcal{N}(0, \sigma^2) \tag{5.2}$$

Suppose, we now have N readings from a single sensor, whereby $\boldsymbol{x}$ denotes the vector of the N observations (i.e. $\boldsymbol{x} = \{x_1, x_2, \ldots, x_i, \ldots, x_N\}$) and T denotes the corresponding input matrix (i.e. $T = \{\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_N\}$). The objective within linear regression is to find a homogeneous real valued function, $g(\boldsymbol{t}) = \boldsymbol{t}^{\mathrm{T}}\boldsymbol{w}$, that best interpolates the training set $S = \{(\boldsymbol{t}_1, x_1), (\boldsymbol{t}_2, x_2), \ldots, (\boldsymbol{t}_N, x_N)\}$.

Now, there are a number of error functions (based on various norms such as the Manhattan or Mahanalobis distance) that could be used in order to characterise this "best" interpolation. The one used in standard linear regression is that of root mean square error. In this case, the objective is to to choose the parameters $\boldsymbol{w}$ so as to minimise the Euclidean distance between the projected data and the measured data. That is, choose $\boldsymbol{w}^*$ such that:

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \frac{1}{n} \sum_{i=1}^{n} \left(x_i - \boldsymbol{t}_i^{\mathrm{T}}\boldsymbol{w}\right)^2 \tag{5.3}$$

By applying the first-order condition on equation 5.3, it can be calculated that[3] :

$$\boldsymbol{w}^* = (T^{\mathrm{T}}T)^{-1}T^{\mathrm{T}}\boldsymbol{x} \tag{5.4}$$

See Appendix B for proof of this. The above equation thus provides us with a way to find the most likely fit of the data (the best linear unbiased estimate). However, it does not inform us about the likeliness of this fit (i.e. it does not quantitatively tell us how well this model explains the data as opposed to other models). This implies that we cannot then easily provide a confidence bound on predictions based on this estimate. In order to obtain such a quantitative assessment of our predictions, we shall turn to Bayes' Theorem. In essence, Bayes's Rule assigns a probability that an event E_1 occurs, given we have observed an event E_2, as:

$$P(E_1 \mid E_2) = \frac{P(E_2 \mid E_1)P(E_1)}{P(E_2)} \tag{5.5}$$

In GlacsWeb's case, it is desirable to predict the sensor's next measurement (the E_1) based on all previous observations (the E_2). More mathematically, we want to obtain the distribution $p(x_{n+1} \mid \boldsymbol{t}_{n+1}, T, \boldsymbol{x})$. This can be achieved by first finding the probability distribution for all linear models and then averaging out over these linear models:

$$p(x_{n+1} \mid \boldsymbol{t}_{n+1}, T, \boldsymbol{x}) = \int p(x_{n+1} \mid \boldsymbol{t}_{n+1}, \boldsymbol{w})p(\boldsymbol{w} \mid T, \boldsymbol{x})d\boldsymbol{w} \tag{5.6}$$

We thus now need to find $p(\boldsymbol{w} \mid T, \boldsymbol{x})$. That is, the probability distribution of the different linear models which can explain the data. In order to do so, we again apply Bayes' Rule. Assuming that the prior distribution $p(\boldsymbol{w}) \sim \mathcal{N}(0, \Sigma_p)$ and marginalising out $p(\boldsymbol{x} \mid T)$, the following can be derived (the proof of this is provided in Appendix C):

$$\begin{aligned} p(\boldsymbol{w} \mid T, \boldsymbol{x}) &= \frac{p(\boldsymbol{x} \mid T, \boldsymbol{w})p(\boldsymbol{w})}{p(\boldsymbol{x} \mid T)} \\ &\propto p(\boldsymbol{x} \mid T, \boldsymbol{w})p(\boldsymbol{w}) \\ &\sim \mathcal{N}(\overline{\boldsymbol{w}}, A^{-1}) \end{aligned} \tag{5.7}$$

[3] A number of methods exist in order to reduce the computation load of finding the optimal weights. However, this study does not consider such techniques since the focus is on placing a value on the sensed data, rather than optimising the computations when calculating such a value.

where $A = \sigma^{-2}TT^{\mathsf{T}} + \Sigma_p^{-1}$ and $\overline{\boldsymbol{w}} = \sigma^{-2}A^{-1}T\boldsymbol{x}$. Combining the result of equation 5.6 with that of equation 5.7, we obtain the following:

$$
\begin{aligned}
p(x_{n+1} \mid \boldsymbol{t}_{n+1}, T, \boldsymbol{x}) &= \int \boldsymbol{t}_{n+1}^{\mathsf{T}} \boldsymbol{w} p(\boldsymbol{w} \mid T, \boldsymbol{x}) d\boldsymbol{w} \\
&= \mathcal{N}(\frac{1}{\sigma^2} \boldsymbol{t}_{n+1}^{\mathsf{T}} A^{-1} T\boldsymbol{x}, \boldsymbol{t}_{n+1}^{\mathsf{T}} A^{-1} \boldsymbol{t}_{n+1})
\end{aligned} \tag{5.8}
$$

Another useful term is the marginal likelihood (or evidence), $p(\boldsymbol{x}|T)$ which is given by:

$$
p(\boldsymbol{x}|T) = \int p(\boldsymbol{x}|T, \boldsymbol{w}) p(\boldsymbol{w}) d\boldsymbol{w} \tag{5.9}
$$

For the prior used in this chapter, it can be shown that the log of the above equation reduces to (Rasmussen and Williams, 2005):

$$
logp(\boldsymbol{x}|T) = -\frac{1}{2}\boldsymbol{x}^{\mathsf{T}}(K + \sigma^2 I)^{-1}\boldsymbol{x} - \frac{1}{2} log \mid K + \sigma^2 I \mid - \frac{n}{2} log 2\pi
$$

where $K = T^{\mathsf{T}}\Sigma_p T$.

The Bayesian linear regression model discussed so far works under the assumption of a linear model (i.e. the observed data $\boldsymbol{x}$ is linearly related to the input T). This model can however be readily extended to a non-linear regression model by projecting the input vector $\boldsymbol{t}$ onto higher dimensions (called the feature space) to give rise to a new input vector $\widehat{\boldsymbol{t}} = \phi(t)$. This gives rise to what are commonly known as kernels or basis functions. Then, the model is given by:

$$
x_i = \phi(\boldsymbol{t})^{\mathsf{T}}\boldsymbol{w} + \epsilon \tag{5.10}
$$

in contrast to the linear regression model given by equation 5.1. The results derived so far are equivalently applicable to this new model with the only difference being that $\boldsymbol{x}$ is replaced by $\phi(\boldsymbol{t})$. So, for example, the data predicted at $\boldsymbol{t}_{n+1}$ is now given by:

$$
\begin{aligned}
p(x_{n+1} \mid \boldsymbol{t}_{n+1}, T, \boldsymbol{x}) &= \int \phi(\boldsymbol{t}_{n+1})^{\mathsf{T}} \boldsymbol{w} p(\boldsymbol{w} \mid T, \boldsymbol{x}) d\boldsymbol{w} \\
&= \mathcal{N}(\frac{1}{\sigma^2} \phi(\boldsymbol{t}_{n+1})^{\mathsf{T}} A^{-1} \phi(T)\boldsymbol{x}, \phi(\boldsymbol{t}_{n+1})^{\mathsf{T}} A^{-1} \phi(\boldsymbol{t}_{n+1}))
\end{aligned} \tag{5.11}
$$

An example of such a kernel regression is illustrated in figure 5.2 where the decrease in variance of the model as the number of points sampled increases is evident.

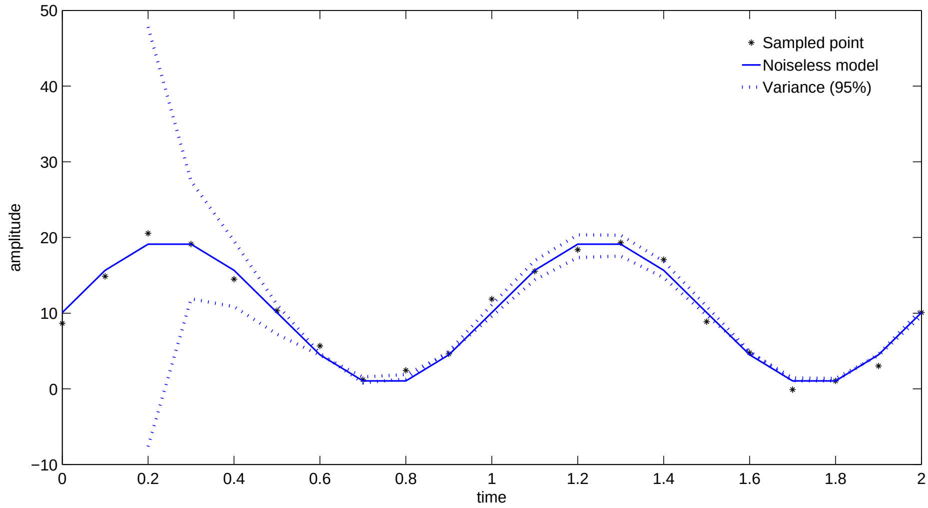


FIGURE 5.2: Example of Bayesian Kernel regression with a simple sinusoidal kernel

Given this background, the Bayesian kernel regression model can be viewed as an appropriate method of forecasting data when we are aware of some prior knowledge about the data (for example, we believe that it is a piecewise linear function or a sum of periodic functions). This knowledge is used in order to construct the kernels which are then used for regression. However, if such knowledge is not present, then we effectively need to produce a set of kernels on which to perform the regression. In order to deal with this, we could turn to Gaussian processes which provides us with a principled way of generalising the Bayesian linear regression model so as to generate sets of (potentially infinite) kernel functions given more meta-level descriptions of the input models (e.g. a squared exponential kernel is used to model the fact that observations from geographically near sensors tend to be highly correlated). More details about the Gaussian process regression are provided in Williams (1998).

To summarise, this section has provided an overview of how to update data models in order to forecast future data. The next section explains how to use these data models in order to value the data that is observed.

5.1.2 Valuing Data

This section addresses the problem of valuing data once it has been measured. An information-theoretic approach to solving this problem is to evaluate the amount of information that an observation provides. Now, following the Bayesian approach, a new observation leads to an update of the original belief about the probability distribution of the data model. Thus, the information gain of a particular observation is intuitively the difference between the prior and posterior probabilities. This can be gauged by the

Kullback-Leibler (KL) divergence measure Kullback and Leibler (1951) (a.k.a. information gain or relative entropy) which quantifies the difference between two probability distributions (f_1 and f_2) as:

$$\delta_{KL}(f_1, f_2) = \int_{-\infty}^{+\infty} f_1(y) \log \frac{f_1(y)}{f_2(y)} \tag{5.12}$$

Using this measure, we can then assess the impact of an observation x_n on the prediction of the next data point x_{n+1} by means of equation 5.8 for the linear Bayesian regression model. Specifically, let $\boldsymbol{x}_{n-1}$ denote the set of observations $\{x_1, x_2, \ldots, x_{n-1}\}$ and T_{n-1} be the corresponding input matrix. Then, before obtaining observation x_n, the prior for estimating x_{n+1} is:

$$p(x_{n+1} \mid \boldsymbol{t}_{n+1}, T_{n-1}, \boldsymbol{x}_{n-1}) = \mathcal{N}(\frac{1}{\sigma^2}\boldsymbol{t}_{n+1}^{\intercal} A_{n-1}^{-1} T_{n-1}\boldsymbol{x}_{n-1}, \boldsymbol{t}_{n+1}^{\intercal} A_{n-1}^{-1}\boldsymbol{t}_{n+1})$$

where $A_{n-1} = \sigma^{-2} T_{n-1} T_{n-1}^{\intercal} + \Sigma_p^{-1}$. Combining this with the posterior given by equation 5.8, provides us with the value of the information contained within x_n, given we have observed the vector $\boldsymbol{x}_{n-1}$ as:

$$V(x_n|\boldsymbol{x}_{n-1}) = \int_{-\infty}^{+\infty} p(x_{n+1} \mid \boldsymbol{t}_{n+1}, T_{n-1}, \boldsymbol{x}_{n-1}) \log \frac{p(x_{n+1} \mid \boldsymbol{t}_{n+1}, T_{n-1}, \boldsymbol{x}_{n-1})}{p(x_{n+1} \mid \boldsymbol{t}_{n+1}, T, \boldsymbol{x})} dx_{n+1} \tag{5.13}$$

Now it is known that if $f_1 = \mathcal{N}(\mu_0, \Sigma_0)$ and $f_2 = \mathcal{N}(\mu_1, \Sigma_1)$, then:

$$\delta_{KL}(f_1, f_2) = \frac{1}{2}\left(\log\left(\frac{\det \Sigma_1}{\det \Sigma_0}\right) + tr(\Sigma_1^{-1}\Sigma_0) + (\mu_1 - \mu_0)^{\intercal}\Sigma_1^{-1}(\mu_1 - \mu_0) - N\right) \tag{5.14}$$

Thus, it is straightforward to calculate equation 5.13 using equation 5.14, thereby yielding the value of the data. There are a few important and interesting points to note about this measure. Firstly, the KL divergence is always greater than or equal to zero, with $\delta_{KL}(f_1, f_2) = 0$ when $f_1 = f_2$. This implies that each new observation will always provide us with *some* new information. Secondly, the KL divergence does not satisfy the triangle inequality which implies that after making an observation, the agent may revise its estimate of the value of the previous observation Kullback (2001). Furthermore, observation of a data at a future time step is more highly valued if the agent does not carry out an observation now. This is a particularly useful property in the context of this work because it can help the agent decide *when* it will next make an observation in the future based on some threshold value. This is explained further in section 5.1.3

So far, this section has shown how to value an observation based on a purely information theoretic setting. This inherently assumes the equivalence of the information gained at various input points without any contextual preference assigned to the information. For example, an observation leading to an increased certainty that the temperature model is around 5 degrees is equally valued as an observation causing the same change in certainty for a temperature model of around −5 degrees. Whilst this is an appropriate model for GlacsWeb[4], this may not be the case for other sensor networks. For example, in a sensor network deployed for surveillance, reducing the uncertainty in a model resembling a humanoid shape might be of far greater value than reducing uncertainty in one resembling a cubic shape. Thus, the challenge that arises in these types of sensor network is how to modify the value of an observed data based on contextual information. This challenge can be addressed within the framework (specifically within the box labelled "domain knowledge" in figure 5.1) by using a Bayesian classifier Rasmussen and Williams (2005) (or any other principled classifier), which assigns a probability to the data belonging in a certain class[5]. The different classes are assigned weights according to their importance which can be derived from the context in which the sensor network is deployed. Then, using these weights and the probabilities derived from the classifier, an expected value of the data can be calculated for these sensor networks.

