A Utility-Based Adaptive Sensing and Multihop Communication Protocol for Wireless Sensor Networks

PARITOSH PADHY, RAJDEEP K. DASH, KIRK MARTINEZ,
and NICHOLAS R. JENNINGS
University of Southampton

This article reports on the development of a utility-based mechanism for managing sensing and communication in cooperative multisensor networks. The specific application on which we illustrate our mechanism is that of GlacsWeb. This is a deployed system that uses battery-powered sensors to collect environmental data related to glaciers which it transmits back to a base station so that it can be made available world-wide to researchers. In this context, we first develop a sensing protocol in which each sensor locally adjusts its sensing rate based on the value of the data it believes it will observe. The sensors employ a Bayesian linear model to decide their sampling rate and exploit the properties of the Kullback-Leibler divergence to place an appropriate value on the data. Then, we detail a communication protocol that finds optimal routes for relaying this data back to the base station based on the cost of communicating it (derived from the opportunity cost of using the battery power for relaying data). Finally, we empirically evaluate our protocol 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, we demonstrate that the efficiency gains of our new protocol, over the currently implemented method over a 6 month period, are 78%, 133%, 100%, and 93%, respectively. Furthermore, we show that our system 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.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing protocols*; H.1.1 [**Models and Principles**]: Systems and Information Theory—*Value of information*; J.2 [**Computer Applications**]: Physical Sciences and Engineering—*Earth and atmospheric sciences*

General Terms: Algorithms, Performance

This research was funded by GlacsWeb and ALADDIN (Autonomous Learning Agents for Decentralised Data and Information Systems) projects. GlacsWeb is funded by EPSRC (Engineering and Physical Research Council) and ALADDIN is jointly funded by a BAE Systems and EPSRC strategic partnership (EP/C548051/1).

Authors' addresses: P. Padhy, R. K. Dash, K. Martinez, and N. R. Jennings, School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK; email: {pp04r, rkd, km, nrj}@ecs.soton.ac.uk.

DOI 10.1145/1754414.1754423 http://doi.acm.org/10.1145/1754414.1754423

Additional Key Words and Phrases: Adaptive sampling, cost of communication, data gathering, dynamic environment, energy efficiency, network lifetime, sensor networks

ACM Reference Format:

Padhy, P., Dash, R. K., Martinez, K., and Jennings, N. R. 2010. A utility-based adaptive sensing and multihop communication protocol for wireless sensor networks. ACM Trans. Sensor Netw. 6, 3, Article 27 (June 2010), 39 pages.
DOI = 10.1145/1754414.1754423 http://doi.acm.org/10.1145/1754414.1754423

1. INTRODUCTION

Sensor networks are being deployed in a wide variety of application areas and, in particular, they have recently been used for a number of environmental monitoring applications [Martinez et al. 2004; Cerpa et al. 2001; Delin et al. 2003; Roure 2005; Lorincz et al. 2004; Steere et al. 2000]. Moreover, there is an increasing interest in conceptualizing these networks as a collection of interacting, autonomous agents and then adopting techniques from the field of multiagent systems [Lesser et al. 2003; Rogers et al. 2009] in order to control them in a decentralized fashion. In this vein, we consider a particular sensor network, GlacsWeb [Martinez et al. 2004], that we have deployed in the Briksdalsbreen glacier in Norway, and examine how it can be modeled as a (cooperative) multiagent system. In this case, the two main tasks performed by the nodes (agents) are gathering data from the environment and communicating it to a central sink node (i.e., an agent that harvests data from all other agents). In general, the agents work towards the predefined system goal of maximizing data collection (hence the cooperative nature of the system). However, they are invariably constrained in at least one of the following dimensions: their available power, their communication bandwidth, their memory storage, and/or their processing capability. Of these, power is the most important in this context since it is required for everything else. Thus, it directly influences the life-span of the agents and, hence, that of the system as a whole. Given this, we focus on developing an energy-aware sensing and communication protocol for the GlacsWeb network which aims at maximizing the information gathered about the glacial environment. Nevertheless, the solution we develop is more broadly applicable; specifically, it is suitable for any networks that have nodes with limited power supply.

In more detail, the purpose of the GlacsWeb sensor network is to monitor subglacial behavior in order to understand climatic change. Figure 1 shows GlacsWeb's central base station that is located on top of the glacier and Figure 2 shows a typical GlacsWeb node on which the agents run. In the current system, each individual node senses its own data and then communicates it directly to this sink node in a single hop. As such, the system's communication protocol is energy inefficient since it lacks the energy savings that a multihop approach would provide [Woo et al. 2003] (i.e., one in which agents relay data for one another). Furthermore, at present, sensing in GlacsWeb is carried out at a predetermined constant rate which is insensitive 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

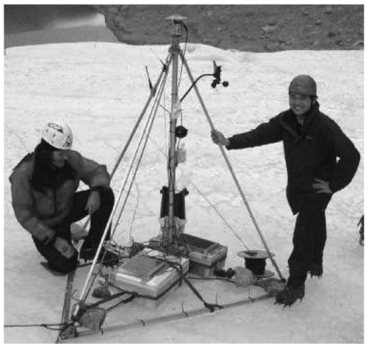


Fig. 1. GlacsWeb base station.

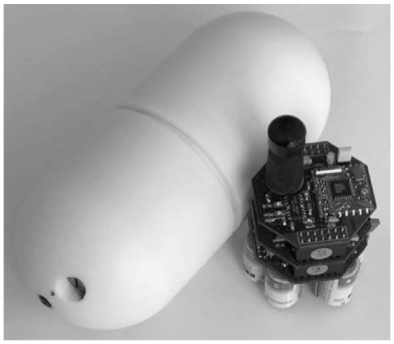


Fig. 2. A GlacsWeb sensor probe.

varying environment is less than what could be gained in a more dynamic situation.

Against this background, this article 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, we advance the state-of-the-art in the following ways.

- We develop a novel, decentralized 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 regresssion and the Kullback-Liebler divergence [Kullback and Leibler 1951] which gives it a sound information-theoretic foundation.
- We devise a new *multihop routing* protocol that finds the least expensive cost route from an agent to the center. 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 for sensing instead of relaying).

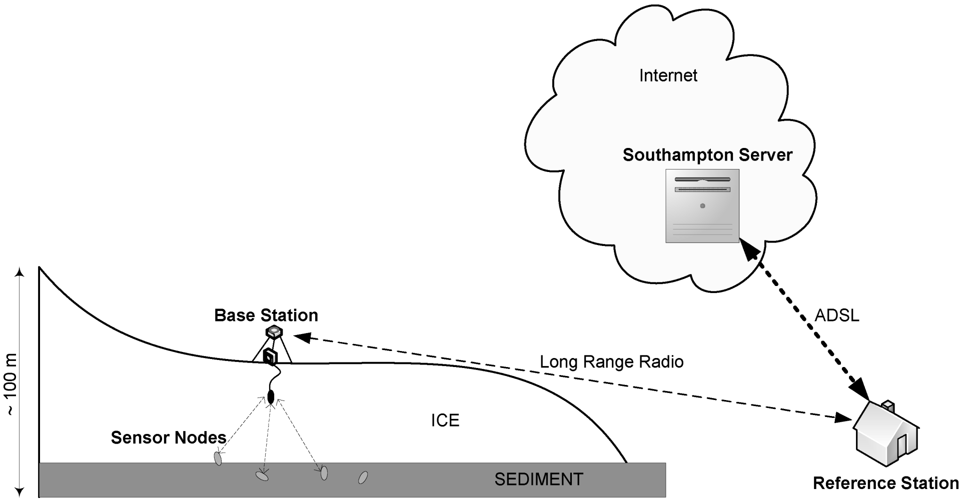


Fig. 3. Architecture of the GlacsWeb network. The system is composed of sensor nodes embedded in the ice and the subglacial sediment to monitor data and transmit it to the base station positioned on the surface of the ice. The base station in turn accumulates additional information about the weather 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.

—We empirically evaluate 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. We show that against the latter three, USAC provides a significantly higher gain in information, whilst reducing power consumption. Furthermore, it compares favorably 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 remainder of this article is organized as follows. Section 2 provides the basic background on GlacsWeb and a discussion of related work in the area of sensor networks modeled as multiagent systems. We then detail the sensing and the routing aspect of USAC in Section 3. Section 4 discusses the computational feasibility of the GlacsWeb node and Section 5 discusses the routing protocol. The protocol is then empirically evaluated in Section 6. We conclude in Section 7.

2. BACKGROUND AND RELATED WORK

GlacsWeb is a deployed pervasive sensor network that directly monitors subglacial movement to determine how it is affected by climatic changes. In order to do this, it uses a network of subglacial nodes that are placed at different locations inside the glacier as shown in Figure 3. The protocol currently followed by GlacsWeb is a simple one in which the nodes sample the glacier every four hours and then transmit these readings directly to the base station located at the top of the glacier daily. The on-board sensors measure temperature, resistivity, case stress, tilt angle, and water pressure. At the time of writing, there are twenty nodes in the glacier with an additional batch of ten deployed every summer.[1] However, as a result of the hostile environment (due to the low temperatures, the strain resulting from moving ice and the englacial water bodies) and the power-hungry protocol, there is a high probe failure rate of around 50% per year. This provides the motivation for the research carried out in this article.

However, as previously stated, this protocol and the techniques we use within it are more broadly applicable. We will now present the most salient research in the two fields that are most closely related with work presented in this article. The first concerns that of adaptive sensing and the second deals with routing protocols.

