Tuning Complex Sensornet Systems Using Principled Engineering Methods

Jonathan Tate and Iain Bate
Department of Computer Science
University of York
York, United Kingdom
{*jt* | *iain.bate*}*@cs.york.ac.uk*

Abstract

Sensornet lifespan and utility is limited by the energy resources of individual motes. Network designers seek to maximise energy efficiency while maintaining acceptable Quality of Service. However, the interactions between multiple tunable protocol parameters and multiple performance metrics are generally complex and unknown, and combinatorial explosion renders impossible any exhaustive search approach. In this paper we describe an engineering method to address this multi-dimensional optimisation problem. We apply a Design Of Experiments approach to sample the entire search space. Statistical models are fitted to experimental results to define relationships between inputs and outputs, and to obtain near-optimal solutions.

1. Introduction

Wireless sensor networks, or sensornets, compose many autonomous *motes* into ad-hoc networks for distributed sensing and processing applications. Motes are small, cheap computers equipped with independent power supplies, wireless communication capability, and sensors with which to passively monitor with their environment. Typical applications include environmental monitoring or surveillance. Interaction with the physical environment implies that sensornets have real-time requirements.

Distributed applications generally require the network to deliver a minimal specified Quality of Service (QoS). The greater the margin by which the achievable QoS exceeds the minimal required QoS, the greater the tolerance of the deployed network to unexpected events and conditions. Most non-trivial protocols have parameters which can be fine-tuned to influence network performance [1]. However, it is generally not trivial to identify the subset of candidate tunings which achieve the required QoS, nor to identify which specific tuning offers the optimal QoS.

Sensornet designers must identify the most significant factors to avoid being swamped by unnecessary detail. Unfortunately, even identifying the relative importance of factors and their interactions is rarely trivial. Discovering the best values to assign to these factors and understanding their impact on network behaviour tradeoffs is harder still. Tunable parameters are often defined without clear default values and may be defined over an infinite range.

Where multiple controllable factors exist, each of which can take many values, combinatorial explosion renders exhaustive exploration impossible. Designers may resort to inefficient trial-and-improvement techniques or accept sub-optimal tunings. However, pragmatic approaches can approximate exhaustive exploration in bounded time. In this paper we propose a principled search method based on full factorial design experiments [2].

A common design goal is to maximise sensornet lifetime. This is usually achieved by maximising the efficiency of radio communications, as wireless communication components are generally the most energy-hungry subsystem of sensornet motes [3]. In this paper we explore how existing well understood energy-ignorant protocols can be optimised. The tuning method we describe is equally applicable to both energy-ignorant and energy-aware protocols. We demonstrate its efficacy and versatility by applying it to two fundamentally different protocols: *TTL-Bounded Gossip* (TBG) [4] and *Implicit Geographic Forwarding* (IGF) [5].

Undertaking this work presented interesting challenges. Sensornet protocol tuning is not a simple, idealised problem. It is a complex real-world problem with multiple inputs, multiple outputs, and multiple objectives. The non-trivial interrelationships between these factors were not known at the outset, so could not be targeted specifically during experiment design. Two important engineering challenges addressed by our method are the production of solutions exhibiting robustness to deployment context, and the uncertainty and noise inherent in any experimental data.

The remainder of the paper is structured as follows. Section 2 places this paper in the context of related work. Section 3 outlines the research objectives of this paper. Section 4 describes the experimental method in general terms. Section 5 describes the specific problems to which this method is applied. Section 6 describes the implementation and results of a three-phase experimental approach. Finally, section 7 draws conclusions against our objectives.

2. Related work

Energy awareness and *energy management* are themes running throughout most aspects of sensornet design and operation. The energy resources of nodes are typically small and non-renewable. Energy consumption is *the most important factor that determines sensor node lifetime* [6].

Optimising sensornets for energy efficiency is complex. Raghunathan [6] observes that it *involves not only reducing the energy consumption of a single sensor node but also maximising the lifetime of an entire network*, requiring dynamic trade-offs between *energy consumption*, *system performance*, and *operational fidelity*, yielding *up to a few orders of magnitude* of improved lifetime.

With many controlled factors and measured responses it is generally difficult to understand the resulting complex interrelationships. Totaro and Perkins [7] apply a *systematic statistical Design Of Experiments* approach to evaluate and model the complex tradeoffs in designing Mobile Ad-Hoc Networks (MANETs). This work considers the impact of varying network design with a fixed network application. In contrast, we consider the impact of varying network protocol and application behaviour for a fixed network design, finding near-optimal solutions that demonstrate *robustness* to network design factors.