5.1.3 Application to GlacsWeb Data

We now discuss how to apply the general principles of the sensing protocol explained in the previous subsections to the specific GlacsWeb application. It should be noted upfront that the data within GlacsWeb can be characterised as piecewise-linear functions of time with added Gaussian noise since this impacts substantially on the forecast model used. Thus, the model of the data can be represented as:

$$p(x_n \mid t_n, w_1, \ldots, w_K, \sigma_1^2, \ldots \sigma_K^2) \sim \begin{cases} \mathcal{N}(t_n w_1, \sigma_1^2) & \text{if } 0 < n < \beta_1 \\ \vdots & \vdots \\ \mathcal{N}(t_n w_K, \sigma_K^2) & \text{if } \beta_{K-1} < n < \beta_K \end{cases} \tag{5.15}$$

where K is the number of different line segments comprising the overall model. Furthermore, the input vector in the case of GlacsWeb consists only of time points, thereby meaning that T consists only of the time vector $\boldsymbol{t} = \{t_1, \ldots, t_n\}$. From equation 5.15, it can therefore be observed that there are three major aspects of the data which are unknown:

1. The model parameters (i.e. $w_1, \ldots, w_K$).

[4] This has been confirmed by glaciologists working on the GlacsWeb project.

[5] An in-depth explanation is outside the scope of this study and will be developed in future work.

2. The point at which a phase change occurs (i.e. $\{\beta_1, \beta_2, \ldots, \beta_K\}$).

3. The level of noise in the environment (i.e. $\sigma_1^2, \ldots, \sigma_K^2$).

From the model above, observe that each segment of data, $\{x_{\beta_{k-1}}, \ldots, x_{\beta_k}\}$, poses exactly the same problem as a normal linear regression problem, with the model of the data changing at each breakpoint. Furthermore, a sensor only needs to consider whether the current observation will cause the current model to be refined or trigger the start of a new model. Hence, the explanation of the sensing protocol focuses on how to regress two linear models around one phase change. Then, as new data comes in, the sensor needs to decide whether it should refine its existing model of the data or whether it should switch model.

In more detail, as the sensor obtains data, it needs to find out whether a phase change has occurred and the point at which it has occurred. Let $\boldsymbol{x}_i^j = \{x_i, \ldots, x_j\}$ and $\boldsymbol{t}_i^j = \{t_i, \ldots, t_j\}$ where $j > i$. Then, the probability that a phase change happened at time n can be calculated as:

$$p(\beta_k = n \mid S) = \frac{p(\beta_k = n)p(\boldsymbol{x}_1^{n-1} \mid \boldsymbol{t}_1^{n-1})p(\boldsymbol{x}_n^N \mid \boldsymbol{t}_n^N)}{\sum_{\beta_k=3}^{N-1} p(\beta_k = n)p(\boldsymbol{x}_1^{n-1} \mid \boldsymbol{t}_1^{n-1})p(\boldsymbol{x}_n^N \mid \boldsymbol{t}_n^N)} \tag{5.16}$$

where $p(\beta_k = n)$ is the prior probability that the breakpoint occurs at n. We only concentrate on n being between 3 and $N-1$ since it is meaningless to consider less than two data points in each model. Furthermore, since we have no prior information about where the breakpoint occurs, a flat prior is used (i.e. $p(\beta_k = n) = U[3, N-1]$). Note that the normalising constant, $P(S|\exists\ \beta_k)$ is the probability of explaining the data set given a phase change:

$$P(S|\exists\ \beta_k) = \sum_{\beta_k=3}^{N-1} p(\beta_k = n)p(\boldsymbol{x}_1^{n-1} \mid \boldsymbol{t}_1^{n-1})p(\boldsymbol{x}_n^N \mid \boldsymbol{t}_n^N) \tag{5.17}$$

Now in order to estimate whether a phase change has actually occurred, we divide equation 5.17 by equation 5.9 (which can be interpreted as the probability of explaining the data set without any phase changes).

Having thus updated its data model accordingly, the sensor then needs to calculate the value of the data it has just sensed. This can be done using the KL divergence method outlined previously (see equations 5.13 and 5.14). Notice that the KL method automatically places a very high value on a data which signals a phase change (in equation 5.13 the denominator becomes very small if the new data does not conform to an existing model).

Finally, based on the data it has observed so far, the sensor calculates the next sampling time. In order to do so, it needs to calculate $V_{threshold}$ first. This is derived from the average of all data values (except the first two) in the current model of the data and is given by:

$$V_{threshold} = \frac{\sum_{n=3}^{i} V(x_n)}{i - n} \tag{5.18}$$

where i is the latest data sample in the current window. We do not incorporate the first two samples in this calculation because their extremely high value (since they indicate the start of new model) could set the threshold unreasonably high and thereby force the node to sample too far in the future. This can be detrimental since the sensor may miss out on sampling important data.

Having explained the sensing protocol we now consider how feasible it is to carry out the necessary computations on the GlacsWeb nodes in the next section.

5.2 Computational Feasibility

This section addresses the energy consumption and the time it takes to carry out the computations on the nodes. A significant amount of computation load is assigned to each sensor node in section 5.1.1 and therefore it is important to measure the feasibility of doing so. Each GlacsWeb node is installed with a 1MHz, 8-bit PIC18F4320 microcontroller which is responsible for controlling the sensors and processing data.

The model used in the sensing protocol (equations 5.17, 5.13 and 5.14) performs calculations dominated by the basic building block involving matrix exponential, i.e. e^A where A is some $N \times N$ matrix. Now, this expression is calculated as follows:

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!}$$

The above series always converges and therefore the exponential of A is well-defined. However, in order to save computational time and energy we only evaluate the above expression from $k = 0$ to $k = 20$ which provides a very good estimate of the final value(s) upto 3 decimal places. If the matrix A were to be of a generous size of 20×20, it would take take approximately 193734 instruction cycles to evaluate the above expression. The PIC processor uses 4 clock cyles for every instruction cycle. Therefore, time t taken to compute the above expression would be

$$t = \frac{193734 \times 4}{1000000} \approx 775ms$$

Furthermore, the PIC consumes 110μ A across 2V and therefore the total energy $E_{compute}$ required to calculate this matrix exponential is given by

$$E_{compute} = 110\mu A \times 2V \times 775ms = 170.5\mu J$$

Whilst 775ms may seem a long time for many real time applications, latency is not deemed to be an important factor within GlacsWeb as long as nodes transmit their data to the base station once a day Martinez et al. (2004). Furthermore, energy saving is of prime importance in this application and 170.5μJ is an excellent compromise for the time it takes to carry out the heavy computations.

The study goes one step further to simulate the average time and energy needed to evaluate equations 5.17, 5.13, and 5.14 using a Microchip compiler and results suggested that the PIC micro-controller would require computation time betwen $15 - 20s$ whilst consuming energy upto $4.27mJ$. This is again acceptable for GlacsWeb due to the extremely small amount of power consumption.

Table 5.1 illustrates the different PIC micro-controller families and the trade-off between processor speed and current consumption. The PIC16 could perform the calculation with the least amount of power, but does take a significant amount of time. At the other extreme, the PIC32 required the least amount of time, but it does have a higher idle current. PIC18 micro-controller was used as a compromise between idle and calculation current to suit GlacsWeb's needs.

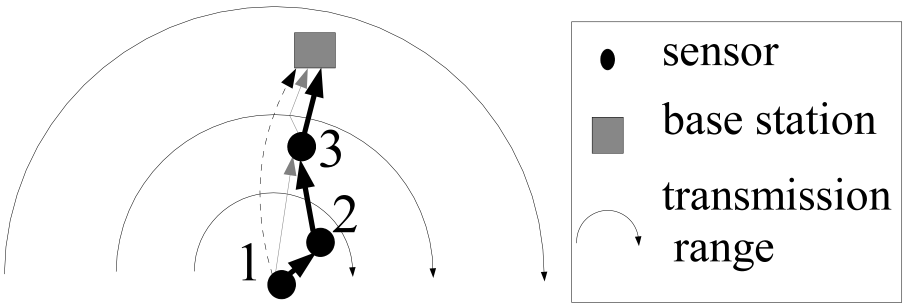
TABLE 5.1: PIC micro-controllers and their characteristics

PIC family	Width (bits)	Voltage	Current	Speed
PIC16	8	2	0.018mA	0.1MHz
PIC18	8	2	0.11mA	1MHz
PIC24	16	2	2.6mA	8MHz
PIC32	32	55mA		8MHz

Having now explained the salient feature of the sensing protocol including adjusting the sampling rate, calculating the value of the data and evaluating the computational feasibility in deriving it, we consider how this value can be used by the communication protocol to transmit data to the base station.

5.3 USAC's Routing Protocol

Once a sensor has collected data from the environment, it needs to transmit it towards the base station. In the initial version of the system this was done by direct transmission to the centre Padhy et al. (2005). However, as discussed in Chapter 3, this is inefficient since the power required to transmit data from one node to another is proportional to the square of the distance between the nodes (from basic radio transmission theory Bertoni (1999)). As a result, the total energy spent by transmitting data directly to the centre via a single hop is more than the energy spent when the data is relayed via successive intermediaries to the centre. In order to see this effect, consider the example shown in figure 5.3. Here, sensor 1 could transmit data to the base station (bs) via the following three routes: $1 \rightarrow 2 \rightarrow 3 \rightarrow bs$ (bold), $1 \rightarrow 3 \rightarrow bs$ (grey) and $1 \rightarrow bs$ (broken line). The total energy consumed for the transmission of one packet of data would then be 12 $(4+4+4)$, 20 $(16+4)$ and 36 respectively, thereby suggesting the use of route $1 \rightarrow 2 \rightarrow 3 \rightarrow bs$.



receiver / sender	2	3	base
1	4	16	36
2	X	4	16
3	4	X	4

FIGURE 5.3: Three possible routes via which sensor 1 can transmit its data to the base station. The concentric semi-circles show the range of sensor 1 with three power levels chosen such that the range grows linearly. The table shows the energy required for a sensor to transmit a packet directly to another.

However, an approach based solely on the transmission power is too naïve since it disregards the two following aspects:

1. **The opportunity cost of the energy used by each sensor.** If a sensor does not relay data, it could then use that energy in order to carry out additional sensing (which contributes towards the value of the network). Since each sensor is in a different local environment (due to the different placement of the sensors in the glacier), they derive different values by sensing the environment. Hence, it might be preferable for a sensor to transmit its data via a more energy-consuming

route if a lower energy-consuming route contains a sensor in a highly dynamic environment.

2. **The total power required to transmit along a particular route.** The transmission of data also requires the receiving node to be in a listening mode (i.e. the agent needs to switch on its antenna for receiving data which also consumes power). Thus the route $1 \rightarrow 2 \rightarrow 3 \rightarrow bs$ requires both sensor 2 and 3 to additionally spend energy receiving the data.

These two problems are tackled by developing a utility-based communication protocol. This protocol is based on the value of the data to be routed to the base station (which is derived according to methods detailed in section 5.1.2) and the cost of transmitting the data. The next section details how to calculate the cost of communication, before going onto the algorithm used for the communication protocol in section 5.3.2.

5.3.1 The Cost of Communication

The network is modelled as a multi-agent system consisting of a number of agents, $\mathcal{I} = \{1, \ldots, n\}$, that each have K different discrete power levels, $\{pt_i^1, \ldots, pt_i^K\}$, (with $pt_i^{k+1} > pt_i^k$) at which they can transmit. At each level, there is a set of neighbours $n_i(pt_i^k) \subseteq \mathcal{I}$ to which agent i can transmit data. Due to the nature of radio transmission, $n_i(pt_i^k) \subseteq n_i(pt_i^{k+1})$.

Thus, the direct communication of data from any agent i to another agent j, where $j \in n_i(pt_i^k)$ consumes a certain amount of energy Et_i^j which is given by:

$$Et_i^j(data) = pt_i^{k*} \times t_i^j(data)$$

where pt_i^{k*} is the lowest power level at which $j \in n_i(pt_i^k)$ and $t_i^j(data)$ is the amount of time a data packet takes to transmit. Now, in this scenario, the size of each sensed data packet and the bandwidth available to each agent is the same, so $t_i^j(data)$ is constant for all agents and sensed data packets. Therefore, by slight abuse of notation, we shall hereafter refer to $Et_i^j(data)$ as Et_i^j.

The cost of communication of an agent i to another agent j is then the opportunity cost of that decision. In this case, there are two particular scenarios to consider when communicating data. If, on one hand, an agent is originating the data, then its cost of communication is given by:

$$c_i^j(originate) = \frac{Et_i^j}{Et_i^j + E_i^{sense}} \times v_i^{sense}(t_n) \tag{5.19}$$

where E_i^{sense} is the energy spent by i in sensing new data and $v_i^{sense}(t_n)$ is the value of the new data. On the other hand, if an agent is relaying data, then its cost of transmission is given by:

$$c_i^j(relay) = \frac{Et_i^j + E_i^{receive}}{Et_i^j + E_i^{sense}} \times v_i^{sense}(t_n) \tag{5.20}$$

where $E_i^{receive}$ is the energy spent by the agent receiving the data which it then relays.