2.1 Adaptive Sensing

In order to produce energy-aware protocols, a number of researchers have focused on intelligent adaptive sensing [Deshpande et al. 2005; Willett et al. 2004; Jain and Chang 2004; Popa et al. 2005]. Such work has demonstrated how a sensor's lifetime can be extended by sensing only during the most informative situations. The key differences between the various approaches is how they select these most informative situations. We now discuss three different adaptive sensing protocols which are representative of the main approaches used for sensing intelligently.

The BBQ model [Deshpande et al. 2005] is an adaptive sensing protocol which is similar to our sensing protocol, despite the fact that it is developed for a query-based sensor network (one where data can be classified and the base station propagates interests among the nodes to gather it). In this model, Deshpande et al. [2005] study how to best sample from the environment given correlations (that can be learnt using a Kalman Filter (KF) approach) between the observation vectors of the different sensors. The sensor network architecture incorporates statistical models of real-world processes which 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 then provides a framework for optimizing the acquisition of sensor readings. Specifically, nodes are used to acquire data only when the model itself is not sufficiently rich to answer the query with acceptable confidence. This yields an optimal sensing schedule for each of the nodes. However, the nodes act under a centralized control regime within this architecture and it is thereby inadequate for a distributed environment where communication between nodes incurs a substantial cost and the topology is constantly changing.

[1] The glacier has subsequently melted, as predicted by the researchers (BBC news reported this and a real player recording of the news item can be viewed at http://news.bbc.co.uk/nolavconsole/ifs_news/hi/newsid_6170000/newsid_6171300/nb_rm_6171392.stm). A new glacier in Iceland is currently being surveyed for further deployment.

The second approach, also a statistical protocol for adaptive sampling, is discussed by Jain and Chang [2004]. This is one of the very few adaptive sampling techniques where each node adapts to the streaming-data characteristics. Here, in contrast to the BBQ model, the nodes autonomously decide their sampling rate within a given range using the KF estimation error. When the desired sampling rate violates the range, a new sampling rate is requested from the base station. Here, the base station allocates new sampling rates under the constraint of available resources such that the KF estimation error over all the active streaming nodes is minimized. Note that in this case the sensors transmit all their sensed data to the base station, however, this is undesirable in our context because our objective is to minimize transmission, in addition to minimizing the sensing activity for the energy-starved nodes.

A third approach is that of Backcasting [Willett et al. 2004]. This is based on a hierarchical structure, such as LEACH [Heinzelman et al. 2000] or PEGASIS [Lindsey and Raghavendra 2002], in which small subsets of sensor nodes communicate their observations to a fusion center. This center, in turn, provides an initial estimate of the environment being sensed. Based on this coarse estimate, the center then determines which regions of the field may contain boundaries or sharply varying behavior 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 center. The key idea in this approach 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 center. However, this hierarchical approach again suffers from a high computational load for the fusion centers, a high communication cost for coordination sensing actions, and it is not robust to a constantly changing topology.

The focus in all of the aforesaid systems is on information fusion/data aggregation and, in general, they do not consider how to communicate efficiently to the base station. Given this, we next provide an overview of research that does address the problem of routing data to the base station.

2.2 Routing Protocols

A number of routing protocols have been investigated to enhance the performance of wireless sensor networks and these are generally classified under one of three categories, namely, *data-centric*, *hierarchical*, or *location-based*. For the purposes of this research, we can discount the last category as most protocols falling under it are not energy aware. Therefore, we focus the remainder of this subsection on the other two.

Traditionally, nodes were queried based on their preassigned 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 the data), in order to eliminate redundant transmissions. One of the most common techniques employed in data-centric protocols is flooding [Downey and Cardell-Oliver 2004] where each agent receives an item of data and then repeats it by broadcasting unless the destination of the packet is the agent itself. Although this reactive technique does not require costly topology maintenance or complex route discovery algorithms, it causes the undesired effects of data implosion and/or data overlap. In an attempt to refine this methodology, the SPIN [Heinzelman et al. 1999] protocol considers data negotiation between agents to eliminate redundant data. It adopts a publish-subscribe approach where agent nodes operate efficiently and conserve energy by only sending metadata describing the data, instead of sending the actual data. Thus, this model is useful for those agents interested in the data advertised and is an effective protocol to minimize energy spent in consumption until the actual data is transmitted. However, it fails to place a limit on the energy consumed in wasted advertisements (i.e., for which there are no subscribers for the data).

A conceptually converse approach is directed diffusion [Intanagonwiwat et al. 2000] which eliminates the unnecessary energy-consuming operations of routing by diffusing data through agent nodes using a naming scheme for data. It suggests the use of *attribute-value* pairs (such as objects, intervals, durations, and geographical areas) for data queries which are disseminated through the network as *interests*. This dissemination sets up several initial *gradients* (reply links to neighbors from which interests were received). Each such gradient is characterized by the data rate, duration, and expiration time derived from the received interests fields. The network then uses these properties to *reinforce* one or a small number of these paths for data to flow through. This on-demand data querying approach does not require maintaining a global network topology and agent nodes have the advantage of carrying out data aggregation and caching of interests, in addition to sensing. However, it is not employable in GlacsWeb because it does not continuously deliver data to the base station, a prerequisite set by the glaciologists. In addition, there is still a significant communication overhead associated with the queries.

We now turn to hierarchical protocols that aim to cluster the agent nodes so that cluster heads can do some aggregation and reduction of data in order to save energy. Specifically, LEACH [Heinzelman et al. 2000], and its variants, is one of the first hierarchical routing algorithms in sensor networks. In its basic form, it forms clusters of agent nodes based on the received signal strength and uses local cluster heads as direct routers to the base station. This saves energy since only the cluster heads carry out long-distance transmissions. To balance the energy dissipation of nodes, cluster heads change randomly over time. However, the use of single-hop routing is not applicable to networks deployed over large regions (such as a glacier) because nodes further away might be physically unable to transmit to the base station. Moreover, this dynamic clustering introduces extra overhead in organizing the clusters and electing a head, both of which diminish the gain in energy consumption.

To sum up, our research substantially differs from all the aforementioned approaches in that it reuses the value derived from the sensing protocol in the communication protocol, thereby intertwining these two critical aspects. The proactive task of amalgamating sensing and communication has also been proposed through the RMR/IDEALS protocol in Merrett et al. [2007] and Byers and Nasser [2000] where each node decides its individual network involvement based on its own energy resources and the information contained in each data packet. However, these approaches are founded on a rigid residual energy model in which the cost of communication remains constant. Our protocol, on the other hand, is based on the concept of utility that provides a measure of the usefulness and scarcity of the resources and can sense and communicate much better. Finally, the concept of utility in sensor networks has also been addressed in Dash et al. [2005] and Rogers et al. [2005] for selfish nodes that are owned by different stake holders. Specifically, these approaches show that it is possible to develop a distributed mechanism in which the goals of the individual selfish components are aligned with the goals of the overall system. In contrast, our work is based on a system where there is only one stake holder (the University of Southampton) and we require consistent interactions between agent nodes to exchange information regarding communication to the base station. Hence we deem selfishness in these nodes as an unnecessary attribute and focus primarily on developing cooperative agent nodes which we believe will lead to superior performance in this context.

3. 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. In this section, we therefore discuss a generic framework for this decision-making process within the agent, and explain how this can then be used in relation to the specifics of 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 we use in this case is derived from *information theory* because this enables us 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.

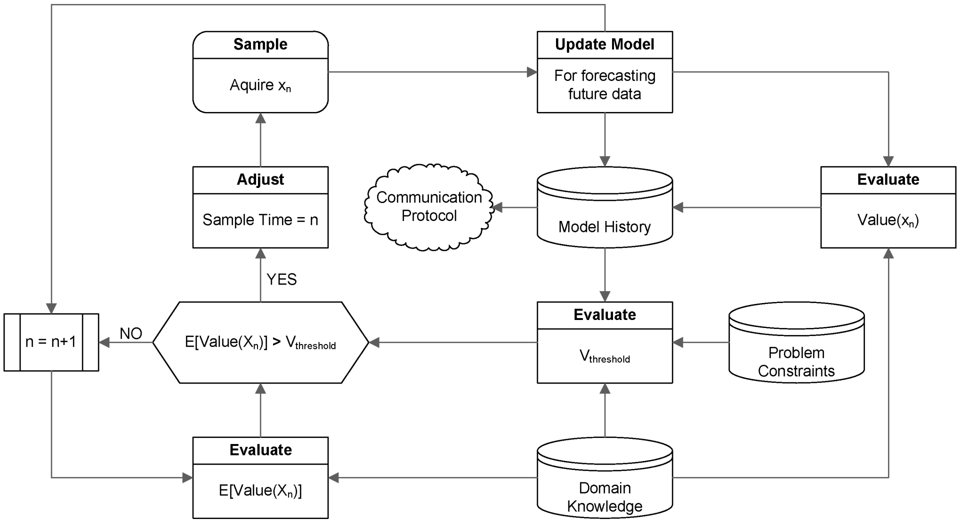


Fig. 4. Decision making process for sensing within an agent. The domain knowledge part may not be present dependent on the application. When a sample is acquired, it is used to update the existing model for forecasting future data. The updated model is then in turn used to place a value on the acquired 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.

In more detail, the sampling protocol we propose can be described by Figure 4. 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.

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[2] against a threshold value. This threshold value is important since without it, the sampling would occur at each available opportunity.[3] 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

[2] The time-step is determined by the maximum sampling frequency available to the sensor node.

[3] This is the information maximizing sampling rate when no constraints are present.

