GWMAC- A TDMA Based MAC Protocol for a Glacial Sensor Network

A.Elsaify, P.Padhy, K.Martinez, G.Zou
School of Electronics and Computer Science
University of Southampton, United Kingdom
+44 (0)2380 594583
[ae, pp04r, km, gz]@ecs.soton.ac.uk

ABSTRACT

Wireless sensor networks demand the need to design practical and robust communication protocols to meet the application specifications. Our research focuses on designing and implementing an *environmental sensor network* to be used for sub-glacial study. The glacier is a very hostile environment presenting severe challenges and complications in the smooth functioning of such a network. In light of these challenges, we present a low power sensor node design and an energy-efficient *medium access control* protocol called GWMAC developed for a network deployed in a glacier in Norway. The general architecture of GWMAC is based on scheduling and *time division multiple accesses* (TDMA). We argue that for a highly dynamic network such as ours, GWMAC is more desirable over more widespread protocols such as S-MAC and LMAC. In doing so, we perform extensive series of simulations to empirically evaluate our claim. Our results 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 network life time.

Categories and Subject Descriptors

C. Computer Systems Organization,C.2 COMPUTER-COMMUNICATION NETWORKS, C.2.1 Network Architecture and Design, Subjects: Centralized networks. C.2.2 Network Protocols, Nouns: TDMA.

General Terms

Design, Algorithms, Reliability, Experimentation, Verification.

Keywords

WSN, ESN, Wireless Ad-hoc, MAC, TDMA.

1. INTRODUCTION

Environmental Sensor Networks (ESN) is an emerging field of research which combines many challenges from earth science and Wireless Sensors Networks (WSN). Furthermore, research in WSN brings together the challenges of modern computer science, ambient intelligence, wireless communication and mobile computing.

The GlacsWeb project [1] is a deployed WSN to directly monitor glaciers in order to provide data for glaciologists. The glaciologists use this data to study sub-glacial dynamics and understand glacier response to climate change. However, the glacial environment comprises of different media such as ice, water, air and till, all of which make radio communication between sensor nodes very unreliable. Furthermore, freezing temperatures and massive strain from the moving ice attests the deployment process to be a very tough challenge; restricting the size, shape and number of sensor nodes that can be deployed. This hostile nature of the environment puts the nodes at high risk of getting damaged and hence providing no guarantee of a set lifetime. Off the shelf sensor node hardware platforms such as Mica, BTnode and Intel-mote are not suitable for a glacier which requires radio modules to function at a much lower frequency in order to penetrate ice and water. Moreover, the requirement to install as many as 10 different sensors on each node with minimal power consumption and control capability poses additional challenges of designing low power hardware. These demanding requirements forced us to design custom-made node hardware and consequently necessitated the need for a new medium access control (MAC) protocol.

Against this background, in this paper, we present GWMAC 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. GWMAC is based on centralised Time Division Multiple Access (TDMA) that completely eliminates collisions, overhearing and idle listening.

The rest of this paper is structured as follows. Section 2 describes related work in this area and section 3 provides an overview of the GlacsWeb system and its components. Section 4 discusses the design of our MAC protocol detailing network discovery and setup of routing links between nodes. Section 5 empirically evaluates our protocol against existing decentralised protocols such as (LMAC and S-MAC) and section 6 concludes the paper.

2. RELATED WORK

It is widely recognised that there is a strong need for a well defined protocol stack in a WSN that helps combine power and routing awareness and that integrates data with the networking protocols. To this end, a number of MAC protocols have been investigated that can be broadly divided into two main categories. Namely, contention based and schedule based protocols. The primary aim of all MAC protocols is to avoid packet collisions, reduce idle listening time and curtail overhearing. Schedule based MAC eliminate contention and collisions by allowing several

PE-WASUN'07, October 22, 2007, Chania, Crete Island, Greece.

nodes to transmit in rapid succession in their uniquely assigned time slots. This approach provides deterministic guarantees about message communication and, therefore, is desirable for reliable dissemination of bulk data in sensor networks. However, the major drawbacks of using TDMA schemes include issues with scalability and the strong need for accurate time synchronisation between nodes.

The EYES [2] project addresses these challenges by proposing TDMA schemes that are not dependant on a central manager or base stations. Nodes frequently exchange control packets to maintain synchronisation and choose their own time slots, based on local information. Slots are assigned in such a way that no two nodes within a 2-hop neighbourhood control the same time slot.