Now, since it is not possible to assign $v_i^{sense}(t_n)$ before actually carrying out the observation, we need to estimate it. Due to the nature of the data (where sudden changes are possible) we estimate $v_i^{sense}(t_n)$ using a moving average with window size w. Thus, at time t_n, the estimated value of the data is given by:

$$\overline{v_i}^{sense}(t_n) = \frac{1}{\min(n, w)} \sum_{i=\max(n-w,0)}^{n-1} v_i^{sense}(t_i)$$

Such a forecasting method is chosen because it evens out the changes in value that random noise can introduce, whilst at the same time updating the value of the data fairly quickly as time progresses. However, it should be noted that this forecasting method (or for that matter any forecasting method) cannot guarantee to correctly predict the value of the data all the time. Also, the moving average only starts once the number of samples collected by the sensor $> w$. Up to that point, the estimated value is just an average.

Having thus explained how the cost of communication is calculated, we now detail the algorithm followed by each agent when communicating data.

5.3.2 The Communication Algorithm

The algorithm used for the communication protocol is given in figure 5.4. It consists of four main steps, namely:

1. **Initialisation.** In this phase, the network topology is discovered and each agent is made aware of the power level it must transmit at in order to reach each of its neighbours. This phase is run each time a change in network topology is anticipated (either due to deployment of new nodes in the glacier, or because nodes have moved with the ice).

Algorithm 1.

1. **Initialisation**.
   Run a network discovery protocol that establishes $n_i(pt_i^k)$ $\forall k, i$. Go to step 2.

2. **Update transmission power**.
   At predefined intervals of time, t_{update} $(> 1/f_{min})$, the centre broadcasts a message $msg_0 = \langle P_0^{trans}, P_0^{rec} \rangle$ at the highest power level pt_0^K, where P_0^{trans} is the power at which the centre has transmitted this message and P_0^{rec} is the minimum power at which the centre can receive data. An agent can then calculate the minimum power required to transmit to the centre as:

   $$P_i^{min} = \frac{P_0^{rec} P_0^{trans}}{P_i^{rec}(msg_0)}$$

   assuming that the dissipation of power is symmetric between 0 and i. From P_i^{min}, an agent can then determine the minimum power level $pt_i^{k*}(0)$ required to transmit to the base station (since $0 \in n_i(pt_i^k)$ if $pt_i^k > P_i^{min}$). It can then determine $Et_i^0(data)$ and thus $c_i^j(originate)$.

3. **Update cost of transmission to base**.
   Let $I(k*) \in I$ be the set of agents that require the minimum power level pt_i^{k*} to transmit to base $(I(k*) = n_0(pt_0^{k*}) - n_0(pt_0^{k*-1})))$. Note that $pt_0^0 = \{0\}$. Agents $i \in I(1)$ can calculate their own cost of relaying data to the base station, $c_i^0(relay)$. Upon calculation, they broadcast the message $\langle c_i^0(relay) \rangle$ at power level pt_i^K.
   Then for $k* = 2$ to K do

   - Agents $i \in I(k*)$ calculate the cost of relaying data $c_i^j(relay)$ to all agents $j \in I(k* - 1)$.
   - They also update their cost of transmission, $c_i^0(originate)$, as $\min(c_i^l(originate) + c_l^0(relay))$ where $l \in \cup_{a=1}^{k*-1} I(k* - a)$
   - They then broadcast the message $\langle c_i^0(relay) \rangle$ at power level pt_i^K where $c_i^0(relay) = \min(c_i^l(relay) + c_l^0(relay))$.

4. **Transmit data.**
   Send the data packet through the lowest cost path if $V_i(data(t_s)) > c_i^0(originate)$. Update $c_i^j(originate), c_i^j(relay)$ from the value of newly sensed data.

5. **Repeat Step.**

   If time to update transmission power levels, **then** go to Step 2
   else if time to update relay and originate costs, **then** go to Step 3
   else sense data and go to step 4.

FIGURE 5.4: The routing algorithm.

2. **Update energy band of agent.** This step is responsible for dividing the agents into different power level groups with respect to the base. This segmentation is then used in the next step in order to update the cost of relaying data.

3. **Update cost of transmission to base.** This step is required so as to find the minimum cost route from each agent to the centre. In order to do so, agents in each power level group successively transmit the cost of their cheapest route to the centre.

4. **Transmit data**. Having found the cheapest cost of transmission to the base, the agent then decides whether or not to transmit its observed data.

Having detailed the communication aspect of USAC and how it intertwines with the sensing protocol by considering the value of sensed data, we proceed to next section where its performance is evaluated against some benchmark protocols in a simulation.

5.4 Model Evaluation

This section evaluates the performance of USAC through a series of experiments in a simulation. In particular, USAC is compared against four other alternative protocols (discussed below) in networks with varying topology, agent numbers and the degree of dynamism in the environment. USAC is also benchmarked against a theoretical optimal strategy. This strategy assumes prior knowledge of the complete observation environment of each node and then computes the best sampling points and communication route to the base station such that the networks achieves a theoretical maximum lifetime. This is clearly impossible in practice, but, nevertheless indicates how effective the strategy is in absolute terms.

5.4.1 Experimental Setup and Performance Metrics

The benchmark strategies represent the dominant approaches available in the literature to deal with power-efficient routing:

1. **Infrastructure Based.** This is the strategy originally employed in GlacsWeb. Each agent transmits to the base station in a single hop. If the agent realises that the base station is outside its transmission range, it simply fails to transmit the data. The plot for this strategy is labelled *DIRECT*.

2. **Forced Obligation.** In exploratory uses of sensor networks, a priori model best describing the sensor values being monitored is not always known. Since data alone is the *ground truth*, physical scientists want to collect all data. Therefore in

the simulations employ a strategy in which each agent is obliged to communicate all its sensed data, even if the cost of transmission is higher than its worth. The plot for this strategy is labelled *FORCED*.

3. **MintRoute** This strategy is based on a metric to capture end-to-end capability of forwardness proposed in Woo et al. (2003). It is defined as the expected number of transmissions (including re-transmissions) for a successful end-to-end data forwarding. It defines link quality as

$$etx(l) = \frac{1}{p_f(l) \times p_r(l)}$$

where $p_f(l)$ is the forward probability of link l and $p_r(l)$ is its reverse probability. $p_f(l)$ is calculated as the ratio of successfully transmitted packets to the total number of packets transmitted over l. $p_r(l)$ is the calculated as $p_f(\bar{l})$ where $\bar{l}$ is the reverse link of l. The route metric of a n-hop path p is then calculated as $ETX(p) = \sum_{i=1}^{n} etx(l_i)$.

4. **Optimal Sensing and Communication.** This strategy represents an optimum solution. It is executed by recasting the network as a centralized global optimization problem and using simulated annealing to gather information about the entire network's environmental data. It then calculates an optimal communication path between an agent and the base station such that the lifetime of the network is maximised. However, because it assumes knowledge about each agent's data and how their opportunity costs will change over time, it is not itself a viable solution to the problem.

It is obvious that adaptive sampling will result in significant energy savings and, therefore, to avoid giving USAC an unfair advantage, all other protocols in the simulation, with the exception of the OPTIMAL algorithm, were endowed with the same adaptive sampling mechanism. GWMAC protocol developed in Chapter 4 is used to eliminate the lower level problems of overhearing.

In terms of measuring performance, the following definitions are adopted:

Definition 5.1. Efficiency is the total value of the data received by the base station divided by the total energy consumed by the entire network in collecting it.

Definition 5.2. Network Lifetime is defined as the time taken for 50% of the initial number of agent nodes in the network to die.

A formal definition of network lifetime is not straightforward and depends on application scenario in which the network is used. The simulations suggested that failure of approximately 50% of the nodes (almost all of them being intermediate i.e. closer to the base station) led to little or no connectivity with the network. Hence, the above definition of network liftime refers to the capability of the network to provide the services it was designed for (i.e gather data from the nodes).

In the simulation, each agent is allowed to take one of the following actions in a single time period: *sense*, $idle - listen$ (where an agent enables its antenna so that it is ready to receive data), *transmit* a single packet, *receive* a single packet and *sleep*. With the exception of *transmit*, all actions have a set energy consumption value affixed to them. The radio propagation model is assumed to be symmetric and deterministic in which the energy to reliably transmit over a distance d is proportional to the square of this distance ($energy \propto d^2$). Each agent is provided with five different levels of transmission power to communicate with other agents at five different transmission ranges. The energy consumption of the *transmit* action is dependent on this variable transmission power of the agent transmitter using the square law. We decided to ignore the *processing* action of the agent due to its near negligible energy consumption. Specifically, table 4.2 in Chapter 4 shows the typical energy consumption of each action based on the values obtained from the fielded system. Furthermore, the initial energy capacity of the agent was set to 2000J and the confidence level within the sensing protocol was set to 10% (again based on the experience with the fielded system).

For the purpose of simulation, the transmission link quality between any two nodes was set to 0.8. This value was based on the real experience of deploying the system in the glacier where it was discovered that the average rate of successful packet transmission drops down from almost 100% in the winter months to 80% in the summer months when the en-glacial water bodies start to attenuate the radio signals. See Figure 3.6.

Finally, for statistical significance, the simulations report average results and standard deviations of 100 simulations in each of the experiments carried out. The data used in the experiments is derived from segments of the data collected[6] by the fielded probes over the last three years. The graphs show the standard error of the mean, as well as the 95% confidence intervals. Thus, error bars in the plots are in in the form: y e, implying that we are 95% confident that the true mean (i.e. average) lies within the range of values: y -e to y +a. Since the simulations instances were conducted over a period of 6 months, the maximum lifetime of a network was capped at 183 days.

5.4.2 Network Topology (Static Nodes)

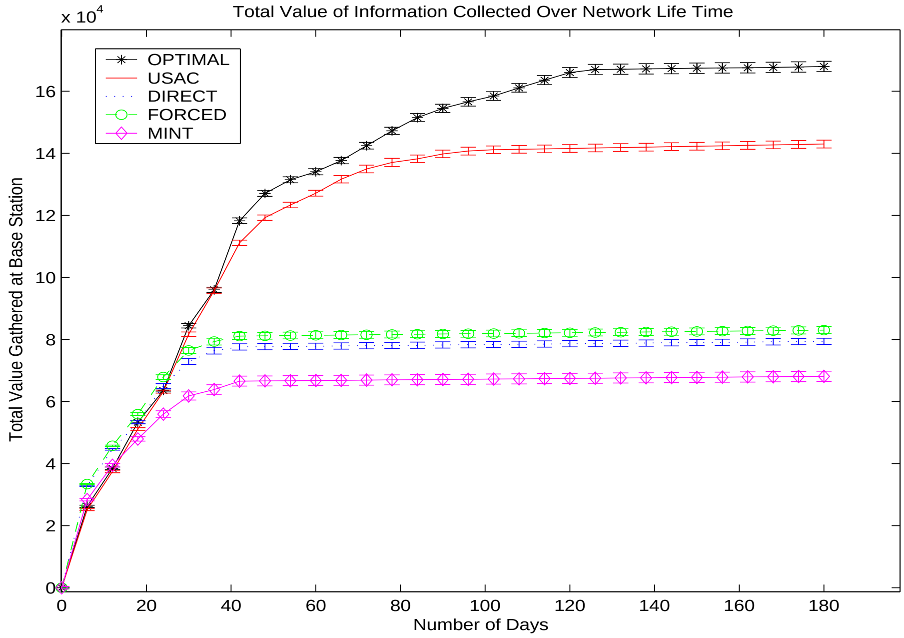
In this experiment, simulations were carried out for a fixed number of agents (10) randomly distributed around the centre. The sensed data model for each agent and the number of agents in the networks were fixed for each instance of the simulation. The purpose of this was to analyse how the protocols fared against each other on a daily basis in light of different network topologies. The results of these simulations are shown in

[6] http://leo.ecs.soton.ac.uk/GlacsWeb/plotter.php

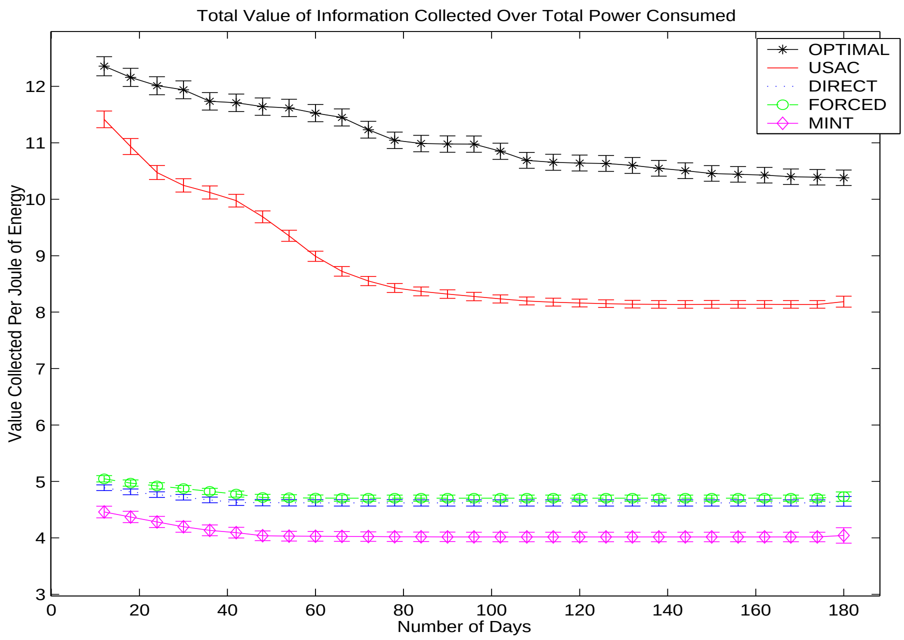
Figure 5.5. Both plots show the superiority of USAC over the other practical protocols. Specifically, sub-figure 5.5(a) shows how the total value of information collected at the base station increases and then stagnates for each protocol through the 6-month period. The point of stagnation (start in the flatness of the lines) indicates when the batteries of 50% of nodes are flat. The plot shows that initially the MINT, DIRECT and USAC base stations manage to collect the same value of information whilst the FORCED base station manages to accummulate higher information gain. This is expected as all samples sensed by the FORCED agents is transmitted as opposed to the the selective data transmitted in the other protocols. This approach is not the best in the long run as the intermediate agents in the network are obliged to receive and forward data from other agents and therefore drain their resouces quickly. MINT performs the worst in terms of network lifetime and information gain. This validates the theory that multi-hopping is not necessarily energy efficient in circumstances where direct transmission is a possibility. On the other extreme transmitting directly at the highest transmission power can also having mitigating effects as seen as in DIRECT. Therefore, a middle ground needs to sought which is provided in USAC. Sub-figure 5.5(b) verifies this by plotting the daily efficiency of each protocol. It can be seen that although USAC collects a lower value of information at the start, it is almost three time more efficient than the others. Towards the end of the 6-month period, USAC extracts an efficiency gain of 78% over DIRECT, 74% over FORCED and upto 100% over MINT. In addition USAC performs at 79% efficiency of the OPTIMAL protocol.