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) 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 4, the forecasting method chosen will impact on the value of the data being measured by the sensor. In this work, we focus on a Bayesian model for updating the model held by the sensor. However, this model can be generalized to Gaussian processes as explained by Williams [1998]. In our particular case, this generalization is not required since we already have a fair amount of prior knowledge about the data from the glaciers. Thus, the additional computational load of Gaussian process regression is not warranted in this case. In the next section, we give details of the framework of Bayesian regression which is used in this article.

3.1 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 our framework since the agent has to value the data it measures as well as predict the value of future data. We present the Bayesian linear regression analysis framework, which can be extended in a straightforward manner to nonlinear 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

$$x_i = \boldsymbol{t}^{\mathrm{T}} \boldsymbol{w} + \epsilon, \tag{1}$$

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.

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

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 characterize 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 minimize the Euclidean distance between the projected data and the measured data. That is, choose $\boldsymbol{w}^*$ such that

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \frac{1}{n} \sum_{i=1}^{n} \left(x_i - \boldsymbol{t}_i^{\mathrm{T}} \boldsymbol{w}\right)^2 . \tag{3}$$

By applying the first-order condition on Eq. (3), it can be calculated that[4] (refer to Appendix A for the proof)

$$\boldsymbol{w}^* = (T^{\mathrm{T}} T)^{-1} T^{\mathrm{T}} \boldsymbol{x}. \tag{4}$$

The preceding 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 turn to Bayes' theorem. In essence, Bayes' rule assigns a probability that an event E_1 occurs, given we have observed an event E_2, as

$$P(E_1 \mid E_2) = \frac{P(E_2 \mid E_1) P(E_1)}{P(E_2)}. \tag{5}$$

In our case, we want to predict the sensor's next measurement (the E_1) based on all our observations so far (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.

$$p(x_{n+1} \mid \boldsymbol{t}_{n+1}, T, \boldsymbol{x}) = \int p(x_{n+1} \mid \boldsymbol{t}_{n+1}, \boldsymbol{w}) p(\boldsymbol{w} \mid T, \boldsymbol{x}) \, d\boldsymbol{w} \tag{6}$$

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 the prior distribution $p(\boldsymbol{w}) \sim \mathcal{N}(0, \Sigma_p)$ and marginalizing out $p(\boldsymbol{x} \mid T)$, the following can be derived (the proof of this is in Appendix B).

$$\begin{aligned} p(\boldsymbol{w} \mid T, \boldsymbol{x}) &= \frac{p(\boldsymbol{x} \mid T, \boldsymbol{w}) p(\boldsymbol{w})}{p(\boldsymbol{x} \mid T)} \\ &\propto p(\boldsymbol{x} \mid T, \boldsymbol{w}) p(\boldsymbol{w}) \\ &\sim \mathcal{N}(\overline{\boldsymbol{w}}, A^{-1}) \end{aligned} \tag{7}$$

[4] A number of methods exist in order to reduce the computation load of finding the optimal weights. However, in this article, we do not consider such techniques since our focus is on placing a value on the sensed data, rather than optimizing the computations when calculating such a value.

where $A = \sigma^{-2}TT^{\mathrm{T}} + \Sigma_p^{-1}$ and $\overline{\boldsymbol{w}} = \sigma^{-2}A^{-1}T\boldsymbol{x}$. Combining the result of Eq. (6) with that of Eq. (7), we obtain the following.

$$\begin{aligned} p(x_{n+1} \mid \boldsymbol{t}_{n+1}, T, \boldsymbol{x}) &= \int \boldsymbol{t}_{n+1}^{\mathrm{T}} \boldsymbol{w} p(\boldsymbol{w} \mid T, \boldsymbol{x}) d\boldsymbol{w} \\ &= \mathcal{N}\left(\frac{1}{\sigma^2}\boldsymbol{t}_{n+1}^{\mathrm{T}} A^{-1} T\boldsymbol{x}, \boldsymbol{t}_{n+1}^{\mathrm{T}} A^{-1}\boldsymbol{t}_{n+1}\right) \end{aligned} \tag{8}$$

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

$$p(\boldsymbol{x}|T) = \int p(\boldsymbol{x}|T, \boldsymbol{w})p(\boldsymbol{w})\, d\boldsymbol{w}. \tag{9}$$

For the prior used in this article, it can be shown that the log of the previous equation reduces to [Rasmussen and Williams 2005]

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

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

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 nonlinear 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, our model is given by

$$x_i = \phi(\boldsymbol{t})^{\mathrm{T}}\boldsymbol{w} + \epsilon \tag{10}$$

in contrast to the linear regression model given by Eq. (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

$$\begin{aligned} p(x_{n+1} \mid \boldsymbol{t}_{n+1}, T, \boldsymbol{x}) &= \int \phi(\boldsymbol{t}_{n+1})^{\mathrm{T}} \boldsymbol{w} p(\boldsymbol{w} \mid T, \boldsymbol{x})\, d\boldsymbol{w} \\ &= \mathcal{N}\left(\frac{1}{\sigma^2}\phi(\boldsymbol{t}_{n+1})^{\mathrm{T}} A^{-1}\phi(T)\boldsymbol{x}, \phi(\boldsymbol{t}_{n+1})^{\mathrm{T}} A^{-1}\phi(\boldsymbol{t}_{n+1})\right). \end{aligned} \tag{11}$$

An example of such a kernel regression is illustrated in Figure 5 where the decrease in variance of the model as the number of points sampled increases is evident.

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 provide us with a principled way of generalizing the Bayesian linear regression model so as to generate sets of (potentially infinite)

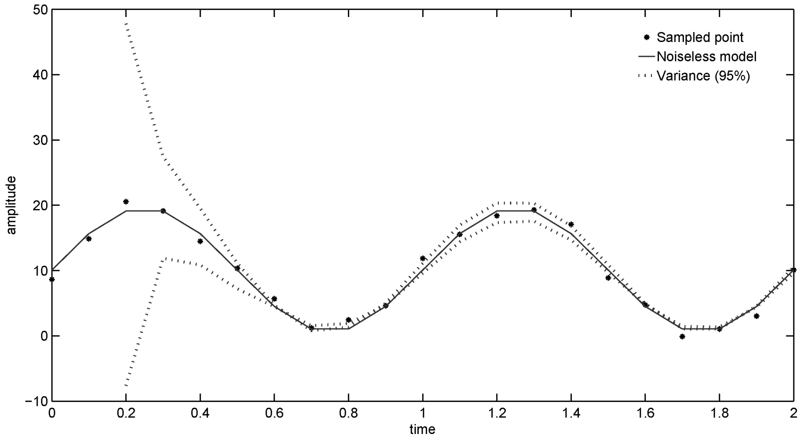


Fig. 5. Example of Bayesian kernel regression with a simple sinusoidal kernel.

kernel functions given more metalevel 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 summarize, this section has provided us with an overview of how to update data models in order to forecast future data. In the next section, we explain how to use these data models in order to value the data that is observed.

3.2 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

$$\delta_{KL}(f_1, f_2) = \int_{-\infty}^{+\infty} f_1(y) \log \frac{f_1(y)}{f_2(y)}. \tag{12}$$

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 Eq. (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

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

where $A_{n-1} = \sigma^{-2} T_{n-1} T_{n-1}^{\mathrm{T}} + \Sigma_p^{-1}$. Combining this with the posterior given by Eq. (8), provides us with the value of the information contained within x_n, given we have observed the vector $\boldsymbol{x}_{n-1}$ as

$$V(x_n|\boldsymbol{x}_{n-1}) = \int_{-\infty}^{+\infty} p(x_{n+1} \mid \boldsymbol{t}_{n+1}, T_{n-1}, \boldsymbol{x}_{n-1}) \log \frac{p(x_{n+1} \mid \boldsymbol{t}_{n+1}, T_{n-1}, \boldsymbol{x}_{n-1})}{p(x_{n+1} \mid \boldsymbol{t}_{n+1}, T, \boldsymbol{x})} dx_{n+1}. \tag{13}$$

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

$$\delta_{KL}(f_1, f_2) = \frac{1}{2}\left(\log\left(\frac{\det \Sigma_1}{\det \Sigma_0}\right) + tr\big(\Sigma_1^{-1}\Sigma_0\big) + (\mu_1 - \mu_0)^{\mathrm{T}} \Sigma_1^{-1}(\mu_1 - \mu_0) - N\right). \tag{14}$$

Thus, it is straightforward to calculate Eq. (13) using Eq. (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 3.3.

So far, we have 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[5], 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 our framework (specifically within the box labeled "domain knowledge" in Figure 4) 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.[6] 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

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

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

from the classifier, an expected value of the data can be calculated for these sensor networks.