Light-Weight MAC (LMAC) [3] is one such scheme where each node controls a unique slot. However, nodes still have to contend to transmit data to an intended receiver in it's time slot. The receiver (slot controller) is responsible for settling contention and deciding who it receives data from. Contention often leads to collisions and therefore such protocols require some form of Carrier Sense Multiple Access (CSMA) [4], where nodes commence their own transmission only after ensuring there is no other ongoing transmission. CSMA is often combined with Collision Avoidance (CSMA/CA)[4] to tackle the hidden terminal problem. However, this requires regular monitoring (idle listening) of the communication channel and may not be very suitable for some energy-scarce WSN networks.

S-MAC [5] another hybrid protocol tackles the problem of idle listening by making nodes constantly switch between two periodic states: *sleep* and *active*. During the latter, nodes either listen for any communication addressed to them or initiate communication themselves. Synchronisation is achieved with the exchange of relative (rather than absolute) time stamps. It implements both physical and virtual carrier sense. Overhearing is avoided by putting all immediate neighbours of the sender and the receiver to sleep for the duration of transmission.

T-MAC [6] improves on S-MAC's energy consumption by using a very short listening window at the beginning of each active period. This period is used to send or receive RTS and CTS packets. If no activity occurs within that period, the node is put to sleep thereby making the duty cycle more adaptive and saving power at a cost of reduced throughput and additional latency.

These protocols, whilst fully decentralised and distributed, still require an abundant use of control packets essential for coordinating transmit and receive actions between nodes. Moreover, they are catered for networks with a static topology and reliable communication links that avoid the need for regular network self-organisation. Furthermore, carrier sense doesn't fully guarantee avoidance of collisions on control packets. Thus the focus of our work is on a centralised TDMA based protocol called GWMAC designed for unreliable networks where contention is completely eliminated and control packets are minimised.

3. SYSTEM OVERVIEW

Before presenting GWMAC it is essential to provide a brief overview of the specifically designed GlacsWeb hardware that takes into account all the lessons learnt and experience gained from deploying the network in previous years.

The system uses sensors nodes (*GWnodes*) that specifically position by glaciologists inside the glacier (Figure 1). The base station is located on the surface of the glacier. Due to the significant radio losses in the upper ice layer, the nodes are unable to communicate directly with base station. In order to establish (or enhance) these communication links, the base station is connected to some of the nodes (called *anchor nodes*) via a serial cable (10-15m long). These anchor nodes are responsible for communicating with the remaining network on behalf of the base station.

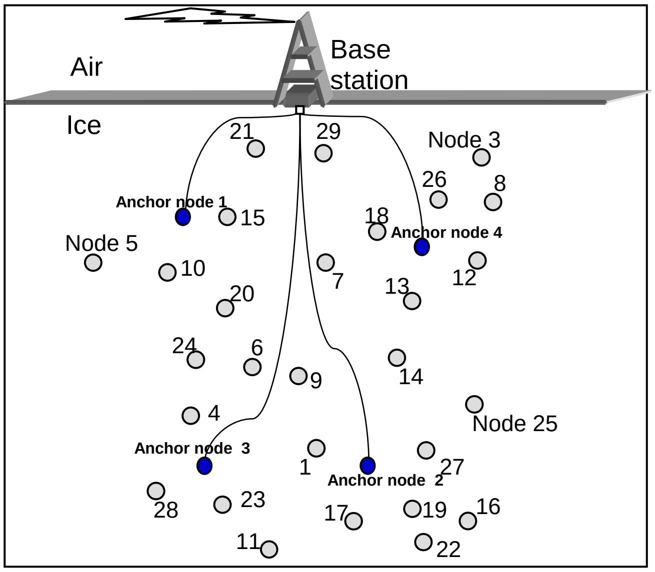

Figure 1. GlacsWeb System

The node is based on the PIC18 microcontroller controls the entire node system including reading the sensors and running the radio interface (Figure 2). The onboard 128 Kbyte memory is used to store programs along with the communication control data and the sensor data. It uses a very accurate external RTC (±2ppm accuracy), (from Maxim Semiconductor) to control wake up and sleep time.

The node hardware is integrated into one multilayer 40X50 mm PCB containing a Radiometrix transceiver and an antenna which are encapsulated in a permanently sealed polyester case as shown in Figure 3. The node runs in 5 different modes: *sleep mode* (all circuits are powered off apart from the RTC); *sensor mode* (only sensors are powered to take readings); *transmission mode*; *receiving mode* and *idle mode*. The energy consumption in each mode is presented in Table 1. Power to the node is provided by three AA-sized Lithium batteries (2.25Ah).