5.4.3 Dynamic Network Topology (Mobile Nodes)

The previous experiment was extended by introducing node mobility in the network in order to simulate the effect of glacial movement. Based on the glaciologists' advice and GlacsWeb's own fielded system experience, agent nodes were programmed to move randomly in one of three directions (left or right, but predominantly down the slope of the glacier) at every time step. Like the previous experiment, the sensed data model for each agent was fixed for each instance of the simulation. The purpose of this was to analyse how the protocols fared against constant change in topology during the network lifetime. The results of these simulations are shown in figure 5.6. Again, the plots tells us that USAC performs significantly better than MINT, DIRECT and FORCED although their performance degrades slightly in comparison to when agents are static. However, it can be noted from sub-figure 5.6(b) that in comparison to the OPTIMAL protocol, USAC still performs at 76% efficiency of the OPTIMAL protocol whilst the other protocols do not fare any better than in the static topology case. This demonstrates USAC's ability to better adapt in hostile conditions, such as that of a glacier, where topology is constantly changing and communication links are continuously breaking.

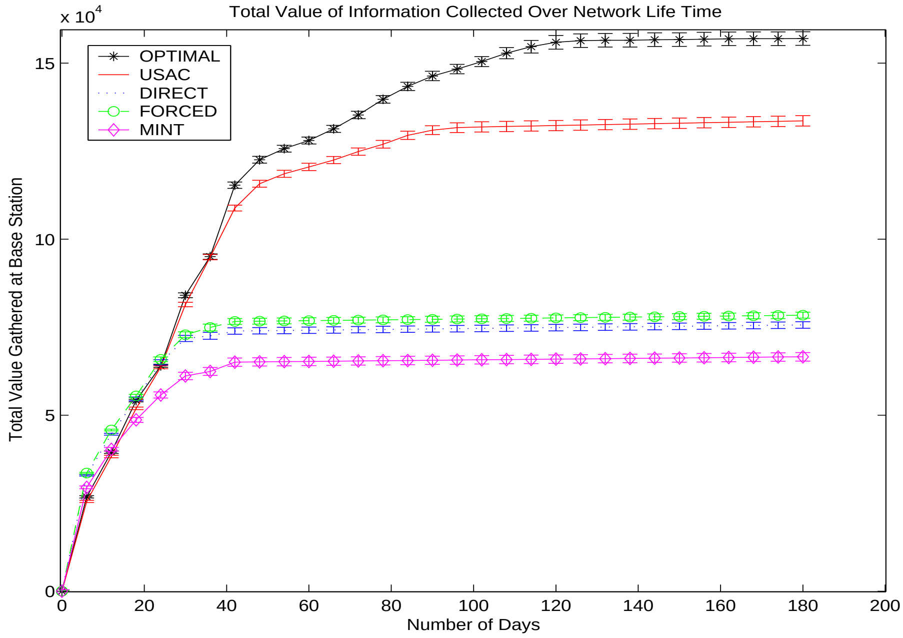


(a) Value Derived

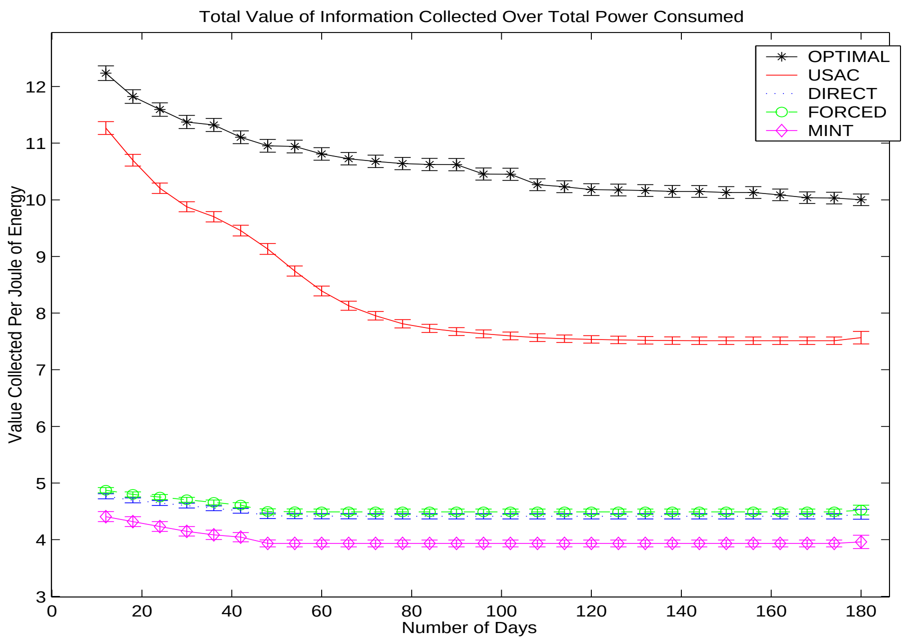


(b) Efficiency

FIGURE 5.5: The total value of data gathered and total value of data gathered per joule over a 6-month period plotted against time (Fixed Topology)



(a) Value Derived



(b) Efficiency

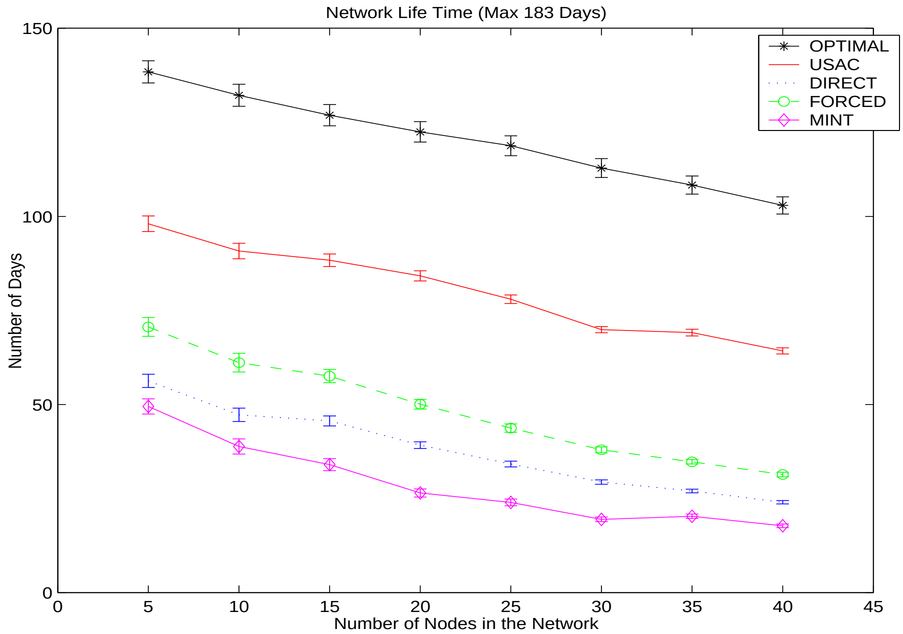
FIGURE 5.6: The total value of data gathered and total value of data gatherd per joule over a 6-month period plotted against time (Dynamic Topology)

5.4.4 Network Size

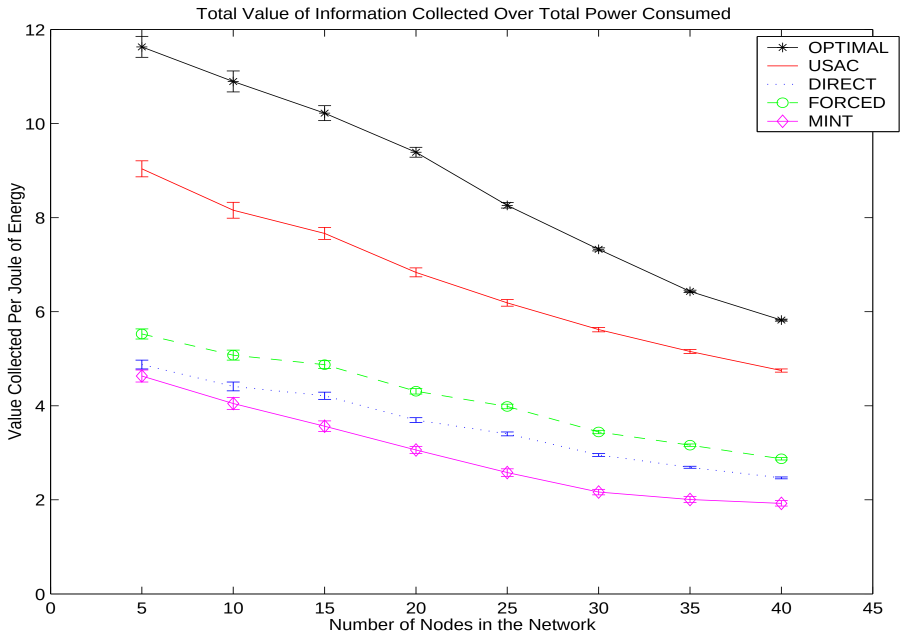
In this part of the experiment, the simulation varied the number of agents in the network from 5 to 40. The aim here is to explore how well USAC adapts to network scalability. Figure 5.7 illustrates the performance of all five protocols as network size is increased. Due to the variability of the observation environment, the value of data collected by the nodes and their efficiency changes every day. For this reason, the study was concerned with the average of the final total of these attributes at the end of each simulation to ensure an effective comparison. In more detail, sub-figure 5.7(a) shows how the network lifetime declines as more agents participate in the network. This is because the amount of data that a node might forward as a relayer increases with an increase in the number of nodes. This in turn leads to a faster depletion of energy reserves. Whilst, the graphs seem fairly linear, the gap between USAC and the remaining practical protocols is quite big. In particular, USAC is able to extend the network lifetime by almost 60% in comparison to FORCED, 100% in comparison DIRECT and by almost 200% in comparison to MINT. On average, the lifetime of a network employing USAC is approximately 65% that of one employing the OPTIMAL protocol. This is important because an extended lifetime means that more value of information can be gathered at the base station.

In this case, however, a much better comparison metric is the total value of information gathered at the base station for every joule of energy consumed in the network. Thus, sub-figure 5.7(b) illustrates very nicely how this metric is affected by the size of the network for each protocol. Specifically, the initial cheap cost of communication for MINT agents ensures that those close to the base station die quickly which makes communication more expensive for agents further away in the longer run (since isolated agents are only left with the option of transmitting directly). DIRECT agents show marginal improvement in their network lifetime over MINT agents. However, because their transmission is independent, agents further away from the base station suffer a similar fate to that of the isolated MINT agents. The FORCED agents choose the cheapest cost path to communicate their data. However, because they send all their sensed data (not all of which provides significant new information) it's value is not significant for the amount of energy expended in transmitting it.

In contrast, USAC agents manage to extract more worthwhile data over the network lifetime for the same amount of energy used by the other protocols. Their performance in obtaining a total value of information per Joule of energy consumption is rated at 70% of that of the OPTIMAL.

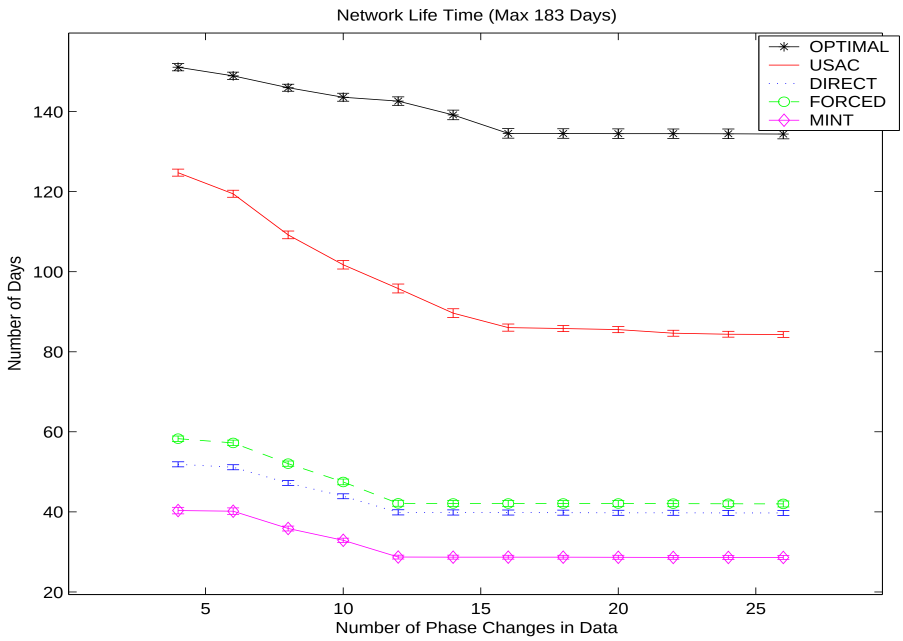


(a) Network Lifetime

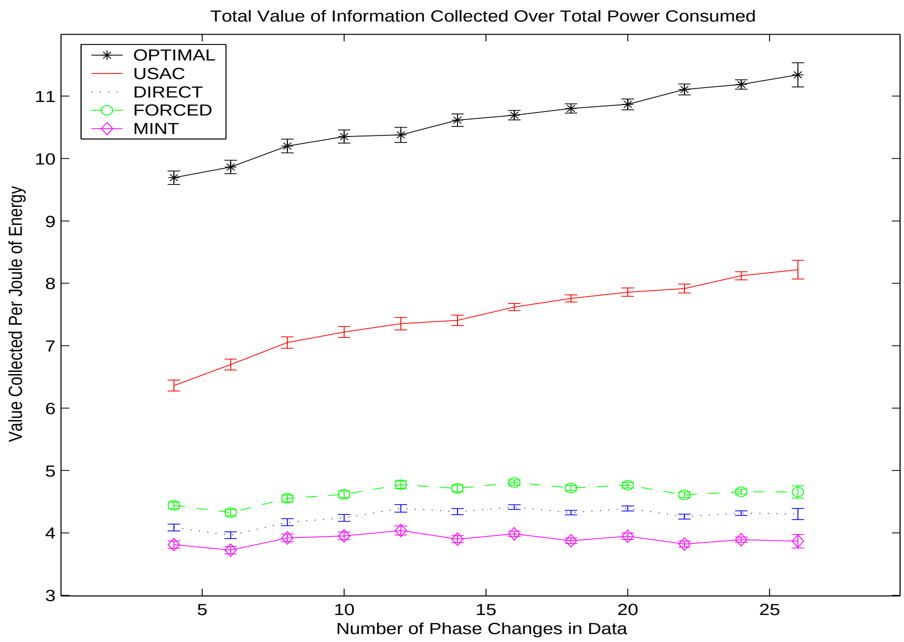


(b) Efficiency

FIGURE 5.7: Network Lifetime and Efficiency (at end of Network lifetime) plotted against number of agents in the network



(a) Network Lifetime



(b) Value Per Joule

FIGURE 5.8: Network Lifetime and Efficiency plotted against a measure of dynamism of the environment.