In classic flooding a node broadcasts a packet to each of its neighbours which in turn rebroadcast the packet and so on. Flooding is utilised by most non-geographical routing protocols [8] and often succeeds where more sophisticated protocols cannot react quickly to rapidly changing networks [9]. Counterintuitively complex behaviour is observed [10] despite the protocol's simplicity. *Broadcast storms* [4] are particularly problematic with significant redundant broadcasts, contention, collisions, and high energy consumption.

Gossiping extends flooding by implementing probabilistic rebroadcast but can provide only probabilistic guarantees of delivery, displaying bimodal behaviour where either hardly any nodes receive the packet, or almost all do [11]. Gossip probabilities in the range [0.6, 0.8] often, but not always, ensure that most nodes receive most packets. Appropriate gossip probability selection is generally difficult, and may need to vary across nodes and time [12].

More sophisticated energy-efficiency protocols can employ a variety of underlying techniques, in which nodes maintain online models of network activity and structures. Other protocols such as Implicit Geographic Forwarding [5] exploit geographic context in routing decisions. A comprehensive survey of sensornet protocols can be found in [1].

3. Research objectives

We define the following objectives that form the principal contributions of this paper:

Objective 1: *Define a reusable methodology for addressing the protocol tuning problem*

Objective 2: *Identify the significant factor-response relationships for the protocol tuning problem*

Objective 3: *Obtain near-optimal compromise solutions to the protocol tuning problem*

4. Experimental method

In this section we define the experimental method through which the parameter landscape is explored. We also consider the experimental cost and analysis of results.

4.1. Three-phase experiment design

Full factorial design [2], [13] is used to systematically explore the entire parameter landscape. This approach gives broad but shallow coverage of all possible combinations of all acceptable ranges of controlled factors. We address the combinatorial explosion explored in section 4.3 by applying a three-phase method designed to avoid wasting resources and analytical effort on matters which will not significantly influence the outcome, allowing more detailed statistical models to be derived for a given cost.

Phase 1 determines the point at which simulations are sufficiently stable to be sampled as representative of long-term stable behaviour. Phase 2 samples the problem space at low resolution, identifying insignificant factors which can be dropped to make high resolution modelling tractable. Phase 3 samples the problem space at high resolution for significant factors only, yielding a partial mapping from possible protocol tunings to consequent network behaviour.

Statistical models are fitted to experimental results obtained in Phase 3 to summarise the complex interrelationships between controlled factors and each measured response. This derived model is useful for predicting likely network performance for any arbitrary set of input values. The derived model can also be applied in the opposite direction by defining sections of the multi-response hypersurface corresponding to the desired network performance, and working backwards to the input values. The simultaneous equations of the fitted model are solved to yield a set of inequalities defining usable ranges of input controlled factors. We implement both usage patterns in section 6.

4.2. Fitting statistical models to data

Assume we define p controlled factors, and sample each at q evenly-spaced points. This sampling defines q^p design points, distributed evenly throughout the protocol configuration space. The factorial design experiments implemented in section 6 map each sampling point to a set of metrics. These pairs of sample points and simulation-derived metrics represent exact solutions to specific known points in the generalised model of the relationship between controlled factors and output metrics. However, these are not directly usable if we wish to know the relationship between controlled factors and measured responses, or vice-versa, for other points in the factor-response phase space.

To consider points in the parameter space other than those measured directly we use interpolation techniques. A statistical model is fitted to sampled points and corresponding measurements to derive a set of equations describing a hypersurface in the phase space [14]. An appropriate statistical model must be selected which approximates the surface shape which would be observed under an infinite number of sample points. We then work with the fitted surface rather than specific individual experimental results.

Sampling the parameter space at more points yields a fitted model which is a better approximation of the real relationship by providing more data for the model fitting algorithm. For a finite set of sample points there exists the risk that an interesting feature of the solution landscape falls between sample points, and hence is not present in the fitted model. Interpolation allows every candidate parameter set to be considered simultaneously, including those not measured directly, but there is a risk that the optimal solution lies between directly measured points and is not revealed in the fitted model.