4. PROTOCOL DESIGN

The MAC protocol initially used by GlacsWeb [1] was a simple star network where nodes could only directly communicate with the base station. Furthermore, the hostile nature of the environment made it very difficult for this single hop based scheme to work efficiently. This called for the design of a more efficient protocol that would allow inter-node hopping.

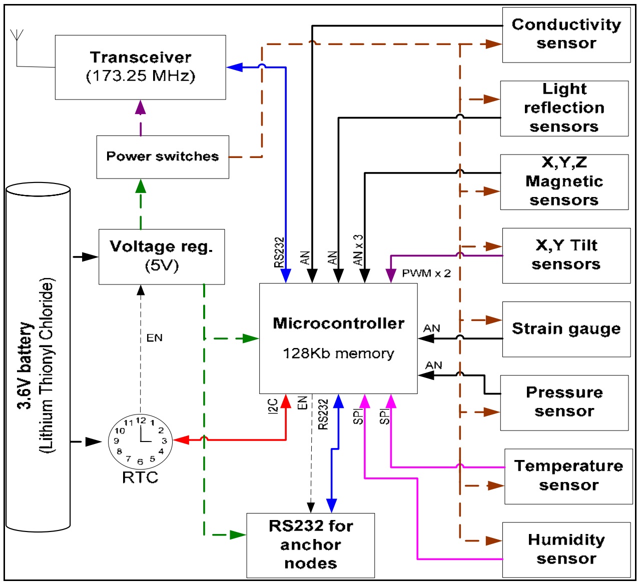

Figure 2. GWnode design architecture

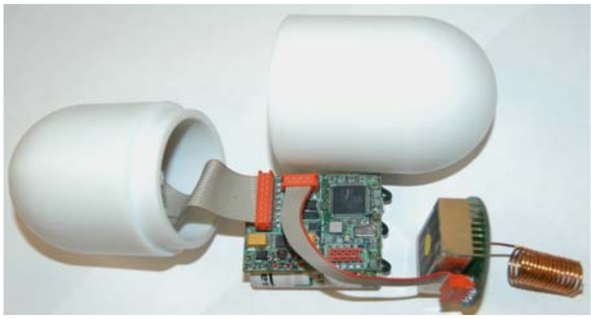

Figure 3. GWnode electronics and casing

Table 1. GW node current consumption

GWnode Mode	Current
Sleep	1 µA <
reading 10 Sensors (collective)	60 mA
Idle	10 mA
Transmitting at 10mW	35 mA
Transmitting at 100mW	90 mA
Receiving	18 mA
Switch From Sleep to Tx or RX	10 ms X (TX or RX cost)

In Section 3, we discussed several MAC protocols developed for various types of networks. Whilst these protocols achieve the primary goal of energy efficiency they are tailored for highly specialised networks with specific attributes. Certain characteristics of the GlacsWeb application made these protocols impractical for use. These characteristics 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 £200 to make. Hence 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, we had to design a protocol scalable enough to easily accommodate such network changes.

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 [7], some nodes may have dramatically more data to send than others during certain times. Therefore, we did not deem fairness 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 we expect its application 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 doesn't 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 frequency to reduce errors meant that we had to constantly thrive for an ideal throughput value limited by the selected radio transceiver capabilities.

4.1 Physical Layer

Specifying the radio communication infrastructure (for example choice of radio frequency, transceiver and size of antenna) has been a long-standing challenge in GlacsWeb. Earlier deployments of the network used radio frequencies of 868MHz and 433 MHz that resulted in very "lossy" communication due to presence of en-glacial water bodies. This prompted the use a transceiver with a lower frequency to enhance the communication signal. Unfortunately, due to legal requirements, we could only use license-exempted channels and this restricted us in our choice of transceivers for lower frequencies. After a thorough investigation we decided to use a commercially available single channel transceiver module (BiM1), manufactured by Radiometrix.

This BiM1 transceiver module operates at 173.250MHz and is licensed for general applications at 10mW. Most 173MHz commercial antennas are designed for applications in air and so their performance degrades severely inside a glacier because of the different dielectric properties of ice. Furthermore, they are too long (75 mm) to fit inside the node case. Therefore, we designed a compact 173 MHz helical antenna tuned to the ice after a series of experiments conducted in the glacier. The BiM1 module offers a maximum bit rate of 10Kbps, but it has to use 50:50 bit codes to avoid errors. We decided to use Manchester encoding for the signal, providing a throughput of 5Kbps and a byte time of 1.6ms. The module is also capable of measuring the received signal strength (RSSI) over a range of 60dB or more which is essential for a routing protocol algorithm to decide the best transmission paths and gateways for each node in the network.