5.4.5 Dynamism of the Environment

In this experiment, simulations were carried out by varying the data model (observation environment) of the agent nodes whilst keeping the topology and size of the network constant. Here the degree of dynamism in the data model was defined to be the number of phase changes that occur in the piecewise linear data model of the environment used by the agent. The aim in undertaking this experiment was to evaluate how well the protocols reacted to the change in their observation environment.

In more detail, sub-figure 5.8(a) shows how network lifetime for each protocol is affected by increasing the number of phase changes from 4 through to 26. The initial trend of each graph indicates that the lifetime decreases rapidly with an increase in the dynamism. This can be attributed to the adaptive sampling mechansim, because more phase changes imply that the agents have to perform more sensing and this results in them acquiring larger amounts of data than usual that is valued much higher. In this context, more valuable data allows agents to exploit cheaper communication costs and indulge in more frequent transmission activity which consequently leads to quicker exhaustion of energy reserves. The sub-figure also shows that following the initial linear decrease in lifetime, there comes a point for each protocol at which its performance flattens out and no further decline takes place. This suggests the concept of a minimum network lifetime (a lower bound on the network lifetime) and can be seen as a period during which data collection from the network is guaranteed. Specifically, the experiment shows that USAC increases the minimum lifetime of a network by 112%, 97% and 183% over DIRECT, FORCED and MINT strategies respectively. The minimum lifetime of a USAC network is 63% of the minimum lifetime of the OPTIMUM network.

Sub-figure 5.8(b) shows the efficiency plot for the different protocols. As can be seen, an environment with twice as many phase changes, results in approximately twice as much data being transmitted, for approximately twice as much enery expenditure for the FORCED, DIRECT and MINT strategies. This explains the reason for their constant efficiency (flat graph). In a FORCED network, agent nodes transmit all data (both high and low valued). The cost of transmitting in a DIRECT network is high and therefore only high valued data gets through to the base station, clarifying why the efficiency is lower than that of the FORCED protocol. The MINT protocol has the poorest efficiency because each node only observes the myopic cost of transmission and therefore manages to transmit most of its data without realising that the overall energy expenditure in the path leading to the base station may be very high. USAC nodes, on the other hand, are very selective in transmitting their data and only do so when it is worthwhile (irrespective of the type of environment they are in). Therefore the USAC plot shows an increasing efficiency as the environment becomes more dynamic. The rate of USAC efficiency increase is very simlar to that of the OPTIMAL protocol. Overall, USAC preforms at an efficiency of 70% that of OPTIMAL.

5.5 Summary

The protocol described in this chapter allows agents to act in a decentralised manner, based on the nature of their local environment, while self-organising to form a network whose performance is high in terms of minimising energy consumption and maximising the value of data gained. It makes use of the localisation ability of individual agents to determine the cheapest cost path to the sink and incorporates the value of the observed data to calculate the cheapest path. It has been demonstrated that USAC is far superior to the one currently deployed in GlacsWeb. Furthermore, results from the simulation also demonstrate that USAC performs exceedingly better, in terms of information gain and extending network lifetime, in comparison to two other benchmark protocols in literature. This is the case even when the network grows in size, the topology of the network changes frequently and when the nature of observed environment varies dynamically. This chapter has also shown that that USAC is also robust in the face of node failure and that its performance is much closer to the optimal protocol than others.

Chapter 6

Conclusions

This chapter provides a summary of the research carried out and presents a number of avenues for future research. Initially, it discusses the implications of the work done as part of this thesis and enumerates the key research achievements. The chapter concludes with a number of future research areas which were identified throughout this research, and which address some of the main limitations of the work presented.

6.1 Implications of the Research

The emergence of large scale, decentralized, autonomous, peer-to-peer systems (Milojicic et al., 2002) is a spectacular phenomenon that has generated a new level of network programming abstraction and presents significant challenges for parallel and distributed computing, distributed data management, and software engineering. This is a fundamental shift from the current client-server based systems.

Advancing peer-to-peer applications from the basics of file sharing towards more general and complex resource sharing, process management, and ultimately towards a sea of global applications such as sensor networks, requires significant understanding and study of algorithms and network programming technologies. As network technologies continue to expand into wireless and ad-hoc networking domains, these applications are becoming more open and complex. In this context, open means that nodes contained within such systems are free to enter and leave at their own will. Furthermore, many of these nodes are heavily constrained in their processing ability, communication bandwidth, memory storage and energy supply. In such environments, these nodes have to make sound decisions to achieve their goals, but are faced with a large amount of uncertainty in this decision-making process. Much of this uncertainty is caused by limited resources available to these nodes, which heavily restricts them in their actions.

Given this background, it is important in such environments to assure efficient use of the nodes' resources if they are to successfully complete their goals. Mutual cooperation is a concept integral to human society, allowing us to effectively make decisions in the presence of uncertainty. This thesis has used this as inspiration, and presented two novel computational models of a cooperative network and shown its application in a sensor network called GlacsWeb. In a dynamic and hostile environment that of a glacier, these models enable the nodes to make effective and sound decisions in light of the inherent uncertainty that exists within.

In summary, the research presented in the thesis has achieved the following:

1. In response to the fact that no one state of the art sensor network model is generic enough to be used in all application purposes, the research in this thesis has been developed as part of a new wireless sensor network called GlacsWeb (Chapter 3), that is capable of measuring several parameters relating to the sub-glacial environment. Work done in this avenue has involved placing sensor nodes in extremely hostile environments: underneath, on and inside the glacial ice. These hazards were not confined to mechanical problems due to the extreme weather and cold but also to attenuation in radio signals used for inter-node communication. GlacsWeb sensor nodes have facilitated in achieving some of the key glaciological objectives such as studying the motion of small rocks in the sub-glacial bed and measuring other parameters key to understanding sub-glacial dynamics. In doing so the thesis has achieved Aims 1 and 2 highlighted in Section 1.3. These are:

   (a) Miniaturisation: Successfully designed the final GlacsWeb probe harbouring the radio, processor and power modules in a palm-sized device so that it could be mimic a small rock and be suitable for easy insertion into the glacier.

   (b) Low Power Design: The probe and base station hardware were designed to communicate everyday and physical conditions permitting, could theoretically survive unattended for months. A few probes communicated with the base station for up to 24 months until the latter itself broke down by falling into a crevasse.

2. Developing the hardware for this network has been a great learning experience and has added to the diversity useful to uncover new ideas and concepts. This aspect of research has necessitated an interdisciplinary team including people from the fields of glaciology, electronics, computer science, communications, mechanical engineering and surveying. The experience has shown that a prototype installation in the field is necessary in order to actually solve all the engineering problems. It has also highlighted the importance of ad-hoc networking, remote diagnostics and control. The dynamic nature of the glacial environment, di-electric properties of ice and the en-glacial water bodies posed various challenges in the appropriate use

radio technology. In order to overcome those, this thesis successfully achieved Aim 3 which comprised of the following work:

(a) Ad-hoc networking: The study identified the one-hop link between the sensor nodes and the base station as one of the key reasons for the high failure rate within the initial GlacsWeb network. In order to facilitate communication between nodes that got "carried away" too far by the glacier and the base station, it designed a new communications layer in GWMAC (Chapter 4), a MAC protocol, that established multi-hopping amongst the nodes. GWMAC ensured that packet collisions were avoided and that unnecessary energy was not wasted in overhearing packets destined for other probes.

(b) Radio technology: Overcoming radio communication in ice was a major challenge. Radio communication was successfully improved at every deployment stage by consistently reducing the transmission frequency from 868MHz to 173 MHz, thereby effectively increasing the wavelength so that the radio waves would not be obstructed by the the en-glacial water bodies. In addition, the transmission power was increased to overcome attenuation, however, GWMAC ensured that energy required during transmission was consumed efficiently.

3. In Chapter 5, the state of the art was extended by developing a Utility based Sensing And Communication protocol (USAC) that established autonomous and adaptive behaviour within the GlacsWeb probes. In USAC, this thesis has described, for the first time, a protocol that meticulously combines both elements of sensing and communication in minimising the energy consumption of the nodes and thereby extending the lifetime of a cooperative sensor network. In doing so it has achieved Aims 4 and 5. These are:

(a) Adaptive Sensing: A Bayesian linear regression approach (due to the piecewise linearity exhibited in the data collected from the GlacsWeb network nodes) has been proposed for the GlacsWeb nodes to formulate the model of their environment and thereby establishing a sensing schedule accordingly. Furthermore, a valuation technique based on Kullback-Leibler divergence has been devised which is used by the node to identify the importance of each data sample.

(b) Intelligent Routing: The study argues that the importance of transmitting any sensed data is directly dependent on the opportunity cost of the energy constrained probe, especially when it is relaying data from other probes. It, therefore, established a cost identification mechanism that relied on the dynamism of the local environment of each node. This forced the nodes to do the following:

i. Choose the lowest cost route to transmit data to the base station. However,

ii. Only transmit if the value of data was higher than the cost.

The USAC protocol demonstrated that the probes operated normally even when additional new nodes were incorporated within the network. It also exhibited a distributed and robust characteristic where probes could identify new routes if some neighbours failed.

4. The research has shown that both protocols (GWMAC and USAC) are a highly energy efficient in governing interactions amongst the nodes in the network.

   (a) Through evaluation in simulation, it has demonstrated that for a dynamic application like GlacsWeb, GWMAC results in a better performance than the approach used by several benchmark MAC protocols including S-MAC and T-MAC. The general architecture of GWMAC is based on scheduling and time division multiple accesses (TDMA) and extensive series of simulations were performed to evaluate the claims made in this research. Results in Chapter 4 illustrate that on average GWMAC can increase the network life time by at least 63%. This also has a significant effect on the amount of data that can be collected over the network life time.

   (b) USAC has been evaluated by examining the impact on efficiency of a static network topology, a dynamic network topology, the size of the network, the degree of dynamism of the environment and the mobility of the nodes. In so doing, study has demonstrated that the efficiency gains of this new protocol, over the original GlacsWeb implementation over a 6 month period, are 78%, 133% , 100% and 93% respectively. Furthermore, it has been shown that USAC performs at 65%, 70%, 63% and 70% of the theoretical optimal respectively, despite being a distributed protocol that operates with incomplete knowledge of the environment. The study also demonstrated that the strategies employed in the USAC protocol boosts energy efficiency of the probes and extends network lifetime significantly in comparison to some of the well-known strategies studied in literature.

In closing, the remainder of this chapter presents ways in which the research contained in this thesis can be extended.

6.2 Further Research

The research presented in this thesis provides a solid basis for further research. Below we detail a number of avenues for further research, in which GlacsWeb, GWMAC and USAC can be used as the based models upon which the proposed issues can be addressed.

Future of GlacsWeb — Designing a sensor network for glaciers was a challenging task because of the problems in predicting the behaviour of radio systems and power sources and because of difficulties in building electronic devices that are sufficiently strong and waterproofed to survive such a hostile environment. The probes transmitted their data from 1366 days over at least 36m through ice and till, and provided data on temperature, water pressure, case stress, resistance and tilt angle. Weather, GPS, other glaciological data and diagnostic data were also collected. The next steps for this aspect of the research are:

1. To increase reliability of the probe communication Currently probes within the glacier get maneuvered, in several directions, changing the angle of the antenna installed. This contributes to being one of several other handicaps in reliable radio communication and therefore requires designing a more efficient radio antenna. Furthermore, probe electronics are vulnerable to moisture since water inevitably leaks through the polyester shell casing encapsulating the probe. Therefore an improved sealing technique needs to be in place so that they don't get easily destroyed due to en-glacial water bodies.

Extending the MAC protocol — Chapter 4 presents a MAC protocol that is robust and continues to function even if there are problems with parts of the network on which it has to operate. The next steps for extending the work done in GWMAC are:

1. **More efficient slot arrangement:** The manner in which the time slots are arranged is linear allowing only one node to transmit during that slot. By way of a more complex algorithm it is possible to allocate several nodes to share one time slot for transmitting to respective neighbours as long as they do not interfere in communication with each other.
2. **From simulation to practice:** Although, the research claims have been evaluated in simulation, and whilst the protocol was tested in the lab, it could not be deployed and tested efficiently in the field due to logistical reasons. This is because the Brikdalsbreen glacier, in which the nodes were meant to be deployed, receded dramatically. Setting up the base station and the drilling equipment on surface ice became increasingly difficult (due to lack of much flat ice) and dangerous (due to development of deep crevasses). Deployment work had to be abandoned and the GlacsWeb team had to investigate an alternative glacier. Whilst, a new glacier (Skalafellsjokull in Iceland) has been tested and identified as a potentially new deployment site, GWMAC needs to be implemented in the next generation of probes to be deployed there.