3.3 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 characterized 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

$$p\big(x_n \mid t_n, w_1, \ldots, w_K, \sigma_1^2, \ldots \sigma_K^2\big) \sim \begin{cases} \mathcal{N}\big(t_n w_1, \sigma_1^2\big) & \text{if } 0 < n < \beta_1, \\ \quad\vdots & \quad\vdots \\ \mathcal{N}\big(t_n w_K, \sigma_K^2\big) & \text{if } \beta_{K-1} < n < \beta_K, \end{cases} \tag{15}$$

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 Eq. (15), it can therefore be observed that there are three major aspects of the data which are unknown:

(1) the model parameters (i.e., $w_1, \ldots, w_K$);
(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 preceding model, 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, our 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

$$p(\beta_k = n \mid S) = \frac{p(\beta_k = n) p(\boldsymbol{x}_1^{n-1} \mid \boldsymbol{t}_1^{n-1}) p(\boldsymbol{x}_n^N \mid \boldsymbol{t}_n^N)}{\sum_{\beta_k=3}^{N-1} p(\beta_k = n) p(\boldsymbol{x}_1^{n-1} \mid \boldsymbol{t}_1^{n-1}) p(\boldsymbol{x}_n^N \mid \boldsymbol{t}_n^N)}, \tag{16}$$

where $p(\beta_k = n)$ is the prior probability that the breakpoint occurs at n and S is the training set described in Section 3.1. 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 normalizing constant, $P(S|\exists\ \beta_k)$ is the probability of explaining the dataset given a phase change.

$$P(S|\exists\, \beta_k) = \sum_{\beta_k=3}^{N-1} p(\beta_k = n)p(\boldsymbol{x}_1^{n-1} \mid \boldsymbol{t}_1^{n-1})p(\boldsymbol{x}_n^N \mid \boldsymbol{t}_n^N) \tag{17}$$

Now in order to estimate whether a phase change has actually occurred, we divide Eq. (17) by Eq. (9) (which can be interpreted as the probability of explaining the dataset 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 Eqs. (13) and (14)). Notice that the KL method automatically places a very high value on data which signals a phase change (in Eq. (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

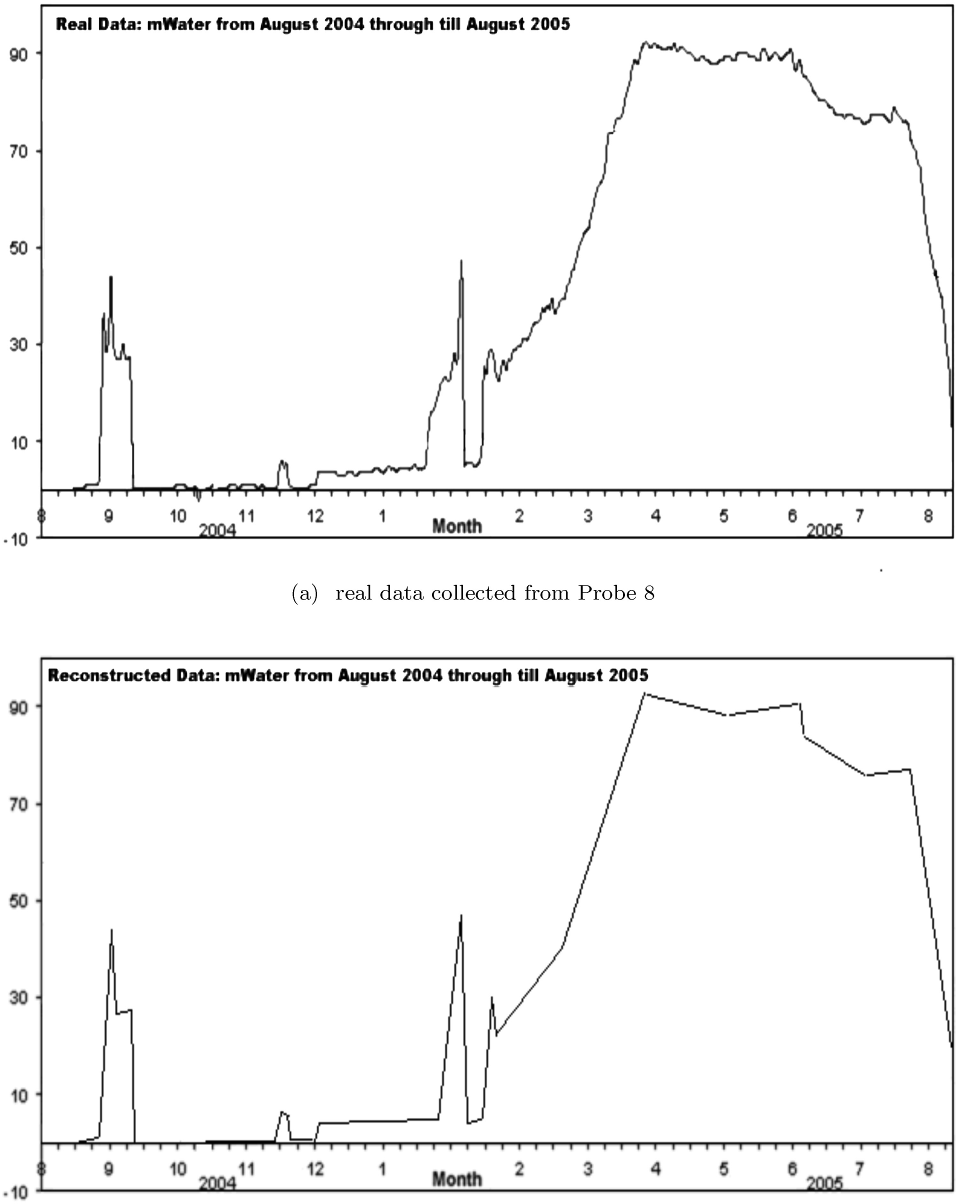
$$V_{threshold} = \frac{\sum_{n=3}^{i} V(x_n)}{i - n}, \tag{18}$$

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 to an unreasonably high value 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. Figure 6 illustrates how the data collected using our sensing algorithm compares against real data. We are aware that this algorithm (like any other discrete sampling algorithm) may miss out ephemeral events. However, it should be noted that events that appear ephemeral can also, in a number of cases, have early warning signs that could be used and valued appropriately with a particular valuation function (as determined by the experts on the phenomenon being studied). In the case of GlacsWeb, for example, the changes in pressure over time within the glacier indicate an imminent potential breakdown and thus the user could assign $V_{threshold}$ appropriately instead of taking the moving average.

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.

4. COMPUTATIONAL FEASIBILITY

As we are seeking to develop a solution that can be used in practice, it is important to consider the energy consumption and the time it takes to carry out the computations on the nodes. In more detail, a significant amount of computational load is assigned to each sensor node by the processes described in Section 3. Therefore it is important to determine that these loads can actually



(a) real data collected from Probe 8

(b) Probe 8's data as sampled by the sensing algorithm

Fig. 6. Comparison of real data vs. data sampled by the sensing algorithm.

Table I. PIC Microcontrollers 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	3.3	55mA	8MHz

be supported by the underlying hardware. Specifically, each GlacsWeb node is installed with a 1MHz, 8-bit PIC18F4320 microcontroller which is responsible for controlling the sensors and processing the data.

Now, the model used in our sensing protocol (Eqs. (17), (13), and (14)) performs operations that are dominated by matrix exponential calculations, that is, e^A where A is some $N \times N$ matrix. This expression is calculated as follows.

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

The preceding 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 previous expression from $k = 0$ to $k = 20$, which provides us with a very good estimate of the final value(s) upto 3 decimal places. Thus, if the matrix A were to be of a generous size, say 20×20, it would take approximately 193734 instruction cycles to evaluate the preceding expression. The PIC processor uses 4 clock cyles for every instruction cycle. Therefore, the time t taken to compute the previous expression would be

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

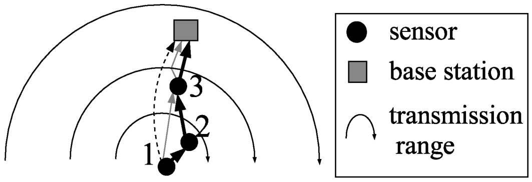
Furthermore, the PIC consumes 110μA across 2V and therefore the total energy, $E_{compute}$, required to calculate this matrix exponential is given by

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

Whilst $775ms$ may seem a long time for some real-time applications, latency is not an overriding factor within GlacsWeb, as long as nodes transmit their data to the base station once a day. Furthermore, energy saving is of much greater importance in this application and 170.5μJ is an excellent compromise for the time it takes to carry out the heavy computations.

Building on this, we went one step further to simulate the average time and energy needed to evaluate Eqs. (13), (14), and (17) using a Microchip compiler and our results show that the PIC microcontroller requires computation time between 15–20s, whilst consuming energy upto $4.27mJ$. This is again entirely acceptable for GlacsWeb, due to the extremely small amount of power consumption.

In more detail, Table I illustrates the different PIC microcontroller families and the trade-off between processor speed and current consumption. As can be seen, the PIC16 performs the calculation with the least amount of power, but does take a significant amount of time. At the other extreme, the PIC32 requires the least amount of time, but it does have a higher idle current. In this



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

Fig. 7. Three possible routes via which sensor 1 can transmit its data to the base station. The concentric semicircles 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.

context, we decided on a PIC18 microcontroller as a good compromise between idle and calculation current that is well-suited to the needs of GlacsWeb.

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

5. 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 center [Padhy et al. 2005]. However, as discussed in Section 2, 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 center via a single hop is more than the energy spent when the data is relayed via successive intermediaries to the center. In order to see this effect, consider the example shown in Figure 7. 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 $1 \rightarrow 2 \rightarrow 3 \rightarrow bs$. This approach is often known as topology control [Ramanathan and Rosales-Hain 2000; Wattenhofer et al. 2001; Pan et al. 2003] since it aims to control the topology of networks for the purpose of maintaining some global graph property (such as connectivity), while reducing energy consumption and/or interference that are strictly related to the nodes' transmitting range. Our communication protocol also takes advantage of the capability of GlacsWeb nodes to adjust tranmission power.

However, such an approach based solely on the readjusting 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.

We tackle these two problems 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 3.2) and the cost of transmitting the data. We next detail how to calculate the cost of communication, before going onto the algorithm used for the communication protocol in Section 5.2.

5.1 The Cost of Communication

The network is modeled as a multiagent 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 neighbors $n_i(pt_i^k) \subseteq \mathcal{I}$ 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

$$c_i^j(originate) = \frac{Et_i^j}{Et_i^j + E_i^{sense}} \times v_i^{sense}(t_n), \tag{19}$$

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

$$c_i^j(relay) = \frac{Et_i^j + E_i^{receive}}{Et_i^j + E_i^{sense}} \times v_i^{sense}(t_n), \tag{20}$$

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

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

We choose such a forecasting method since 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 is greater than 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.