For each output metric under consideration, a separate statistical model of the form given in Equation 1 can be fitted to the result set in MATLAB. β_0 is a constant, X_i is the ith controlled factor value, β_i is the coefficient for controlled factor X_i, β_{ij} is the coefficient for the interaction between controlled factors X_i and X_j, and ε is the normally-distributed noise term. The response M_i is influenced linearly by each factor and each pairing of potentially interacting factors. Our analysis shows this model to be a good fit for the experimental results considered in this paper.

$$M_\alpha = \beta_0 + \sum_{i=1}^{n} \beta_i X_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \beta_{ij} X_i X_j + \varepsilon \quad (1)$$

For each output metric M_1-M_m a separate linear interaction model is produced by Equation 1 in which a set of n axes represents controlled factors X_1-X_n and a further axis in which the height of the hypersurface varies with the values of the output metric M_α. As the axes corresponding to controlled factors X_1-X_n are common to all metrics M_1-M_m it is possible to combine them all to yield a more complex surface representing the interrelationships between all controlled factors and all metrics.

Finding sets of values for controlled factors corresponding to solutions with appropriate characteristics is equivalent to identifying regions of the axes representing controlled factors X_1-X_n with appropriate fitted surface height in the axes corresponding to output metrics M_1-M_m. Similarly, finding optimal or worst-case sets of controlled factors is equivalent to finding minima and maxima in the fitted surface. This is implemented by solving sets of simultaneous inequalities when identifying regions with suitable characteristics, or by solving sets of simultaneous equations when addressing optimal or worst-case characteristics.

Experiment designers can also consider other models, such as higher order linear models, selecting that which offers the best fit to the dataset. For example, the quadratic model shown in Equation 2 includes all terms of Equation 1 with additional terms for squares of controlled factors. Additional terms can be added to consider ever higher degrees of controlled factors and their interactions. Experiment designers might also consider generalised linear models in which the M_α term of Equations 1 and 2 are replaced by $f(M_\alpha)$, where a better fit might be achieved by applying a transformation to the measured response. For example, we could consider taking the natural logarithm of the response by defining $f(M_\alpha) = \ln M_\alpha$.

$$M_\alpha = \beta_0 + \sum_{i=1}^{n} \beta_i X_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \beta_{ij} X_i X_j + \sum_{k=1}^{n} \beta_k X_k^2 + \varepsilon \quad (2)$$

4.3. Cost analysis

Exhaustive exploration of the protocol configuration space is typically infeasible, and often impossible, due to combinatorial explosion. This is a consequence of both the number of controlled factors and the number of values which each factor can take, the latter being infinite for continuously variable factors. Our method, based on full factorial design [2], samples the protocol configuration space at a finite set of points to render tractable the evaluation effort. Increasing the number of experimental configurations increases the quality of fitted statistical models, and hence solution quality, but increases experiment cost. A balance must be found which obtains solutions of acceptable quality within acceptable wall time and experimental overhead.

Consider the algorithmic complexity of this approach. Assume we define p controlled factors and sample each at q evenly-spaced points, giving q^p design points distributed evenly throughout the parameter space. If we evaluate each design point for each of r networks then we define rq^p test configurations. We simulate each test configuration s times to prevent any single simulation instance exerting undue influence, requiring rsq^p test cases in total.

The test suite size grows in p, q, r and s, but in a qualitatively different manner. Linear growth in r and s is observed as the set of design points is repeated for each of r networks, and the set of test configurations is repeated s times without modification. Polynomial and exponential growth in q and p respectively are observed because the design matrix defining the design points set can be represented as unit cells within a hypercube. Increasing q increases the length of the hypercube sides, whereas increasing p increases the dimensionality of the hypercube. Test suite cost grows as $O(n)$ in r and s, $O(n^c)$ in q, and $O(c^n)$ in p.

Although the cost is NP-hard in p, our experimental method addresses this potential problem. Firstly, for a given network protocol there are a finite number of controllable factors, only a subset of which are likely to be of interest or permit alteration by the network designer. This places a small, finite upper bound on p for a given protocol. Secondly, Phase 2 of our experiments implements a *screening* approach which further reduces p by identifying insignificant factors which can safely be disregarded. It is therefore possible in Phase 3 to increase q after reducing p and still have all experiments complete in acceptable wall time.

The polynomial growth in g is also managed in the experiment design. Recall from section 4.2 that we fit linear interaction models to measured values. A linear relationship in one factor can be uniquely defined by just two factor-response pairs. Extending this to a linear relationship in p factors requires two values of each controlled factor to be represented in the set of design points [2]. We therefore require only that $q \geq 2$, with low and high values of each factor representing the range for which the model is required to predict metric values. Higher values of q obtain better fitted models but with decreasing gains for each additional sampling point, so small values of q work well [2] and minimise simulation cost. Higher-order linear models of order d would require $q \geq d$.