Coordinated Sensing — Whilst in Chapter 5, the research has specifically considered evaluating the effectiveness of USAC in the GlacsWeb application, the challenges involved here are very similar to those that occur in the design of many other sensor networks. For example, the possibility of using the work is being explored in the FloodNet[1] system (a sensor network for monitoring river levels in which the sensors are solar powered). Furthermore, the work proposes a Bayesian linear regression approach (due to the piece-wise linearity exhibited in the data collected from the GlacsWeb network nodes) for the agents to formulate the model of their environment. However, this can easily be substituted with the Gaussian process approach in cases where the model is highly non-linear and there is uncertainty regarding its the true functional form without affecting the overall architecture of the sensing and communication protocol.

USAC has been developed in the lab and the adaptive sampling mechanism in the protocol can be computationally time-consuming. Whilst this is acceptable for an application such as GlacsWeb, it may not be feasible for some real time sensor networks where latency is of prime importance. The focus of the work in this thesis was to a assign a value (importance) to sensed data and therefore it did not consider techniques to optimise computations. This is an interesting point of departure for further research in this regard so that USAC can be universally used.

To date, USAC corresponds to a single agent sensing model in which each individual makes decisions about when to sense independently of the other agents. As a result, this approach works well when the models of the data sensed by each agent are independent (or have a very low dependence between them). As part of future work, however, the research would like to address issues concerning sensor networks where data models from various agents are more highly correlated and this can be used to infer useful information. For example, if the resistivity sensors of nodes in a specific region of the glacier all observe a sudden decline, it may indicate that the ice in that particular region has melted and turned into water. To do this, USAC would need to be extended to a multi-agent sensing approach whereby the agents coordinate their sensing actions to maximise the information they extract from the environment.

Having established, through this research, an environmental sensor network for monitoring a glacier and developing a MAC protocol (GWMAC) and the first network protocol to incorporate both the tasks of sensing and communication (USAC), the objectives outlined above have to explored (and solutions developed) to ensure the adoption of such technology into other sensor network applications. If the challenges described are met, the resulting models will help promote the application of such techniques to a range of open and dynamic domains.

[1] http://envisense.org/floodnet/floodnet.htm

Appendix A

Dynamism of the Deployment Environment

The research in this thesis is motivated by the need to understand sub-glacial dynamics and how it affects the overall movement of the glacier. In this appendix we illustrate the timeline of the Brikdalsbreen glacier (Norway). This provides an idea about how much the glacier has receded from 2001 to 2006. In fact, the ice has receded so much that we had to abandon deployment in 2007 because it became too dangerous to work on its surface. This was mainly because the slope had become very steep and several crevasses had formed on the surface eliminating any possibility to survey the area and drill holes.

The following is a description of the figures contained within this appendix.

Figure A.1 this figure shows the snapshot of the glacier from 2001 to 2003. It can be seen that the glacier has a lot of flat surface area enabling both tourist excursion and scientific deployment on it.

Figure A.2 this figure shows the snapshot of the glacier from 2004 to 2006. It can be seen that the glacier recedes dramatically as the sub-glacial bed starts to appear on the sides.

Figure A.1: Melting sequence of Brikdalsbreen glacier from 2001 to 2003. It can be seen that the glacier has a lot of flat surface area enabling both tourist excursion and scientific deployment on it.

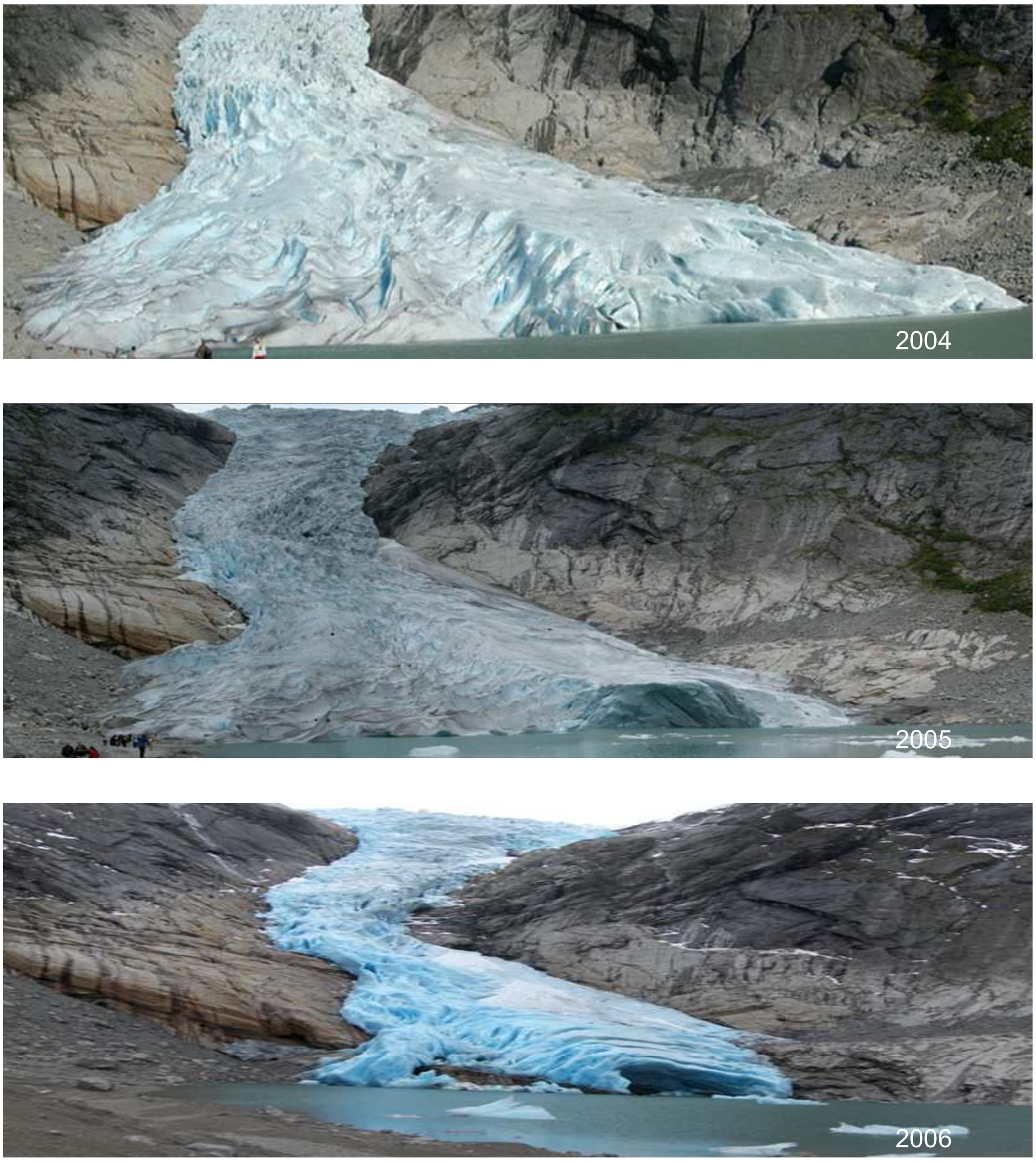


FIGURE A.2: Melting sequence of Brikdalsbreen glacier from 2004 to 2006. It can be seen that the glacier recedes dramatically as the lighter coloured sub-glacial bed starts to appear on the sides.

Appendix B

Best Linear Unbiased Estimate: Proof

Table B.1: Data for Multiple Linear Regression

Observation i	**Response x**	t_1	t_2	**...**	t_M
1	x_1	t_{11}	t_{12}	...	t_{1M}
2	x_2	t_{21}	t_{22}	...	t_{2M}
⋮	⋮	⋮	⋮		⋮
n	x_n	t_{n1}	t_{n2}	...	t_{nM}

$$x_i = w_0 + \sum_{j=1}^{M} w_j t_{ij} + \epsilon_i, \;\; i = 1, 2, \ldots, n \tag{B.1}$$

The least square function is:

$$S(w_0, w_1, w_2, \ldots, w_M) = \sum_{i}^{n} \epsilon_i^2 = \sum_{i=1}^{n} (x_i - w_0 - \sum_{j=1}^{M} w_j t_{ij})^2$$

The function S must be minimised with respect to $w_1, w_2, \ldots, w_M$. The least-squares estimators of $w_1, w_2, \ldots, w_M$ must satisfy

$$\frac{\delta S}{\delta w_0}|_{\widehat{w}_0, \widehat{w}_1, \ldots, \widehat{w}_M} = -2 \sum_{i=1}^{n} (x_i - \widehat{w}_0 - \sum_{j=1}^{M} \widehat{w}_j t_{ij}) = 0$$

and

$$\frac{\delta S}{\delta w_j}|_{\widehat{w}_0, \widehat{w}_1, \ldots, \widehat{w}_M} = -2 \sum_{i=1}^{n} (x_i - \widehat{w}_0 - \sum_{j=1}^{M} \widehat{w}_j t_{ij}) t_{ij} = 0, \quad j = 1, 2, \ldots, M$$

Simplifying the above equation, we obtain the least squares normal equations

$$n\widehat{w}_0 + \widehat{w}_1 \sum_{i=1}^{n} t_{i1} + \widehat{w}_2 \sum_{i=1}^{n} t_{i2} + \cdots + \widehat{w}_M \sum_{i=1}^{n} t_{iM} = \sum_{i=1}^{n} x_i$$

$$\widehat{w}_0 \sum_{i=1}^{n} t_{i1} + \widehat{w}_1 \sum_{i=1}^{n} t_{i1}^2 + \widehat{w}_2 \sum_{i=1}^{n} t_{i1}t_{i2} + \cdots + \widehat{w}_M \sum_{i=1}^{n} t_{i1}t_{iM} = \sum_{i=1}^{n} t_{i1}x_i$$

$$\vdots$$

$$\widehat{w}_0 \sum_{i=1}^{n} t_{iM} + \widehat{w}_1 \sum_{i=1}^{n} t_{iM}t_{i1} + \widehat{w}_2 \sum_{i=1}^{n} t_{iM}t_{i2} + \cdots + \widehat{w}_M \sum_{i=1}^{n} t_{iM}^2 = \sum_{i=1}^{n} t_{iM}x_i$$

Note there are $P = M + 1$ normal equations, one for each of the unknown regression coefficients. The solution to the normal equations will be the least square estimators $\widehat{w}_0, \widehat{w}_1, \widehat{w}_2, \ldots, \widehat{w}_M$.

To allow a very compact display of the model, data and results, it is more convenient to express the multiple regression models in matrix notation.

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \boldsymbol{T} = \begin{bmatrix} 1 & t_{11} & t_{12} & \ldots & t_{1M} \\ 1 & t_{21} & t_{22} & \ldots & t_{2M} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & t_{n1} & t_{n2} & \ldots & t_{nM} \end{bmatrix}$$

$$\boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix}, \quad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

In matrix notation, the model given by equation B.1 is

$$\boldsymbol{x} = \boldsymbol{T}\boldsymbol{w} + \boldsymbol{\epsilon}$$

In general, $\boldsymbol{x}$ is an $n \times 1$ vector of the observations, $\boldsymbol{T}$ is an $n \times p$ matrix of the levels of the regressor variables, $\boldsymbol{w}$ is a $p \times 1$ vector of regression coefficients and $\boldsymbol{\epsilon}$ is an $n \times 1$ vector of random errors.

Vector of least square estimators, $\widehat{\boldsymbol{w}}$ that minimises

$$S(\boldsymbol{w}) = \sum_{i=1}^{n} \epsilon_i^2 = \boldsymbol{\epsilon}^{\mathrm{T}}\boldsymbol{\epsilon} = (\boldsymbol{x} - \boldsymbol{T}\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{T}\boldsymbol{w})$$

$$= \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{T}^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{w} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{T}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{w}$$

$$= \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} - 2\boldsymbol{w}^{\mathrm{T}}\boldsymbol{T}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{T}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{w}$$

Since $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{T}^{\mathrm{T}}\boldsymbol{x}$ is a 1×1 matrix, or a scalar, and its transpose $(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{T}^{\mathrm{T}}\boldsymbol{x})^{\mathrm{T}} = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{w}$ is the same scalar. The least squares estimator msut satisfy

$$\frac{\delta S}{\delta \boldsymbol{w}}|_{\widehat{\boldsymbol{w}}} = -2\boldsymbol{T}^{\mathrm{T}}\boldsymbol{x} + 2\boldsymbol{T}^{\mathrm{T}}\boldsymbol{T}\widehat{\boldsymbol{w}} = 0$$

which simplifies to

$$\boldsymbol{T}^{\mathrm{T}}\boldsymbol{T}\widehat{\boldsymbol{w}} = \boldsymbol{T}^{\mathrm{T}}\boldsymbol{x}$$

$$\Rightarrow \quad \widehat{\boldsymbol{w}} = (\boldsymbol{T}^{\mathrm{T}}\boldsymbol{T})^{-1}\boldsymbol{T}^{\mathrm{T}}\boldsymbol{x}$$

Appendix C

Posterior Distribution as Gaussian with mean and covariance matrix: Proof

$$Posterior = \frac{Likelihood \times Prior}{MarginalLikelihood}$$

$$p(\boldsymbol{w} \mid T, \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid T, \boldsymbol{w})p(\boldsymbol{w})}{p(\boldsymbol{x} \mid T)}$$

where $p(\boldsymbol{w} \mid T, \boldsymbol{x})$ is also known as the normalising constant which is independent of the weights. The posterior in the above equation combines the likelihood and prior and captures everything we know about the parameters. Writing only the terms from the prior and likelihood which depends on weights and then completeing the square the following is obtained:

$$\begin{aligned} p(\boldsymbol{w} \mid T, \boldsymbol{x}) &= e^{-\frac{1}{2\sigma_n^2}(\boldsymbol{x}-T^T\boldsymbol{w})^T(\boldsymbol{x}-T^T\boldsymbol{w})} \times e^{-\frac{1}{2\sigma_n^2}\boldsymbol{w}^2} \\ &\propto e^{-\frac{1}{2\sigma_n^2}(\boldsymbol{x}-T^T\boldsymbol{w})^T(\boldsymbol{x}-T^T\boldsymbol{w})} \times e^{-\frac{1}{2}\boldsymbol{w}^T\Sigma_p^{-1}\boldsymbol{w}} \\ &\propto e^{-\frac{1}{2}(\boldsymbol{w}-\overline{\boldsymbol{w}})^T(\frac{1}{\sigma_n^2}TT^T+\Sigma_p^{-1})(\boldsymbol{w}-\overline{\boldsymbol{w}})} \end{aligned}$$

where $\overline{\boldsymbol{w}} = \sigma_n^{-2}(\sigma_n^{-2}TT^T + \Sigma_p^{-1})^{-1}T\boldsymbol{x}$. The form of the posterior distribution can then be recognised as Gaussian with mean $\overline{\boldsymbol{w}}$ and covariance matrix A^{-1}

$$p(\boldsymbol{w} \mid T, \boldsymbol{x}) = \mathcal{N}(\overline{\boldsymbol{w}}, A^{-1})$$

where $A = \sigma^{-2}TT^{\mathsf{T}} + \Sigma_p^{-1}$ and $\overline{\boldsymbol{w}} = \sigma^{-2}A^{-1}T\boldsymbol{x}$.