A custom packet structure was used as shown in Figure 4. Each packet is made up of 64 bytes. The header comprises of 9 bytes leaving the payload to be up to 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 also shows the default time slot to be 130ms. With a maximum packet size being 102.4 (64 byte x 1.6ms) this allows 27.6ms for the preamble and guard band to switch ON and stabilize the transceiver. It also compensates for any unexpected time drift.

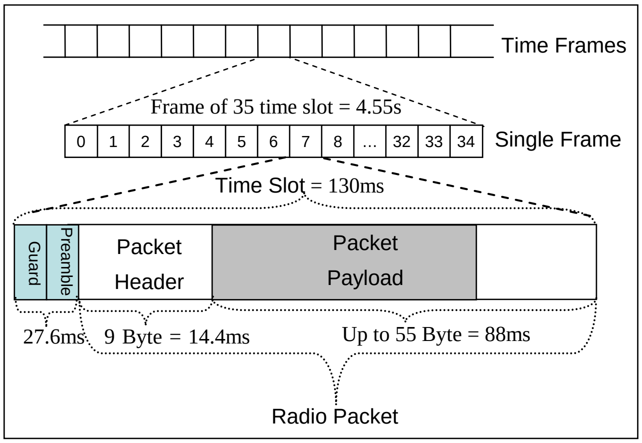

Figure 4. Frames, Time slots, and Packets

4.2 The Protocol Basics

4.2.1 Limited Communication Window

In many WSNs, there are several periods during which no sensing occurs resulting in a low data rate. Switching the transceivers on at these times may cause significant power wastage through idle listening. The S-MAC protocol reduces 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, this protocol caters for networks with near real-time use where the environment or target requires to be monitored very frequently (every few seconds). This has to be complemented with frequent radio communication to transfer the data.

In contrast, GlacsWeb is not a real-time system. The glacier is a slowly changing environment that doesn't require monitoring every few seconds like in the case of a surveillance network. Therefore, it requires less sensing and hence less communication. GWMAC reduces the duty cycle of the nodes to almost by zero providing only one small communication window per day. In this manner, the nodes activate their radio transceivers to communicate for a few minutes daily and sleep for the remainder of the time.

4.2.2 Basic Scheme

The basic scheme is shown in Figure 5. The timeline shows nodes frequently sensing (downward arrows) and communicating only during the communication window. Nodes sleep for majority of the time and wake up only during the communication window daily to transmit (or relay) data packets. Even during the communication window, nodes have their transceivers turned off for the major part (Idle state) and only turn them on either during time slots in which they are expected to receive data or slots uniquely assigned to them for transmission.

4.2.3 Frames and time slots

Like most TDMA based protocols, GWMAC also divides time into frames. These frames are further divided into slots as shown in Figure 4. The number of slots per 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 in the frame are dynamically increased or decreased respectively during the *network discovery phase* (See section 4.3). 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. The task of assigning a time slot to a node is delegated to the scheduling algorithm on the base station.

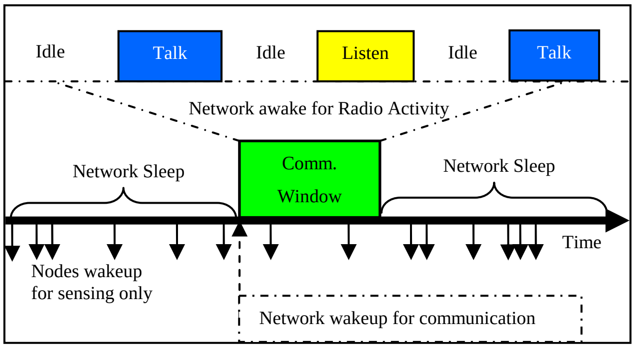

Figure 5. Node wake up and network communication window

4.2.4 Base station and Anchor nodes

As mentioned earlier, in order to enhance communication between the base station and the ice-embedded nodes, the base station is connected to *n* number of anchor nodes (Figure 1). These anchor nodes are embedded in the ice and are fully controlled by the base station in a master-slave manner. They are responsible for communicating with the remaining wireless nodes. The base station can easily identify damage or disconnection of these anchor nodes. Network hops are organised around these anchor and they always occupy the first few TDMA slots in any frame.

4.3 Network Discovery and Configuration

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 frequent initiations). This is to ensure nodes are not lost and multi-hopping routing is always optimised in the face topology change. This section describes the phases through which the network nodes are discovered and configured to carry out the ad-hoc network activities.