5.2 The Communication Algorithm

The algorithm we use for the communication protocol is given in Figure 8. It consists of four main steps, namely:

(1) *Initialization*. 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 neighbors. For the purpose of simplification, the model assumes symmetric links between nodes. Flooding is used for network discovery. Here, the base station initiates the flooding by forwarding a message to all its neighbors, each successive node forwarding it once. As the flooding proceeds, nodes keep track of who they received messages from. 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). This is again application dependent. For GlacsWeb, this phase is executed once a day during summer months (when ice movement is more dynamic) and every three to four days during winter months.

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

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.
```

Fig. 8. The routing algorithm.

(3) *Update cost of transmission to base*. This step is required so as to find the minimum cost route from each agent to the center. In order to do so, agents in each power-level group successively transmit the cost of their least expensive route to the center. This is similar to Adaptive Transmission Power Control (ATPC) [Lin et al. 2006], where the nodes learn about the minimum power and link quality of other nodes in their neighborhood via a feedback mechanism. An important difference between ATPC and USAC

is that the latter finds the lowest cost path to the base station, whereas the ATPC works on a greedy optimization of the path. Furthermore, ATPC nodes adjust their power levels based on the white noise within the communication channel. USAC does not focus on this aspect of link quality as it is outside the scope of this article. However, it should be noted that it would be straightforward to implement this within the USAC protocol by incorporating a minimum signal to noise ratio for the nodes that would accept packets within the communication protocol. Another way in which unreliable links are handled is via blacklisting [Son et al. 2004]. Here, links of poor quality are deselected as and when they are discovered. USAC also adopts a similar policy whereby the power levels of nodes are adjusted based on historic communication (as opposed to setting these power levels at deployment).

(4) *Transmit data*. Having found the least expensive cost of transmission to the base station, the agent then decides whether or not to transmit its observed data.

Our routing algorithm could be considered a variant of distance-vector-based routing [Perkins and Bhagwat 1994], where the cost function is analogous to distance. However, our protocol differs considerably in that the cost function is not stand-alone and is strongly coupled with the value of the sensed data. Thus, 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 the next section where we empirically evaluate its performance against some benchmark protocols.

6. EMPIRICAL EVALUATION

In this section, we empirically evaluate the performance of USAC through a series of experiments. In particular, we compare it against four other alternative protocols (discussed shortly) in networks with varying topology, agent numbers, and degrees of dynamism in the environment. We also benchmark 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 network achieves a theoretical maximum lifetime. This is clearly impossible in practice, nevertheless, indicates how effective our strategy is in absolute terms.

6.1 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 realizes 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, an a priori model that best describes 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 our simulations we employed this 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 the end-to-end capability of forwardness proposed in Woo et al. [2003]. It is defined as the expected number of transmissions (including retransmissions) for a successful end-to-end data forwarding. Specifically, 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 $ETX(p) = \sum_{i=1}^{n} etx(l_i)$. The plot for this strategy is labelled *MINT*.

(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 maximized. 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. The plot for this strategy is labeled *OPTIMAL*.

It is obvious that adaptive sampling should 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. We also decided to use a simple version of the TDMA-based protocol developed in our group [Elsaify et al. 2007] that limits the communication of the nodes to one short window during the day. In doing so, this MAC-layer protocol helps eliminate the lower-level problem of overhearing and reduces the duty cycle of the nodes significantly.

In terms of measuring performance, we adopt the following definitions.

Definition 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 2. *Network Lifetime* is defined as the time taken for 50% of the initial number of agent nodes in the network to die.[7]

[7] A formal definition of network lifetime is not straightforward and depends on specifics of the application scenario in which the network is used. Our simulations suggested that failure of

Table II. Simulation Parameters

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
Packet Time	130ms (6 samples)
Transmission Ranges	14.14, 28.28, 42.426, 56.568, 70.71
Network Area	200m × 200m

We use a discrete even-based simulator developed in Java. 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. 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 II 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 our experience with the fielded system).

The radio propagation model is assumed to be symmetric in which the energy to transmit over a distance d is proportional to the square of this distance (*energy* $\propto d^2$). However, we are aware that 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 [Zhou et al. 2006; Zamalloa and Krishnamachari 2007]. Therefore, in order to make our simulations more realistic, we set the transmission link quality between any two nodes to 0.8. This value was based on the real experience of deploying our 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 9 for a specific example of this phenomena.

Finally, for statistical significance, we report average results and standard deviations of 100 simulations in each of the experiments carried out. The data used in our experiments is derived from segments of the data collected[8] by the fielded probes over the last three years. In our graphs we show the standard