Appendix D

Piecewise Linearity in GlacsWeb

Work done in the Chapter 5 (USAC protocol) is based on the glaciologists assumption that the data collected from the sensor nodes exhibit piece-wise linearity characteristics.

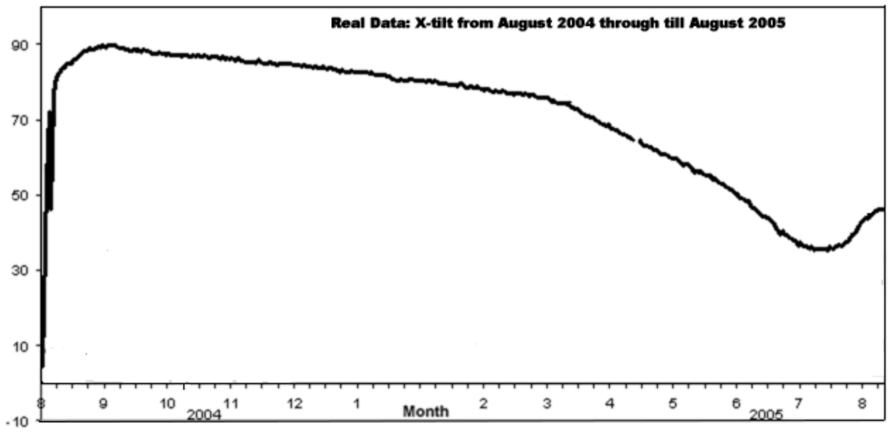


FIGURE D.1: Actual x-tilt data gathered from sensor node 8.

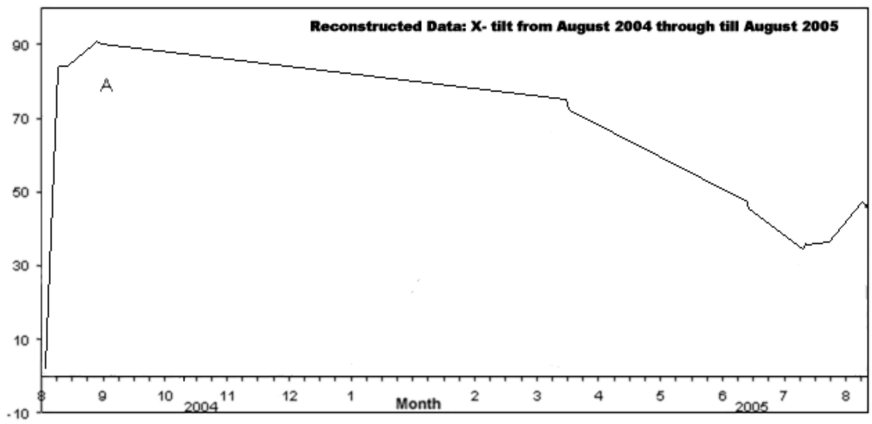


FIGURE D.2: Reconstructed x-tilt data from USAC's adaptive sampling mechanism.

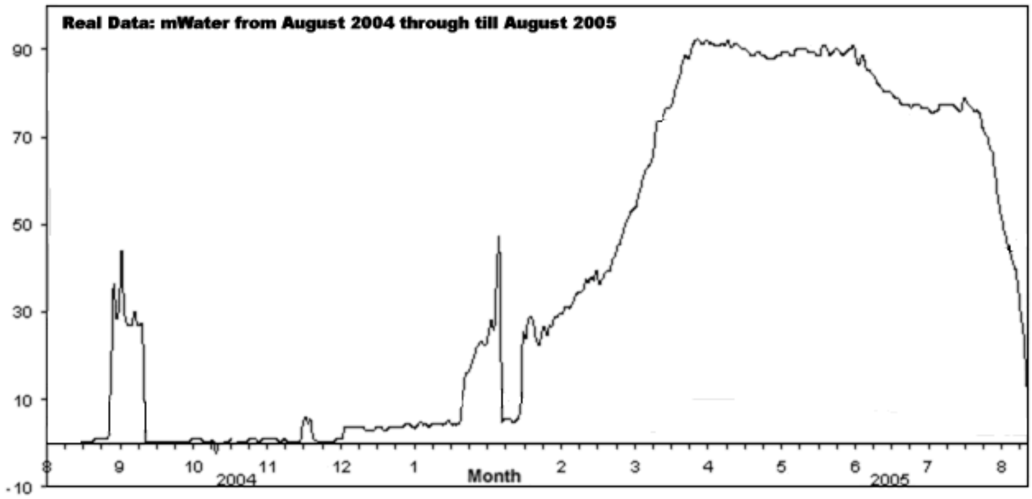


FIGURE D.3: Actual mWater data gathered from sensor node 8.

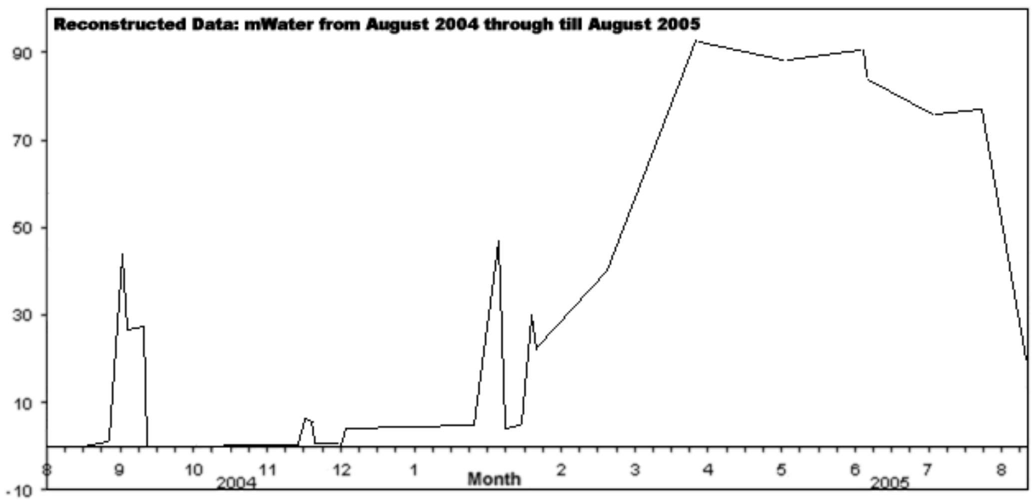


FIGURE D.4: Reconstructed mWater data from USAC's adaptive sampling mechanism.

Bibliography

N. Abramson. The aloha system. *Computer Networks*, pages 501–518, 1973.

N. Abramson. Development of the alohanet. *IEEE Transactions on Information Theory*, 31(2):119–123, 1985.

G.R. Aiello and G.D. Rogerson. Ultra-wideband wireless systems. *IEEE Microwave Magazine*, 4(2):36–47, 2003.

I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

R. B. Alley, D. D. Blakenship, C. R. Bentley, and S.T. Rooney. Deformation of till beneath ice stream b, west antarctica. *Nature*, 322:57–59, 1986.

Z-Wave Alliance. Z-wave specification. Technical Report SDS10243, Z-Wave Alliance, 2005a.

ZigBee Alliance. Zigbee specification. Technical report, Zigbee Alliance, 2005b.

S. P. Beeby, M. J. Tudor, and N. M. White. Energy harvesting vibration sources for microsystems applications. *Measurement Science and Technology*, 17(12):175–195, 2006.

T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 2001.

H. L. Bertoni. *Radio propagation for modern wireless systems*. Prentice Hall, 1999. ISBN 978-0130263735.

V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. Macaw: A media access protocol for wireless lans. In *Proceedings of ACM SIGCOMM*, pages 212–225, 1994.

K. Binmore. Why do people cooperate? *Politics Philosophy Economics*, 5(1):81–96, February 2006.

D. D. Blakenship, C. R. Bentley, S. T. Rooney, and R. B. Alley. Seismic measurements reveal a saturated porous layer beneath an active antarctic ice stream. *Nature*, 322: 54–57, 1986.

G. E. P. Box and G. C. Tiao. *Bayesian inference in statistical analysis*. John Wiley and Sons, 1992.

M. Britton and L. Sacks. The secoas project: Development of a self-organising, wireless sensor network for environmental monitoring. In *Proceedings of Second International Workshop on Sensor and Actor Network Protocols and Applications*, 2004.

W. F. Budd, D. Jenssen, and U. Radock. Calculated dielectric absorptions for antarctica and greenland. In P. Gudmandsen, editor, *Proceedings of the International Meeting on Glaciology*, page 7180, Sydney, Australia, 1970.

Y. U. Cao, A. S. Fukunaga, and A. B. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4(1):7–23, March 1997.

J. Capetanakis. Generalized tdma: The multi-accessing tree protocol. *IEEE Transactions on Communications*, 27(10):1476–1484, 1979. ISSN 0096-2244.

A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. Habitat monitoring: application driver for wireless communications technology. *SIGCOMM Computing Communication Review*, 31(2 supplement):20–41, 2001. ISSN 0146-4833.

B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: an energy-efficient coordination algorithm for topology maintenance in ad-hoc wireless networks. *Wireless Networks*, 8(5):481–494, 2002. ISSN 1022-0038.

S. Y. Cheung, S. C. Ergen, and P. Varaiya. Traffic surveillance with wireless magnetic sensors. In *Proceedings of the 12th ITS World Congress*, 2005.

C. Chong and S. P. Kumar. Sensor networks: Evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91:1247–1256, 2003.

M. Chu, H. Haussecker, and F. Zhao. Scalable information-driven sensor querying and routing for ad-hoc heterogeneous sensor networks. *International Journal of High Performance Computing Applications*, 16(3), August 2002.

T. V. Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, 2003.

R. K. Dash, A. Rogers, S. Reece, S. Roberts, and N. R. Jennings. Constrained bandwidth allocation in multi-sensor information fusion: a mechanism design approach. In *Proceedings of the 8th International Conference on Information Fusion*, 2005.

K. A. Delin, R. P. Harvey, N. A. Chabot, S. P. Jackson, M. Adams, D. W. Johnson, and J. T. Britton. Sensor web in antarctica: Developing an intelligent, autonomous platform for locating biological flourishes in cryogenic environments. In *Proceedings of Lunar and Planetary Institute Conference*, 2003.

K. A. Delin, S. P. Jackson, D. W. Johnson, S. C. Burleigh, R. R. Woodrow, J. M. McAuley, J. M. Dohm, F. Ip, T. P.A. Ferr, D. F. Rucker, and V. R. Baker. Environmental studies with the sensor web: principles and practice. *Sensors*, 5:103–117, 2005.

A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong. Model-based approximate querying in sensor networks. *International Journal on Very Large Data Bases*, 2005. To appear.

P. Djukic and S. Valaee. Distributed link scheduling for tdma mesh networks. In *In Proceeedings of IEEE International Conference on Communications*, pages 3823–3828, 2007. ISBN 1-4244-0353-7.

J. A. Dowdeswell and S. Evans. Investigations of the form and flow of ice sheets and glaciers using radio-echo sounding. *Reports on Progress in Physics*, 67(10):18211861, October 2004.

E. Elsaify, P. Padhy, K. Martinez, and G. Zou. Gw-mac: A tdma based mac protocol for a glacial sensor network. In *Proceedings of 4th ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, November 2007.

C. C. Enz, A. El-Hoiydi, J. Decotignie, and V. Peiris. Wisenet: An ultra low-power wireless sensor network solution. *IEEE Computer*, 37(8), 2004.

D. Estrin, D. Culler, K. Pister, and G. Sukhatme. Connecting the physical world with pervasive networks. *IEEE Pervasive Computing*, 1(1):59–69, Jan-March 2002. ISSN 1536-1268.

S. Evans and B. M. E. Smith. A radio echo equipment for depth sounding in polar ice sheets. *Journal of Physics E: Scientific Instruments*, 2(2):131136, 1969.

Q. Fang, F. Zhao, and L. Guibas. Lightweight sensing and communication protocols for target enumeration and aggregation. In *Proceedings of the 4th ACM International Symposium on Mobile Ad-hoc Networking and Computing*, pages 165–176, 2003.

I. Foster and C. Kesselman, editors. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999. ISBN 1-55860-475-8.

A. M. Gades, C. F. Raymond, H. Conway, H. Jacobel, and R. W. Jacobel. Bed properties of siple dome and adjacent ice streams (west antarctica), inferred from radio echosounding measurements. *Computer*, 46(152):88–94, 2000.