4.3.1 Network Discovery Phase

This phase combines the use of TDMA with an optimised ad-hoc flooding technique to broadcast messages and retrieve logistic information from the nodes. This combination overcomes the potential problems of message redundancy and packet collisions as a result of flooding [8]. The phase makes use default network schedule where each node is allocated a time slot equal to its own ID. For example, in Figure 1, 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 by synchronising all the active anchor nodes. Two different command messages are broadcasted to collect information about the network structure. These are the following:

1. *Direct Echo* (DE). The anchor nodes broadcast a DE command in their respective slots as shown in Figure 6. Each non-anchor node within range, 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, using a list of all deployed nodes, is able to establish all nodes within one-hop range along with those that are not (missing nodes).

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 it's 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 number of frames or depth of exploration is controlled by a time-to-live parameter embedded within the spread echo packet and decremented with each forward transmission. Transmission of SE stops when the parameter decrements to zero upon which *Spread Echo Reply* (SER) message is initiated. 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 reaches the base station (anchor nodes) in quickest manner.

4.3.2 Network Configuration Phase

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

a) *Assign optimised time slots*. The scheduling algorithm in the base station assigns new time slots to nodes in sequential order in the time frame. The assignment is based on how network hop level of the nodes. Figure 7 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.*

b) *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.

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. *Assign 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.

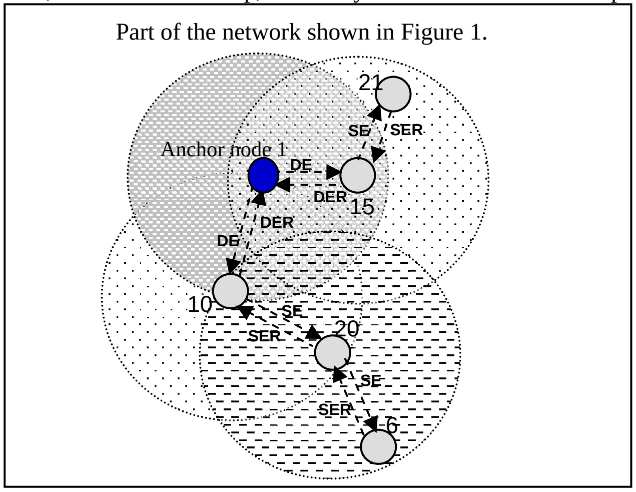

Figure 6. Network Discovery messages

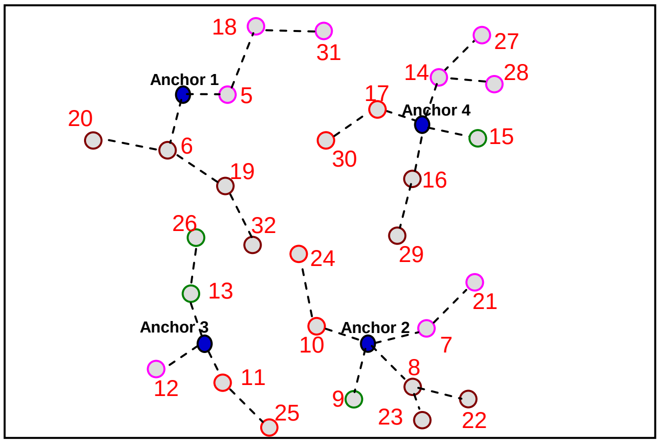

Figure 7: Network shown in Figure 1 after configuration

4.4 Data Acquisition

Data acquisition is scheduled either immediately after the network configuration phase or during the start of the communication window if the base station believes the network hasn't lost its configuration since last time. This phase is initiated by the base station by sending a *Get Data* command in downlink mode to the network. In this mode, all nodes use their recorded configuration to forward the command in their newly assigned time slot $S_{assigned}$, to there allotted children in one super frame. This is followed by the uplink mode where each node *i* transmits its data in time slot S_{uplink}^{i} such that

$$S_{uplink}^{i} = S_{frame} - S_{assigned}^{i}$$

Where S_{frame} are the total number slots in a frame. This ensures that each packet is transmitted from the originator to its intended recipient in one *super frame*. In their reply packets, along with the data, the nodes indicate how much more data needs to be sent or forwarded from others so that the number of uplink frames are adjusted accordingly for all those nodes that fall in that particular multi-hop path. Only nodes that are assigned to more uplink frames stay awake for consequent data packets whilst the rest go to sleep.