As all simulations imply similar computational overheads we assume each simulation completes in approximately constant wall time, t. All simulations are mutually independent and can therefore be executed in parallel, reducing total runtime to that of a single simulation if sufficient processing hosts are available. Assume a multiprocessing environment in which $x \in \mathbb{N}$ independent simulations can execute in parallel. In factorial design test suites there are no dependencies between simulations so any number can execute in parallel, all at cost t. The total wall time cost is $C = \frac{rsq^p}{x}t$. Note that $C \propto \frac{1}{x}$, reaching a minimum of $C = t$ where $x = rsq^p$.

4.4. Simulation environment

It is impractical to perform the experiments described in this paper using real networks due to high overheads of logistics, cost and time. Additionally, it is generally impossible to guarantee a consistent and unchanging environment for the total runtime of the tens of thousands of experiments. This would severely undermine the validity of comparison between results obtained from multiple experiments, which is critical to the analytical methods we propose.

To address these concerns all experiments were conducted by simulation. An appropriate simulation tool must be selected because solution quality is dependent on simulation accuracy. The design and validation of *YASS*, the simulator used in the experiments described in this paper, is considered in [15]. *YASS* is a multithreaded sensornet optimised for efficiency and for this duty pattern. Multiple independent simulation instances can be executed in parallel to take advantage of low-cost commodity hardware.

The resulting efficiency was such that simulated time passed faster than wall time; results were obtained more quickly than real-world experiments could provide results, even if unlimited resources were available. However, the approach presented here could be implemented with equivalent results in any sensornet simulator in which protocol factors can be controlled and solution quality metrics measured, or even in real networks if suitable testbeds could be constructed.

5. Experimental configuration

In this section we define an experimental sensornet configuration for which we will address the protocol tuning problem in section 6.

5.1. Network design

The techniques outlined in this paper are independent of the specific protocols and network designs explored in the following experiments. However, these experiments explore only a finite portion of the unbounded design space of all networks and all protocols. It is likely that the trends we identify in network performance responses as a function of protocol tuning parameters will remain applicable in other similar networking contexts. Nevertheless, we limit the scope of our claims to the portion of design space defined in this section, within which we have confidence in our findings as they are demonstrated to have statistical significance.

A set of three typical sensornets, $\Xi = \{\xi_1, \xi_2, \xi_3\}$, was defined and reused for all experiments. Each sensornet consisted of 500 static motes of identical capability modelled on the Crossbow MICA2 mote. Motes were distributed randomly within a square of side length 21Km yielding a geographic distribution of uniform planar density. This size was selected so that average degree of connectivity was approximately 40, which is typical of sensor networks [5].

All internodal communication was defined to occur through anisotropic radio broadcast in an obstacle-free vacuum. Signal propagation and attenuation was modelled using the Friis free space model with exponent of 2.0. The simulated motes ran a simulated distributed sensing application in which every node periodically produces a small data packet. The destination of each packet is randomly selected from all motes in the network to prevent bias from implicit structure.

Although the protocols considered in this paper can achieve nearly 100% packet delivery under ideal conditions, we tuned the network load sufficiently that any protocol would be unlikely to achieve 100% delivery due to contention, but not so heavily as to load the network substantially beyond its capacity. Little insight is to be gained by experiments addressing unrealistically favourable or disfavourable workloads.

Each node in the network can act as a packet source, a packet destination, or a packet relay. When a source node creates a packet it is queued for broadcast to the wireless medium. If the packet is eventually broadcast it may be received by one or more other nodes within communication range able to successfully extract the packet data from background noise. Packet headers specify one or more destinations, defining the only nodes at which a given packet can be consumed. In our experiments we specify exactly one destination per packet. Packet headers also specify *Time To Live* (TTL) in terms of node-to-node hops and lifespan to prevent stale packets circulating indefinitely.

Each packet recipient node independently determines how to handle the incoming packet. Three main classes of action are possible; the packet may be consumed, queued for rebroadcast, or dropped. The details of the criteria upon which the node makes this decision, and the state information upon which this decision is based, is dependent on the traffic distribution protocol selected by the sensornet designer. Nevertheless, for all protocols the range of available actions is generally limited to these three possibilities.

5.2. Protocol selection

A rich and diverse set of routing protocols have been proposed in the literature and implemented in industry. It is impractical to assess each extant protocol as there are too many. Instead, this paper elects to consider two representative protocols. Lightweight protocols remain relevant to the extreme resource constraints of small, low-cost motes and have the additional benefit that their complexity will not obfuscate the results of the methods proposed. For similar reasons the protocol chosen should be stateless, making no assumptions about the nature of the application, to avoid any form of bias in the findings.