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, our definition refers to the capability of the network to provide the services it was designed for (i.e., gather data from the nodes).

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

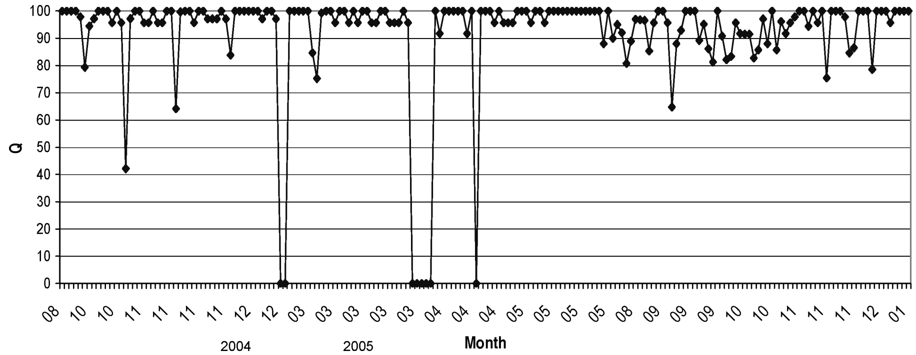
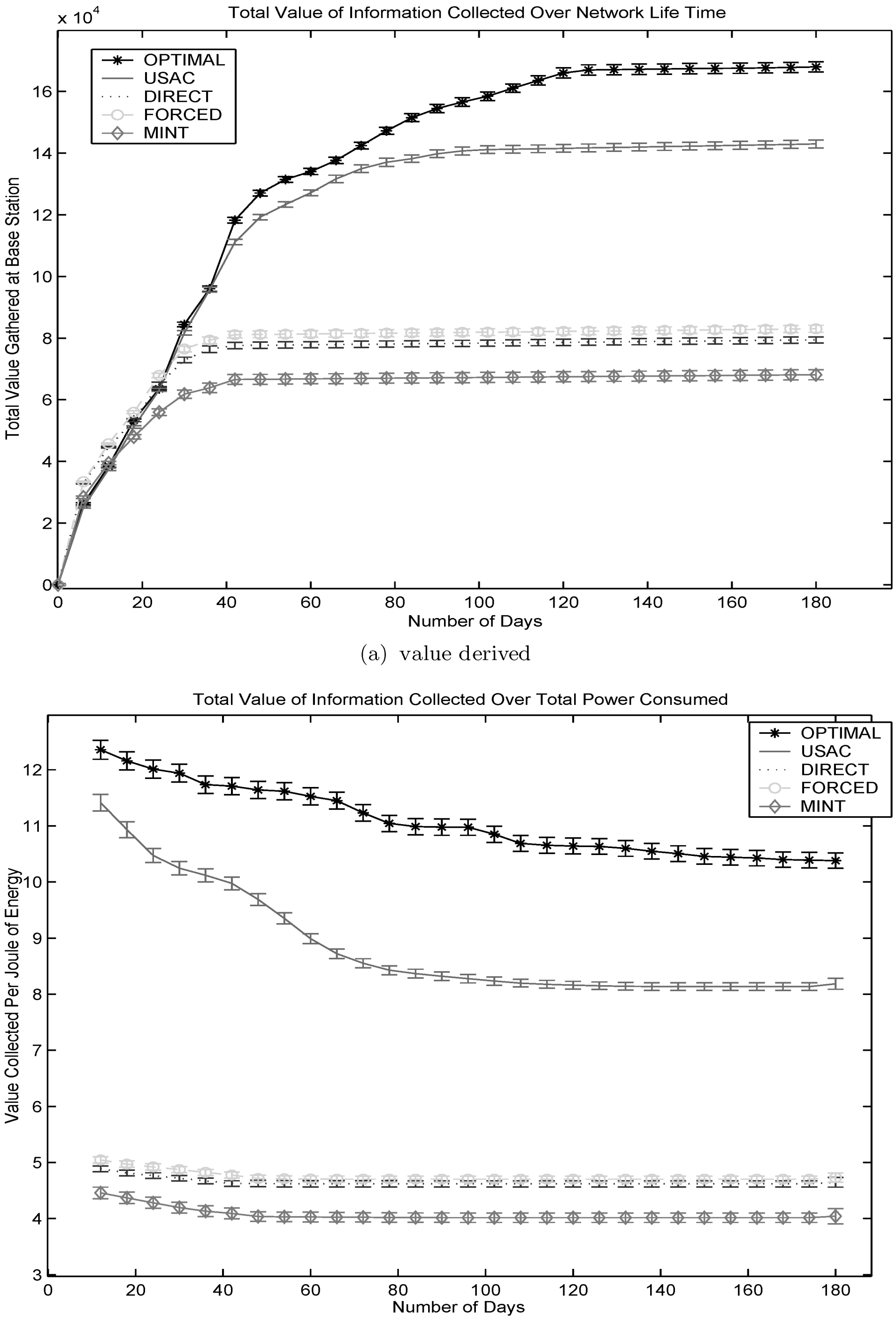


Fig. 9. Percentage of good probe packets received from Probe 8 over 16 months (10000 packets).

error of the mean, as well as the 95% confidence intervals. Thus, error bars in our plots are 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 +e. Since the simulation instances were conducted over a period of 6 months, the maximum lifetime of a network was capped at 183 days.

6.2 Network Topology (Static Nodes)

In this experiment, we carried out simulations for a fixed number of agents (10) randomly distributed around the center. 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 analyze 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 Figure 10. Both plots show the superiority of USAC over the other practical protocols. Specifically, Figure 10(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 the 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 a higher information gain. This is expected, as all samples sensed by the FORCED agents are transmitted, as opposed to the selective data transmitted in the other protocols. However, 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. In particular, MINT performs the worst in terms of network lifetime and information gain. This validates the theory that multihopping 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 with DIRECT. Therefore, a middle ground needs to sought which is provided in USAC. In particular, Figure 10(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 times more efficient than the others.



(a) value derived

(b) efficiency

Fig. 10. The total value of data gathered and total value of data gathered per joule over a 6-month period plotted against time (fixed topology).

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.

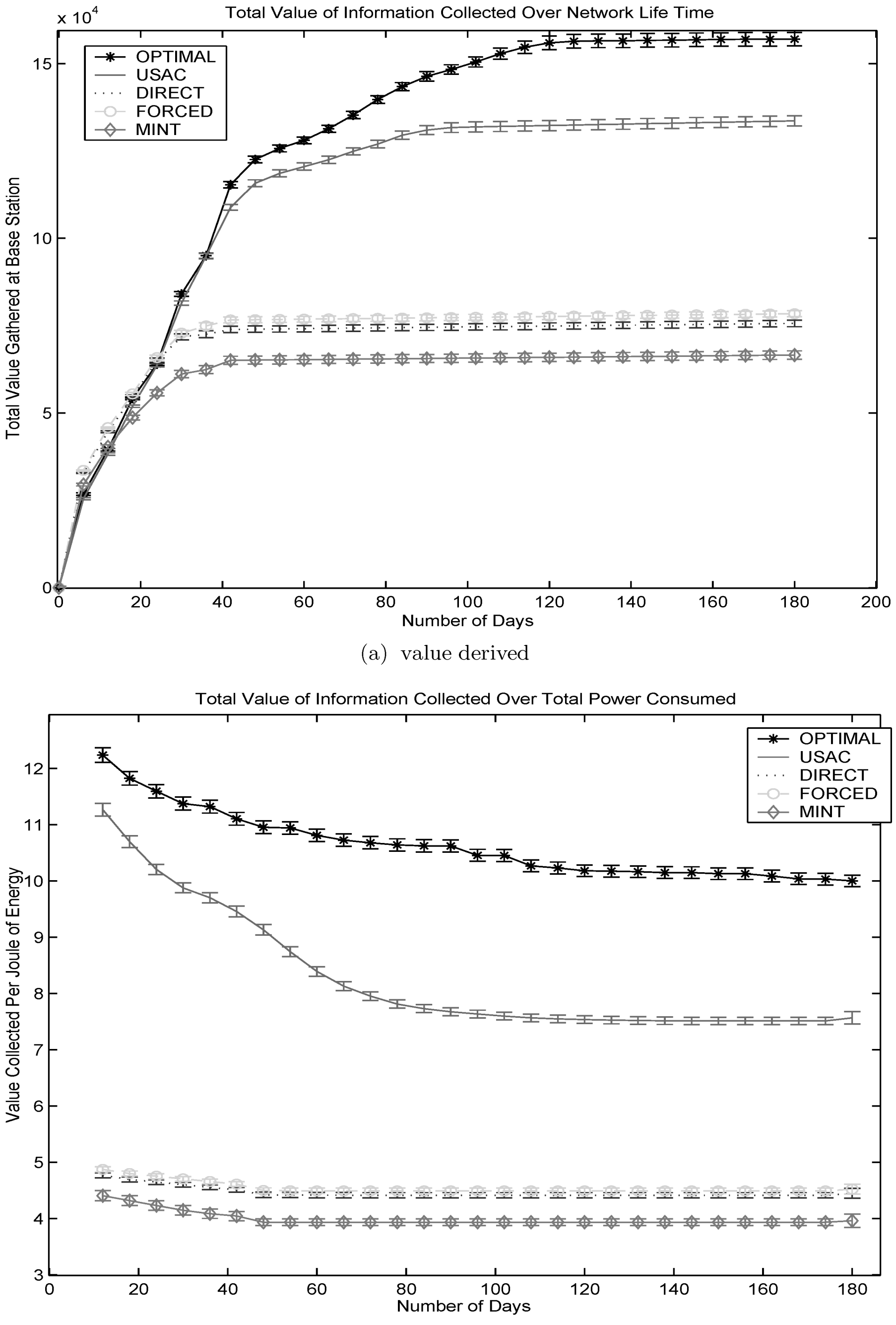
6.3 Dynamic Network Topology (Mobile Nodes)

We extended the previous experiment by introducing node mobility in the network in order to simulate the effect of glacial movement. Based on the glaciologists' advice and our 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 analyze how the protocols fared against constant change in topology during the network lifetime. The results of these simulations are shown in Figure 11. Again, the plots tell 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 Figure 11(b) that in comparison to the OPTIMAL protocol, USAC still performs at 76% of the 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.

6.4 Network Size

In this experiment, we conducted simulations by varying the number of agents in the network from 5 to 40. The aim here is to explore how well USAC adapts to network scalability. To this end, Figure 12 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, we concern ourselves with the average of the final total of these attributes at the end of each simulation to ensure an effective comparison. In more detail, Figure 12(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 actually 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 a greater value of information can be gathered at the base station.

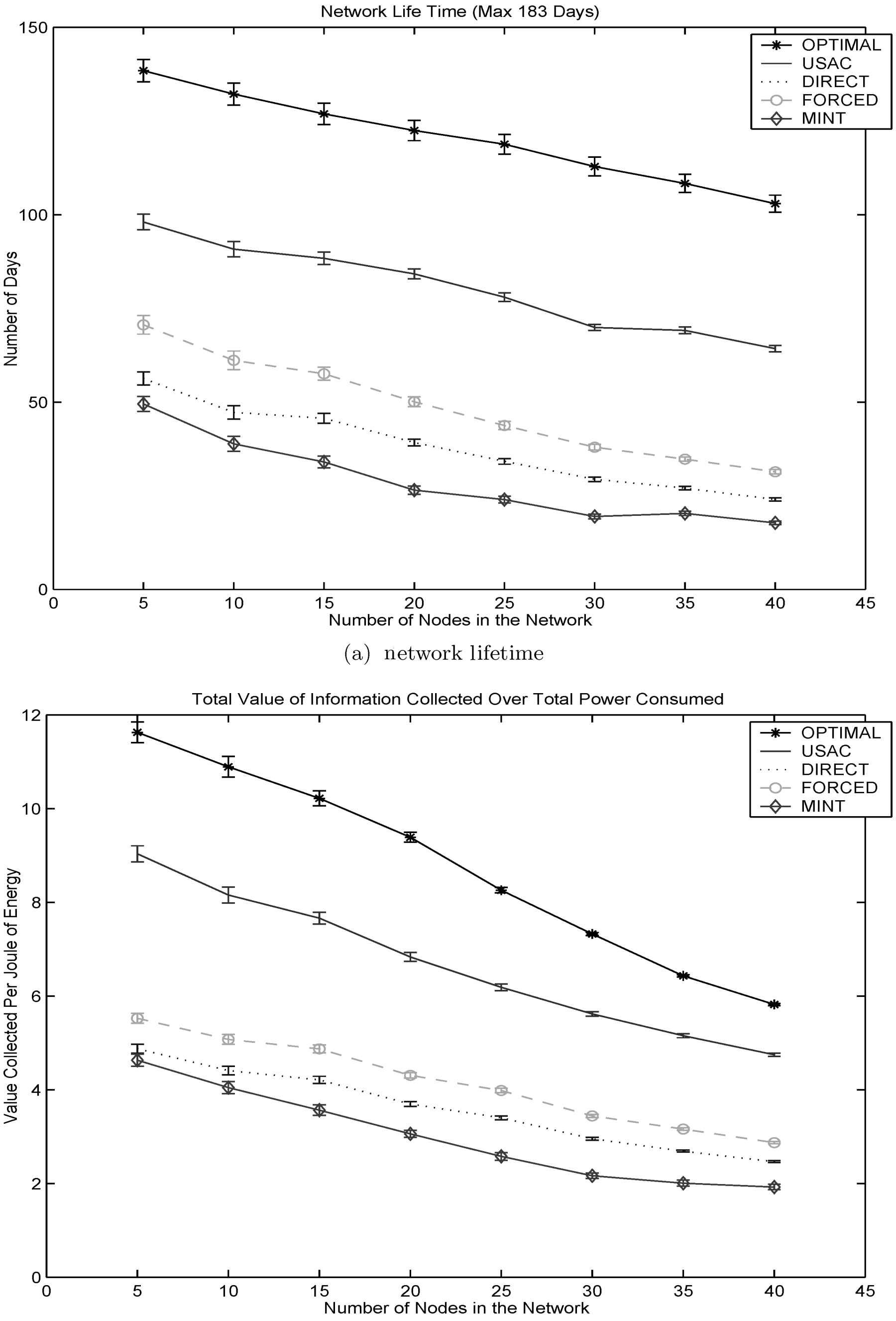
In this case, however, a much better comparison metric is the total value of information gathered at the base station for every joule of energy consumed in the network. Thus, Figure 12(b) illustrates very nicely how this metric is affected by the size of the network for each protocol. Specifically, the initial



(a) value derived

(b) efficiency

Fig. 11. The total value of data gathered and total value of data gatherd per joule over a 6-month period plotted against time (dynamic topology).



(a) network lifetime

(b) efficiency

Fig. 12. Network lifetime and efficiency (at end of network lifetime) plotted against number of agents in the network.

inexpensive 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 least expensive cost path to communicate their data. However, because they send all their sensed data (not all of which provides significant new information) its value is not significant for the amount of energy expended in transmitting it.

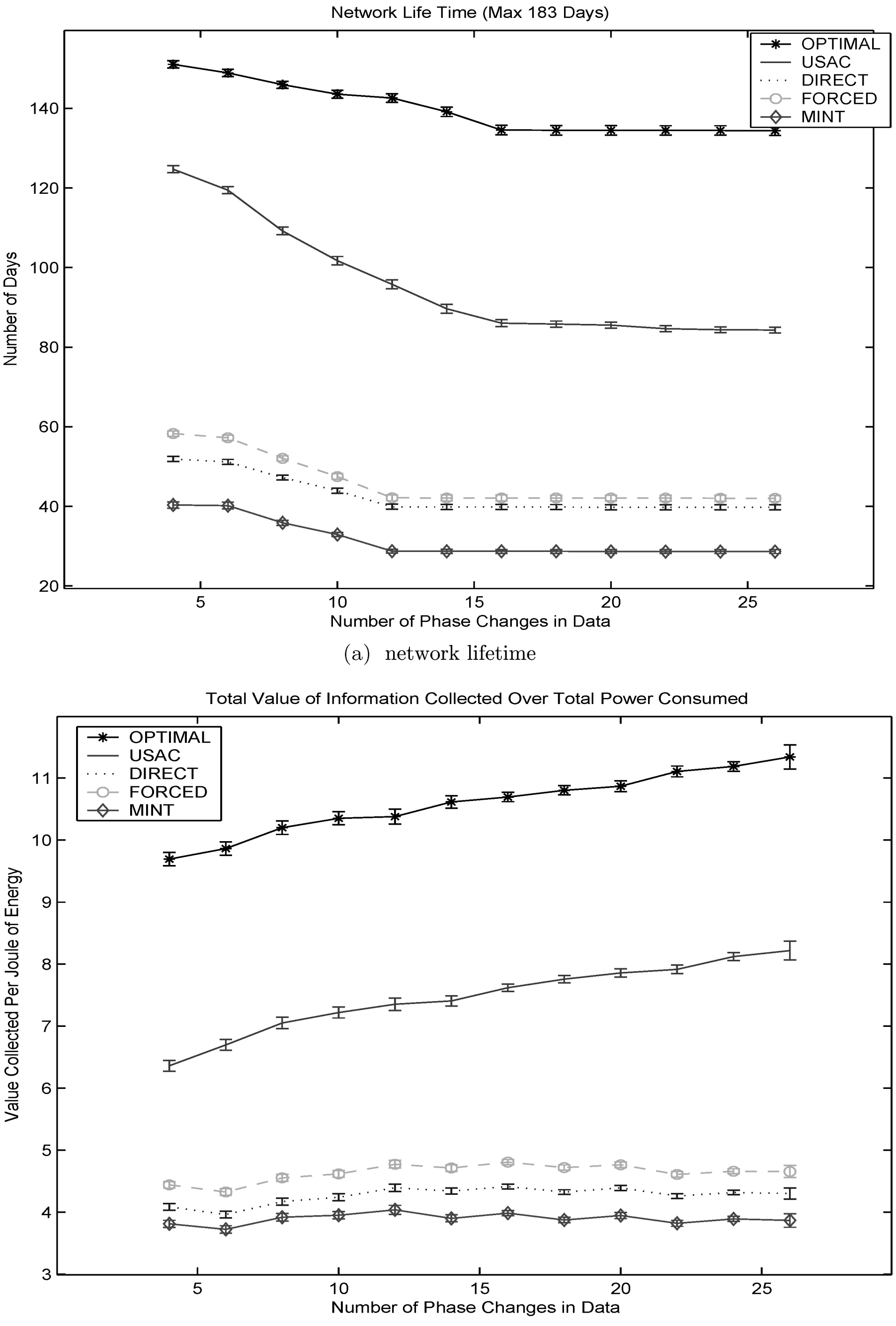
In contrast, USAC agents manage to extract more worthwhile data over the network's lifetime for the same amount of energy used by the other protocols. Their performance in obtaining a total value of information per Joule of energy consumption is rated at 70% of that of the OPTIMAL.

6.5 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 we define the degree of dynamism in the data model 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. This experiment was based on real traces of data collected from the nodes in the field.

In more detail, Figure 13(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 less expensive communication costs and indulge in more frequent transmission activity, which consequently leads to a quicker depletion of energy reserves. The 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 the DIRECT, FORCED, and MINT strategies, respectively. The minimum lifetime of a USAC network is 63% of the minimum lifetime of the OPTIMUM network.

Figure 13(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



(a) network lifetime

(b) value per joule

Fig. 13. Network lifetime and efficiency plotted against a measure of dynamism of the environment.

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 realizing 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.

7. CONCLUSIONS

The protocol we have described in this article allows agents to act in a decentralized manner, based on the nature of their local environment, while self-organizing to form a network whose performance is high in terms of minimizing energy consumption and maximizing the value of data gained. It makes use of the localization ability of individual agents to determine the least expensive cost path to the sink and incorporates the value of the observed data to calculate the most inexpensive path. We have shown that our protocol is far superior to the one currently deployed in GlacsWeb and two other benchmark protocols, even when the size of the network in terms of the number agents, their distribution around the sink, and the nature of observed environment is varied. We have also shown that our protocol is robust in the face of node failure and that its performance is much closer to the optimal protocol.