4.5 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. Furthermore, the network administrator is also able to send commands live when the system is awake by remote login to the base station.

4.6 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 millisecond time stamp value (time at transmission) embedded within the packet, the time the packet took to reach it from the transmitting node (time of flight) and time of processing. Getting the time of flight isn't easy but it is quite small and it can be compensated for by the available guard time. Thus, a node can be synchronised to the current time $T_{current}$ using the following equation:

$$T_{current} = T_{transmission} + T_{preamble} + T_{switching} + (1.6ms * (H + P))$$

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

4.7 Collision and Overhearing Avoidance

The GWMAC protocol adopts a TDMA scheme during both network configuration phase as well as data transmission phase thereby eliminating any form of collisions. One may argue that a 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 node since.

This problem is rectified by programming the nodes to only listen and not transmit until the next network discovery phase if they have not participated in any form of communication within the network for a set period of time. Furthermore, overhearing is minimised as nodes only turn on their receivers during their parents' and children's time slots and sleep the rest of the time.

4.8 Acknowledgment Omission

We mentioned how the use of the BiM1 radio transceiver significantly reduces the bit rate (see section 4.1). This had a significant impact on the use of control packets in GWMAC. The size of time slots used in the TDMA scheme is already too big. Incorporating acknowledgements packets within the scheme to confirm receipt of a data or command packet would have further increased their size. Therefore, we decided to omit the use of acknowledgement packets. Failure to receive data from any node is detected by the base station when it realises missing data from the node at the end of the collection period. This can result in the base requesting the data again from the missing node.

5. EXPERIMENTAL EVALUATION

After having implemented GWMAC on a test bed of 10 GlacsWeb nodes, we decided to evaluate its effectiveness and relative performance in comparison to existing protocols such as LMAC and S-MAC through simulations. A customised sensor network simulator was developed in Java specifically to program the three protocols. The network in the simulations was subjected to change in size (number of nodes), change in average traffic (number of data packets Originated per node) and topology change to plot performance graphs of the three protocols in terms of network lifetime and data gathered over the network lifetime. In addition, we compared the energy expenditure of the nodes during the self-organisation phase (network discovery and configuration) of the network enabling the network administrator to analyse and adjust the frequency of this phase accordingly. To plot the result graphs, we conducted 3 different types of simulations. The first simulation was run to establish the average network lifetime of the network using each MAC protocol. The second simulation was conducted to demonstrate the total data collected by the base station for each MAC protocol over the lifetime of the network. The third simulation was run to demonstrate the total energy consumed by the entire network during the self-organisation phase.

5.1 Setup and Network Parameters

The wireless sensor network simulator was customised and designed independently in Java to give 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 transceiver so that it is ready to receive data or carrier sense), transmit a single packet, receive a single packet 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. We decided to ignore the processing action of the node due to its near negligible power consumption. Specifically, Table 2 shows the typical energy consumption of each action based on the values obtained from the designed GWnode hardware. All nodes were initialised with an energy capacity of 1000 Joules. 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 LMAC slot time was set to 200ms to encompass 130ms for the GlacsWeb packet and 70ms to settle contention for transmission and other control packets. The sleep time for S-MAC was set to 600ms. For a fair comparison between the three protocols the periodic listen and sleep of S-MAC was only activated during the same window of communication as GWMAC. In other words, nodes sleep (and sense) for most of the time but only communicate during set predefined windows during which all three MAC protocols execute. In addition, we ensured that data traffic in the network was not constant so that neither protocol could gain advantage. The idea of varied traffic stems from [7] where we discuss the importance of minimising data transmission through selective sensing and only transmitting important data. Whilst this is a separate research problem, it is important to mention that adaptive sensing and transmission can produce varied data traffic that can greatly affect energy consumption and therefore network lifetime. We also made nodes move about randomly in the network to force a topology change from time to time. Finally, in order to obtain statistically significant results, we report average results of 200 simulations in each of the experiments carried out. The environment data observed by the nodes in the simulation were derived from segments of the data collected by actual deployed nodes network in Norway.

Table 2. Energy Consumption in each action of Sensor Node

Sensor Action	Energy consumption (J)
Transmission per packet	0.5
Reception per packet	0.15
Idle transceiver on	0.105
Sense	0.3
Sleep per second	105µ

5.2 Network Lifetime

In this experiment, we simulated the three protocols to observe how they affected the 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. We define network life time as the time taken for 50% of the nodes to die. Figure 8 shows the average network lifetime for each protocol as network size is varied.