V. Garg, K. F. Sholik, and J. Wilkes. *Application of CDMA in Wireless/Personal Communications*. Prentice-Hall, Englewood Cliffs, 1996.

B. Gedik, L. Ling, and P. S. Yu. Asap: An adaptive sampling approach to data collection in sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 18:1766–1783, December 2007.

J. W. Glen and J. G. Paren. The electrical properties of snow and ice. *Journal of Glaciology*, 15(73):15–38, 1975.

S. Glisic and B. Vucetic. *Spread Spectrum CDMA Systems for Wireless Communications*. Artech House, 1998.

D. A. Gratton, editor. *Bluetooth Profiles: The Definitive Guide*. Prentice Hall, 2002.

J. A. Gutierrez, M. Naeve, E. Callaway, M. Bourgeois, V. Mitter, and B. Heile. Ieee 802.15.4: a developing standard for low-power low-cost wireless personal area networks. *IEEE Network*, 15(5):12–19, 2001. ISSN 0890-8044.

Q. Han, S. Mehrotra, and N. Venkatasubramanian. Energy efficient data collection in distributed sensor environments. In *Proceedings of 24th IEEE International Conference on Distributed Computing Systems*, pages 590– 597, 2004.

T. He, S. Krishnamurthy, J. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh. An energy-efficient surveillance system using wireless sensor networks. In *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys04)*, 2004.

S. Hedetniemi, S. Hedetniemi, and A. Liestman. A survey of gossiping and broadcasting in communication networks. *IEEE Networks*, 18, 1988.

H. Heidt, J. Puig Suari, A. Moore, S. Nakasuka, and R. Twiggs. Cubesat: A new generation of picosatellite for education and industry low cost space experimentation. In *Proceedings of the 14th Annual AIAA/USU Small Satellite Conference*, 2000.

W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *IEEE Proceedings International Conference on System Sciences*, pages 1–10, 2000.

W. R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of ACM MobiCom*, pages 174–185, 1999.

IEEE, editor. *Wireless LAN medium access control (MAC) and physical layer (PHY) specification IEEE Standar 802.11*. LAN MAN Standards Committee of the IEEE Computer Society, NY, USA, 1999.

Texas Instruments Incorporated. Low power rf/ics. http://focus.ti.com/docs/prod/folders/print/cc2430.html, December 2008.

C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of ACM MobiCom*, pages 56–67, 2000.

A. Jain and E. Y. Chang. Adaptive sampling for sensor networks. In *In Proceeedings of the 1st International Workshop on Data management for Sensor Networks*, pages 10–16, New York, NY, USA, 2004. ACM Press.

K. Jamieson, H. Balakrishnan, and Y. C. Tay. Sift: A mac protocol for event-driven wireless sensor networks. Technical report, Massachussets Institute of Technology, 2003.

N. R. Jennings. Agent oriented software engineering. *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World: Multi-Agent System Engineering*, pages 1–7, 1999.

N. R. Jennings. An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4):35–41, 2001.

N. R. Jennings, S. D. Ramchurn, M. Allen-Williams, R. K. Dash, P. S. Dutta, A. Rogers, and I. Vetsikas. The aladdin project: Agent technology to the rescue. In *In Proceedings of the 1st AAMAS Workshop on Agent Technology for Disaster Management*, 2006.

P. Karn. Maca: A new channel access method for packet radio. In *Proceedings of the 9th ARRL Computer Networking Conference*, pages 134–140, 1990.

L. Kleinrock and F. Tobagi. Packet switching in radio channels: Part i - carrier sense multiple access modes and their throughput delay characteristics. *IEEE Transactions on Communications*, 23(12):1400–1416, 1975.

Y. Kotidis. Snapshot queries: Towards data-centric sensor networks. In *Proceedings of 21st IEEE International Conference on Data Engineering*, pages 131–142, 2005.

S. Kullback. *Information theory and statistics*. John Wiley and Sons, New York, 2001.

S. Kullback and R. A. Leibler. A probabilistic approach to concurrent mapping and localization for mobile robots. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

W. C. Y. Lee, editor. *Mobile Cellular Telecommunications - Analog and Digital Systems*. McGraw-Hill Inc., 2nd edition, 1995.

L. Li and J. Y Halpern. Minimum energy mobile wireless networks revisited. In *Proceedings of IEEE International Conference on Communications*, 2001.

Q. Li, J. Aslam, and D. Rus. Hierarchical power-aware routing in sensor networks. In *Proceedings of the DIMACS Workshop on Pervasive Networking*, May 2001.

P. Lin, C. Qiao, and X. Wang. Medium access control with a dynamic duty cycle for sensor networks. In *Proceedings of IEEE Wireless Communications and Networking Conference*, volume 3, page 153439, 2004.

S. Lindsey and C. S. Raghavendra. Pegasis: Power efficient gathering in sensor information systems. In *Proceedings of the IEEE conference on Aerospace*, March 2002.

K. Lorincz, D. Malan, T. R. F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G.Mainland, S. Moulton, and M. Welsh. Sensor networks for emergency response: challenges and opportunities. *Special Issue on Pervasive Computing for First Response*, Oct-Dec 2004.

M. Luck and M. d'Inverno, editors. *Understanding Agent Systems*. Springer 2nd edition, London, UK, 2004.

M. Luck, P. McBurney, O. Shehory, and S. Wilmott, editors. *Agent Technology: computing as interaction, A Roadmap for agent-based computing*. Agentlink, 2005.

A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pages 88–97. ACM, 2002. ISBN 1-58113-589-0.

A. Manjeshwar and D. P. Agrawal. Teen: A routing protocol for enhanced efficiency in wireless sensor networks. In *Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, Los Alamitos, CA, USA, 2001. IEEE Computer Society.

A. Manjeshwar and D. P. Agrawal. Apteen: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. In *Proceedings of the International Parallel and Distributed Processing Symposium*, pages 195–202, 2002.

K. Martinez, J. K. Hart, and R. Ong. Environmental sensor networks. *IEEE Computer*, 37(8):50–56, 2004.

K. Martinez, P. Padhy, E. Elsaify, and G. Zou. Deploying a sensor network in an extreme environment. In *Proceedings of the conference on Sensor Networks Ubiquitous and Trustworthy Computing*, 2006.

D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu. Peer-to-peer computing. Technical report, HP Laboratories, 2002.

S. Ni, Y. Tseng, Y. Chen, and J. Sheu. The broadcast storm problem in a mobile ad-hoc network. *Wireless Networks*, 8:153–167, Mar-May 2002.

T. Nieberg, S. Dulman, P. Havinga, L. van Hoesel, and J. Wu. Collaborative algorithms for communication in wireless sensor networks. In *Ambient Intelligence: Impact on Embedded System Design*, 2003.

P. Ogren, E. Fiorelli, and N. E. Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, 49(8):35–41, 2004.

C. Olston, B. T. Loo, and J. Widom. Adaptive precision setting for cached approximate values. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 355–366, Santa Barbara, California, may 2001.

C. Otto, A. Milenkovic, C. Sanders, and E. Jovanov. System architecture of a wireless body area sensor network for ubiquitous health monitoring. *Journal of Mobile Multimedia*, 1:307–326, 2006.

P. Padhy, R. Dash, K. Martinez, and N. R. Jennings. A utility based sensing and communication model for a glacial sensor network. In *Proceedings of the Conference on Autonmous Agents and Multi Agent Systems*, 2006.

P. Padhy, K. Martinez, A. Riddoch, J. K. Hart, and H. L. R. Ong. Glacial environment monitoring using sensor networks. In *Proceedings of the First Real-World Wireless Sensor Networks Workshop(REALWSN'05)*, pages 10–14, 2005.

M. Rahimi, R. Pon, W. J. Kaiser, G. S. Sukhatme, D. Estrin, and M. Sirivastava. Adaptive sampling for environmental robotics. In *Proceedings of IEEE International Conference on Robotics and Automation*, page 35373544, 2004.

V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. In *Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems*, pages 181–192, Los Angeles, CA, 2003.

C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. The MIT Press, 2005. ISBN 026218253X.

A. Rogers, E. David, and N. R. Jennings. Self organised routing for wireless micro-sensor networks. *IEEE Transactions on Systems, Man and Cybernetics: Part A*, 35:349359, 2005.

D. De Roure. Floodnet: A new flood warning system. *Royal Academy of Engineering Quarterly*, 23:48–51, 2005.

S. Russell and P. Norvig, editors. *Artificial Intelligence A Modern Approach*. Prentice Hall, 2nd edition, 2003.

N. Sadagopan, B. Krishnamachari, and A. Helmy. Active query forwarding in sensor networks (acquire). *Journal on Ad Hoc Networks*, 3(1):91–113, 2005.

C. Schurgers and M. B. Srivastava. Energy efficient routing in wireless sensor networks. In *In the MILCOM Proceedings on Communications for Network-Centric Operations: Creating the Information Force*, pages 304–309, 2001.

R. Shah and J. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, March 2002.

S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin. Data-centric storage in sensor networks. In *Proceedings of the First ACM SIGCOMM Workshop on Hot Topics in Networks*, 2002.

E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan. Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. In *Proceedings of the ACM MobiCom*, pages 272–286, 2001.

D. C. Smith, A. Cypher, and J. Spohrer. Kidsim: programming agents without a programming language. *Communications of the ACM*, 37(7):5467, 1994. ISSN 0001-0782.

K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, pages 16–27, 2000.

C. E. Spurgeon. *Ethernet: the definitive guide*. O'Reilly and Associates, Inc., Sebastopol, CA, USA, 2000. ISBN 1-56592-660-9.

D. C. Steere, A. M. Baptista, D. McNamee, C. Pu, and J. Walpole. Research challenges in environmental observation and forecasting systems. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 292–299, 2000.

M. Stemm and R. Katz. Measuring and reducing energy consumption of network interfaces in hand-held devices. *IEICE Transactions on Communications*, 1997.

N. Tatbul, S. Zdonik, M. Chemiack, and M. Stonebraker. Load shedding in a data stream manager. In *Proceedings of the Conference on Very Large Data Bases*, 2003.

Crossbow Technology. Cross wireless modules. http://www.xbow.com/Products/wproductsoverview.aspx, December 2008.

F. Tobagi and L. Kleinrock. Packet switching in radio channels: Part ii - the hidden terminal problem in carrier sense multiple access and the busy-tone solution. *IEEE Transactions on Information Theory*, 23(12):1417–1433, 1975.

M. A. Weimer, T. S. Paing, and R. A. Zane. Remote area wind energy harvesting for low-power autonomous sensors. In *Proceedings of the IEEE Power Electronics Specialists Conference*, pages 1–5, 2006.

G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees. Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing*, 10(2):18–25, 2006. ISSN 1089-7801.

R. Willett, A. Martin, and R. Nowak. Backcasting: adaptive sampling for sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, pages 124–133, 2004. ISBN 1-58113-846-6.

C. K. I Williams. Prediction with gaussian processes: from linear regression to linear prediction and beyond. Technical Report NCRG/97/012, Neural Computing Research Group, Department of Computer Science and Applied Mathematics, Aston University, 1998.

G. Wittenburg, K. Terfloth, F. L. Villafuerte, T. Naumowicz, H. Ritter, and J. Schiller. Fence monitoring experimental evaluation of a use case for wireless sensor networks. In *Proceedings of the 4th European Conference on Wireless Sensor Networks (EWSN'07)*, 2007.

A. Woo, T. Tong, and D.Culler. Taming the underlying challenges or reliable multihop routing in sensor networks. In *Proceedings of the First International Conference on Embedded Networked Sensor Systems*, pages 14–27, 2003.

M. Wooldridge. Agent-based computing. *Interoperable Communication Networks*, 1(1): 71–97, 1998.

M. J. Wooldridge, editor. *Introduction to Multiagent Systems*. John Wiley & Sons, 2001.

M. J. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115152, 1995.

H. Wu, J. Lee, M. Hunter, R. M. Fujimoto, R. L. Guensler, and J. Ko. Simulated vehicle-to-vehicle message propagation efficiency on atlantas i-75 corridor. *Transportation Research Record, Journal of the Transportation Research Board*, 2005.

Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for adhoc routing. In *Proceedings of the 7th annual International Conference on Mobile Computing and Networking*, pages 70–84, New York, NY, USA, 2001. ACM. ISBN 1-58113-422-3.

G. Z. Yang, editor. *Body Sensor Networks*. Springer, 1st edition, 2006.

Y. Yao and J. Gehrke. The cougar approach to in-network query processing in sensor networks. *ACM SIGMOD Records*, 31(3):9–18, 2002. ISSN 0163-5808.

F. Ye, A. Chen, S. Liu, and L. Zhang. A scalable solution to minimum cost forwarding in large sensor networks. In *Proceedings of the 10th International Conference on Computer Communications and Networks*, pages 304–309, 2001.

F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang. Peas: A robust energy conserving protocol for long-lived sensor networks. In *Proceedings of the 23rd IEEE International Conference on Distributed Computing Systems*, pages 28–37, 2003.

W. Ye and J. Heidemann. Medium access control in wireless sensor networks. Technical report, University of South Carolina, 2003.

W. Ye, J. Heidemann, and D. Estrin. An energy efficient mac protocol for wireless sensor networks. In *Proceedings of the 21st Conference of the IEEE Computer and Communications Society (INFOCOM)*, pages 1567–1576, June 2002a.

W. Ye, J. Heidemann, and D. Estrin. Protocols for self-organization of a wireless sensor network. In *Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies*, NY, USA, 2002b.

Y. Yu, D. Estrin, and R. Govindan. Geographical and energy-aware routing: A recursive data dissemination protocol for wireless sensor networks. Technical Report UCLA-CSD TR-01-0023, UCLA Computer Science Department, May 2001.

M. Zhang, J. Song, and Y. Zhang. Three-tiered sensor networks architecture for traffic information monitoring and processing. *Intelligent Robots and Systems*, pages 2291–2296, 2005.