Whilst we have specifically considered evaluating the effectiveness of our protocol 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, we are currently exploring the possibility of using it in the FloodNet system (a sensor network for monitoring river levels in which the sensors are solar powered).[9] Furthermore, we propose a Bayesian linear regression approach (due to the piecewise 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 nonlinear and there is uncertainty regarding its true functional form without affecting the overall architecture of our sensing and communication protocol.

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

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

Table III. Data for Multiple Linear Regression

Observation i	Response x	t_1	t_2	$\dots$	t_M
1	x_1	t_{11}	t_{12}	$\dots$	t_{1M}
2	x_2	t_{21}	t_{22}	$\dots$	t_{2M}
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
n	x_n	t_{n1}	t_{n2}	$\dots$	t_{nM}

sensed by each agent are independent (or have a very low dependence between them). As part of our future work, however, we 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, we would need to extend USAC to a multiagent sensing approach whereby the agents coordinate their sensing actions to maximize the information they extract from the environment. Furthermore, we would also like to evaluate how our adaptive sampling mechanism compares individually with other state-of-the-art sensing protocols within literature such as KF estimation error and backcasting. Such a comparison would be useful in assessing if there is potential for further energy savings through the task of sensing.

Finally, although USAC performs better than the other protocols compared in this article, it is interesting to see that the efficiency of the scheme drops as network size increases. This might suggest that there exists an optimal network size given a fixed coverage area and it would be useful to evaluate the impact of network density in the future. Furthermore, we are preparing to deploy USAC at a fresh glacier site in Iceland which would help in accurately reflecting the real-life effects of radio connectivity amongst nodes. This would help obtain additional metrics from the network and further refine the protocol.

APPENDIX

A. BEST LINEAR UNBIASED ESTIMATE: PROOF

Linear regression attempts to find a homogeneous real-valued function, $g(\boldsymbol{t}) = \boldsymbol{t}^{\mathrm{T}}\boldsymbol{w}$, that best interpolates the training set $S = \{(\boldsymbol{t}_1, x_1), (\boldsymbol{t}_2, x_2), \dots, (\boldsymbol{t}_N, x_N)\}$. Now, there are a number of error functions that could be used in order to characterize 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 minimize the Euclidean distance between the projected data and the measured data. That is, choose $\boldsymbol{w}^*$ such that $\boldsymbol{w^*} = \arg\min_{\boldsymbol{w}} \frac{1}{n}\sum_{i=1}^{n}(x_i - \boldsymbol{t}_i^{\mathrm{T}}\boldsymbol{w})^2$. By applying the first-order condition on this equation, it can be calculated that $\boldsymbol{w^*} = (T^{\mathrm{T}}T)^{-1}T^{\mathrm{T}}\boldsymbol{x}$. This section shows how this best linear unbiased estimate is derived.

$$x_i = w_0 + \sum_{j=1}^{M} w_j t_{ij} + \epsilon_i, \;\; i = 1, 2, \dots, n \tag{21}$$

In more detail, the least square function is

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

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

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

and

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

Simplifying the preceding 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 Eq. (21) 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.

Here, the vector of least square estimators, $\widehat{\boldsymbol{w}}$ that minimizes S is given as

$$\begin{aligned} S(\boldsymbol{w}) &= \sum_{i=1}^{n} \epsilon_i^2 = \boldsymbol{\epsilon}^{\mathrm{T}}\boldsymbol{\epsilon} = (\boldsymbol{x} - \boldsymbol{T}\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{T}\boldsymbol{w}) \\ &= \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{T}^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{w} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{T}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{w}. \\ &= \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} - 2\boldsymbol{w}^{\mathrm{T}}\boldsymbol{T}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{T}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{w}. \end{aligned}$$

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 must 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

$$\begin{aligned} \boldsymbol{T}^{\mathrm{T}}\boldsymbol{T}\widehat{\boldsymbol{w}} &= \boldsymbol{T}^{\mathrm{T}}\boldsymbol{x} \\ \Rightarrow \widehat{\boldsymbol{w}} &= (\boldsymbol{T}^{\mathrm{T}}\boldsymbol{T})^{-1}\boldsymbol{T}^{\mathrm{T}}\boldsymbol{x}. \end{aligned}$$

B. POSTERIOR DISTRIBUTION AS GAUSSIAN WITH MEAN AND COVARIANCE MATRIX: PROOF

In our case, we want to predict the sensor's next measurement based on all our observations so far. 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 which is given as $p(x_{n+1} \mid \boldsymbol{t}_{n+1}, T, \boldsymbol{x}) = \int p(x_{n+1} \mid \boldsymbol{t}_{n+1}, \boldsymbol{w}) p(\boldsymbol{w} \mid T, \boldsymbol{x})\, d\boldsymbol{w}$. 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 apply Bayes' rule. Assuming the prior distribution $p(\boldsymbol{w}) \sim \mathcal{N}(0, \Sigma_p)$ and marginalizing out $p(\boldsymbol{x} \mid T)$, it can be shown that $p(\boldsymbol{w} \mid T, \boldsymbol{x}) \sim \mathcal{N}(\overline{\boldsymbol{w}}, A^{-1})$ where $A = \sigma^{-2}TT^{\mathrm{T}} + \Sigma_p^{-1}$ and $\overline{\boldsymbol{w}} = \sigma^{-2}A^{-1}T\boldsymbol{x}$.

This appendix provides the derivation of this result. We have

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

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

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

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

where $\overline{\boldsymbol{w}} = \sigma_n^{-2}(\sigma_n^{-2}\boldsymbol{T}\boldsymbol{T}^T + \Sigma_p^{-1})^{-1}\boldsymbol{T}\boldsymbol{x}$. The form of the posterior distribution can then be recognized 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 $A = \sigma^{-2}\boldsymbol{T}\boldsymbol{T}^{\mathrm{T}} + \Sigma_p^{-1}$ and $\overline{\boldsymbol{w}} = \sigma^{-2}A^{-1}\boldsymbol{T}\boldsymbol{x}$.

ACKNOWLEDGMENT