We can see an expected decline in all three graphs as more nodes are introduced in the network. However GWMAC clearly outperforms LMAC and S-MAC. The latter two are decentralised and distributed in nature and this is the main cause of their underperformance, especially in larger networks. More nodes competing for the same medium during the contention phase results in an increase in the number of collisions even with carrier sense. Consequently, many nodes are prevented from getting access to the medium but at the same time are depleted of their energy whilst contending instead of transmitting data itself. This characteristic tends to shorten the network lifetime considerably in comparison to GWMAC where there is absolutely no contention. The results of this experiment suggest that on average GWMAC increases the network life time by at least 63% more than LMAC and S-MAC.

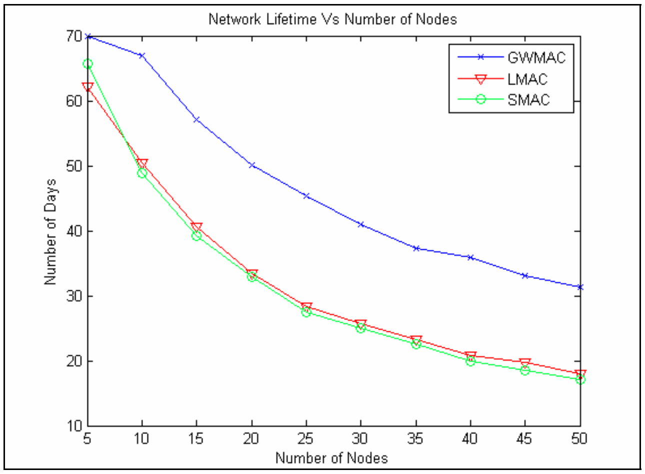

Figure 8. Effect of each MAC protocol on the network lifetime

5.3 Data Collected Over Network Lifetime

In this experiment, we simulated the three protocols to observe how much data the sink (anchor) nodes gathered over the network life time. The results of simulations are shown in Figure 9.

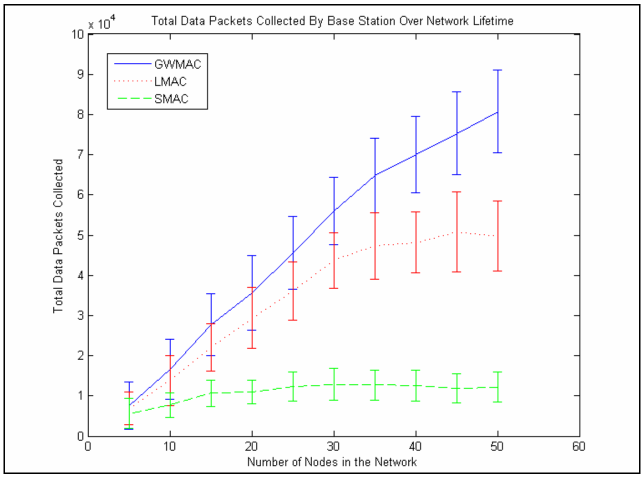

Figure 9. Data Collected By Nodes over Network Lifetime

The plot shows the mean number of data packets collected over the network lifetime using each protocol. It can be seen that the three protocols perform similarly for extremely small sized networks involving a handful of nodes. However, as number of nodes in the network begin to increase, the distinction between the three graphs becomes clearer where GWMAC starts outperforming LMAC and SMAC. The shorter network lifetime of LMAC and S-MAC ensues that less data is collected at the sink. The S-MAC network suffers most because of two reasons. Firstly, there is no presence of a slot (listen period) controller like the receiving node in LMAC or the transmitting node in GWMAC. Secondly, most S-MAC nodes tend to follow the same listen-sleep schedule. These two factors combine to create a bottle neck effect where intended receivers (relaying nodes) also compete for the medium to transmit their own data. This results in less data being transmitted at a high cost of collisions. The effect is emphasised when the number of data packets collected at the sink fail to increase as the number of nodes in the network are increased beyond 25.

5.4 Energy Used during Network Discovery

In this part of the experiment, we varied the number of nodes in the network from 5 to 50 to analyse how the self-organisation/network discovery phase differ in terms of energy consumption for each protocol. 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. The results of this simulation can be seen in Figure 10. The plot shows both LMAC and SMAC consume similar amounts of energy as the schedule choosing schemes adapted by the individual nodes are very similar. The nodes running GWMAC, on the other hand, choose a schedule decided by the central base station. The plot also suggests that all three protocols consume similar amounts of energy in networks of smaller size. However, as the size of the network goes beyond 15 nodes, GWMAC consistently consumes less energy. It can also be seen that the GWMAC energy consumption shares a more linear relationship with the number of nodes in the network whereas LMAC and S-MAC energy consumption increases exponentially with the size of the network. Again, this is attributed is due to the decentralised nature of latter two. LMAC nodes transmit their schedules to each other constantly until it is finalised and 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 this procedure increases with network size and hence consumes more energy.

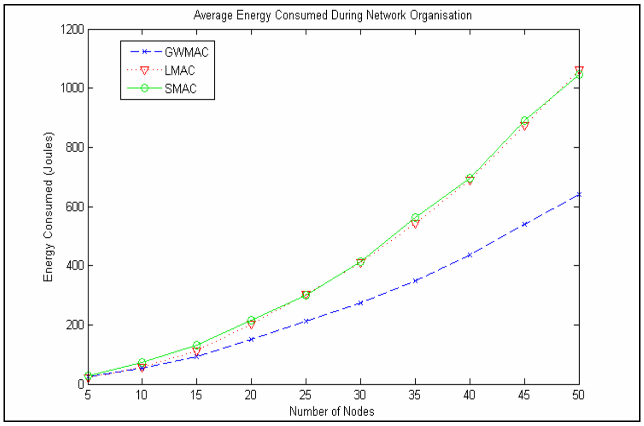

Figure 10. Energy consumed during network setup phase.

6. CONCLUSION

Designing a WSN for real deployment is not an easy challenge. It requires one to consider not only the energy constraints but also the host environment, communication reliability and the needs of the application end-user. In this paper we presented GWMAC, a centralised TDMA-based MAC protocol that is designed specifically for a network where topology is expected to change frequently and communication is unreliable.

By means of simulations we compared GWMAC against decentralised, hybrid (contention and schedule based) protocols such as S-MAC and LMAC in three different experiments. The first experiment suggested GWMAC can increase network lifetime by at least 63%. The second experiment demonstrated that lack of any form of contention between nodes and the assignment of regular transmission slots in GWMAC ensures the base station is able to collect more data packets from the network over its already extended life time. The results of the third experiment show that GWMAC consumes much less energy to organise the network. Energy required in setting up the network increases linearly with the number of nodes compared to the exponential increase with LMAC and S-MAC.

In general, schedule based MACs show higher energy savings than contention based MACs. GWMAC is much better suited for networks with unreliable communication and varying topology and with applications that are not latency sensitive but at the same time require high delivery guarantees.

7. ACKNOWLEDGMENTS

Our thanks to the GlacsWeb team in the School of Geography: Prof. J.K Hart and Kathryn Rose. Also a special thanks to Gro and Inge Melkevol for their logistical help in Norway. This project was funded by the EPSRC UK.

8. REFERENCES

[1] Martinez, K., Hart, J. and Ong, R., Environmental Sensor Networks. *IEEE Computer Vol. 37, No. 8*, (2004), 50-56.

[2] eyes.eu.org. Energy efficient sensor networks, (2007).

[3] Van Hoesel, L.F.W. and Havinga, P.J.M., Design Aspects of An Energy-Efficient, Lightweight Medium Access Control Protocol for Wireless Sensor Networks. *Technical Report TR-CTIT-06-47 Centre for Telematics and Information Technology, University of Twente, Enschede*, (2006), ISSN:1381-3625.

[4] Kleinrock L. and Tobagi F., Packet switching in radio channels: Part I - carrier sense multiple access modes and their throughput delay characteristics, Part II - the hidden terminal problem in carrier sense multiple access and the busy-tone solution. *IEEE Transactions on Communications, vol. 23, no. 12*, (Dec 1975),1400–1433.

[5] Ye, W. Heidemann, J. and Estrin, D. Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks, *IEEE/ACM Transaction on Networking*, (June 2004).

[6] Dam, T. V. and Langendoen, K. An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks, SenSys'03, *Los Angeles*, (Nov 2003), 171-180.

[7] Padhy, P., Dash, R. K., Martinez, K. and Jennings, N. R. A utility-based sensing and communication model for a glacial sensor network. *In Proceedings of 5th Int. Conf. on Autonomous Agents and Multi-Agent Systems*, (2006), 1353-1360.

[8] Tseng Yu-Chee, Ni Sze-Yao, Chen Yuh-Shyan, and Sheu Jang-Ping, The broadcast storm problem in a mobile ad hoc network. *Wireless Networks*, No 8, (2002), 153–167, ISSN:1022-0038.