We would like to express our thanks to the anonymous reviewers for their insightful comments in making the article better.

REFERENCES

BERTONI, H. L. 1999. *Radio Propagation for Modern Wireless Systems*. Prentice Hall.

BOX, G. E. P. AND TIAO, G. C. 1992. *Bayesian Inference in Statistical Analysis*. John Wiley & Sons.

BYERS, J. AND NASSER, G. 2000. Utility-Based decision-making in wireless sensor networks. In *Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking & Computing (Mobihoc'00)*. IEEE Press, 143–144.

CERPA, A., ELSON, J., ESTRIN, D., GIROD, L., HAMILTON, M., AND ZHAO, J. 2001. Habitat monitoring: Application driver for wireless communications technology. *SIGCOMM Comput. Comm. Rev. 31*, 2, 20–41.

DASH, R. K., ROGERS, A., REECE, S., ROBERTS, S., AND JENNINGS, N. R. 2005. Constrained bandwidth allocation in multi-sensor information fusion: A mechanism design approach. In *Proceedings of the 8th International Conference on Information Fusion*.

DELIN, K., HARVEY, R., CHABOT, N., JACKSON, S., ADAMS, M., JOHNSON, D., AND BRITTON, J. 2003. Sensor web in antarctica: Developing an intelligent, autonomous platform for locating biological flourishes in cryogenic environments. In *Proceedings of Lunar and Planetary Institute Conference*.

DESHPANDE, A., GUESTRIN, C., MADDEN, S., HELLERSTEIN, J., AND HONG, W. 2005. Model-based approximate querying in sensor networks. *Int. J. VLDB 14*, 4.

DOWNEY, P. AND CARDELL-OLIVER, R. 2004. Evaluating the impact of limited resource on the performance of flooding in wireless sensor networks. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN'04)*. 785–794.

ELSAIFY, A., PADHY, P., MARTINEZ, K., AND ZOU, G. 2007. Gwmac- 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*. 54–61.

HEINZELMAN, W. R., CHANDRAKASAN, A., AND BALAKRISHNAN, H. 2000. Energy-Efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences*.

HEINZELMAN, W. R., KULIK, J., AND BALAKRISHNAN, H. 1999. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)*. 174–185.

INTANAGONWIWAT, C., GOVINDAN, R., AND ESTRIN, D. 2000. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th Annual International Conference on Mobile*. ACM Press, New York, 56–67.

JAIN, A. AND CHANG, E. Y. 2004. Adaptive sampling for sensor networks. In *Proceeedings of the 1st International Workshop on Data Management for Sensor Networks (DMSN'04)*. ACM Press, New York, 10–16.

KULLBACK, S. 2001. *Information Theory and Statistics*. John Wiley and Sons, New York.

KULLBACK, S. AND LEIBLER, R. A. 1951. A probabilistic approach to concurrent mapping and localization for mobile robots. *Ann. Math. Statist. 22*, 1, 79–86.

LESSER, V., ORTIZ, C., AND TAMBE, M. 2003. *Distributed Sensor Networks: A Multiagent Perspective*. Kluwer Publishing.

LIN, S., ZHANG, J., ZHOU, G., GU, L., HE, T., AND STANKOVIC, J. A. 2006. Atpc: Adaptive transmission power control for wireless sensor networks. In *Proceedings of ACM SenSys*.

LINDSEY, S. AND RAGHAVENDRA, C. S. 2002. Pegasis: Power efficient gathering in sensor information systems. In *Proceedings of the IEEE Aerospace Conference*.

LORINCZ, K., MALAN, D., FULFORD-JONES, T. R. F., NAWOJ, A., CLAVEL, A., SHNAYDER, V., MAINLAND, G., MOULTON, S., AND WELSH, M. 2004. Sensor networks for emergency response: Challenges and opportunities. *IEEE Pervasive Comput. 3*, 4, 16–23.

MARTINEZ, K., HART, J. K., AND ONG, R. 2004. Environmental sensor networks. *Comput. 37*, 8, 50–56.

MERRETT, G., HARRIS, N., AL-HASHIMI, B., AND WHITE, N. 2007. Energy managed reporting for wireless sensor networks. *Sensors Actuators A: Phys. 142,* 1.

PADHY, P., MARTINEZ, K., RIDDOCH, A., HART, J., AND ONG, H. 2005. Glacial environment monitoring using sensor networks. In *Proceedings of the 1st Real-World Wireless Sensor Networks Workshop (REALWSN'05)*. 10–14.

PAN, J., HOU, Y. T., CAI, L., SHI, Y., AND SHEN, S. X. 2003. Topology control for wireless sensor networks. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*. ACM, 286–299.

PERKINS, C. E. AND BHAGWAT, P. 1994. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *ACM SIGCOMM Comput. Comm. Rev. 24*, 4, 234–244.

POPA, D., SANDERSON, A., VADIRAJ, H., KOMERSKA, R., MUPPARAPU, S., BLIDBERG, R., AND CHAPPEL, S. 2005. Optimal sampling using singular value decomposition of the parameter variance space. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS'05)* IEEE/RSJ, 3131–3136.

RAMANATHAN, R. AND ROSALES-HAIN, R. 2000. Topology control of multihop wireless networks using transmit power adjustment. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies*. IEEE Press, 404–413.

RASMUSSEN, C. E. AND WILLIAMS, C. K. I. 2005. *Gaussian Processes for Machine Learning*. The MIT Press.

ROGERS, A., CORKILL, D. D., AND JENNINGS, N. R. 2009. Agent technologies for sensor networks. *IEEE Intell. Syst. 24*, 2, 13–17.

ROGERS, A., DAVID, E., AND JENNINGS, N. R. 2005. Self organised routing for wireless micro-sensor networks. *IEEE Trans. Syst. Man Cybernet. (Part A) 35*, 3, 349–359.

ROURE, D. D. 2005. Floodnet: A new flood warning system. *Roy. Acad. Engin. Quart. 23*, 48–51.

SON, D., KRISHNAMACHARI, B., AND HEIDEMANN, J. 2004. Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks. In *Proceedings of IEEE SECON*. 289–298.

STEERE, D., BAPTISTA, A., MCNAMEE, D., PU, C., AND WALPOLE, J. 2000. Research challenges in environmental observation and forecasting systems. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom'00)*. 292–299.

WATTENHOFER, R., LI, L., BAHL, P., AND WANG, Y. M. 2001. Distributed topology control for power efficient operation inmultihop wireless ad hoc networks. In *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*. IEEE Press, 1388–1397.

WILLETT, R., MARTIN, A., AND NOWAK, R. 2004. Backcasting: Adaptive sampling for sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*. ACM Press, New York, 124–133.

WILLIAMS, C. 1998. Prediction with gaussian processes: From linear regression to linear prediction and beyond. Tech. rep. NCRG/97/012, Neural Computing Research Group, Department of Computer Science and Applied Mathematics, Aston University.

WOO, A., TONG, T., AND D.CULLER. 2003. Taming the underlying challenges or reliable multihop routing in sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys'03)*. 14–27.

ZAMALLOA, M. Z. AND KRISHNAMACHARI, B. 2007. An analysis of unreliability and asymmetry in low-power wireless links. *ACM Trans. Sensor Netw. 3*, 2.

ZHOU, G., HE, T., KRISHNAMURTHY, S., AND STANKOVIC, J. A. 2006. Models and solutions for radio irregularity in wireless sensor networks. *ACM Trans. Sensor Netw. 2*, 2, 221–262.

Received November 2007; revised June 2009; accepted August 2009