In this paper we consider two protocols designed for MANETs, both of which implement a low-state lazy binding approach. The *TTL-Bounded Gossip* protocol [4] is described in section 5.4 and the *Implicit Geographic Forwarding* (IGF) protocol [5] is described in section 5.5. These protocols were chosen for their simplicity. More complex protocols often incorporate simple protocols during early discovery phases or to maintain information. If implemented carelessly these simple protocols can be highly wasteful, and hence offer an excellent opportunity for saving energy. For example, unbounded flooded messages can easily cover the entire network [8] which is wasteful if the source and destination are physically close.

Note that in selecting these protocols we make no claims as to their merit for any given sensornet application. More specifically, we do not claim that when optimally configured they necessarily offer superior performance to other recent and more complex alternative protocols. However, we see no reason that the methods described in this paper could not be applied to these other protocols.

5.3. Protocol-independent controlled factors

In this paper we consider two traffic distribution protocols, *TTL-Bounded Gossip* and *Implicit Geographic Forwarding*. Each of these protocols has parameters which are defined independently of any given network configuration, but can be tuned by a network designer to achieve a desired behaviour or to implement some resource usage tradeoff. Some tunable parameters are specific to a given protocol, but others are common to several protocols.

In this section we define controlled factors $X_1 - X_5$ which are common to both *TTL-Bounded Gossip* and *Implicit Geographic Forwarding*, and may interact with other shared parameters and protocol-specific parameters. We define our experiments to explore as much of the parameter space as is possible. For each parameter $X_1 - X_5$ we limit our search to a subset of the defined range within which a measurable difference in response is known to exist [16].

- X_1: **Seen-packet buffer size** The number of packets received or transmitted by a node of which knowledge is retained. Nodes do not retransmit a previously-transmitted packet if the latter is held in this cache. New packets displace a randomly-selected cached packet if the buffer is full. Measured in *packets*. Defined in the range $[0, \infty)$ for integral values only. Search range is $[1, 10]$.
- X_2: **Waiting-packet buffer size** The number of packets which can be simultaneously enqueued for transmission or retransmission. Packets are consumed from the queue head and added to the queue tail. If the queue is full when a new packet is added, a randomly-selected enqueued packet is dropped. Measured in *packets*. Defined in the range $[1, \infty)$ for integral values only. Search range is $[1, 10]$.
- X_3: **Initial backoff** Before beginning transmission of a packet the sending node will sense the wireless medium. If the medium is clear transmission begins immediately, otherwise an exponential backoff strategy is applied in which the nth term is the nth power of this base value. Measured in *seconds*. Defined in the range $(0, \infty)$. Search range is $[0.1, 1]$.
- X_4: **Packet lifetime** The maximum permitted time for a packet to remain in transit. If the lifetime is exceeded before reaching the destination, the packet is dropped. Measured in *seconds*. Defined in the range $(0, \infty)$. Search range is $[0.1, 10]$.
- X_5: **TTL** The total number of node-node hops permitted for packets traversing the network. If this TTL is exceeded prior to reaching the destination, the packet is dropped. Measured in *hops*. Defined in the range $[1, \infty)$ for integral values only. Search range is $[1, 10]$.

Other networking protocols may be influenced by a different set of factors, which may or may not intersect the above set. However, any networking protocol for which there exists a set of quantitatively-defined factors can be explored using this process.

5.4. TTL-Bounded Gossip protocol

An adapted form [16] of the *TTL-bounded gossiping* protocol [4] is the first protocol under consideration. This protocol is ignorant of energy, network topology, and the host application, ensuring no bias in the results produced. Flooding and gossiping protocols of this form are commonly used within more complex protocols [4] to establish delivery routes or maintain awareness of network status, widening the scope of our results to all such protocols.

The protocol makes no demands of a node wishing to broadcast a packet, either for packets newly created by the application or when forwarding packets. When a packet is broadcast, each recipient makes an independent probabilistic decision whether to rebroadcast the packet to its neighbours, if it is not to be dropped or consumed. The packet thus radiates outward from the source node, hopefully arriving at least once at each intended destination.

In addition to the protocol-independent controlled factors defined above in section 5.3 an additional controlled factor must be specified.

X_6: **Gossip rebroadcast probability** The probability that upon receiving a packet, which is not to be consumed or dropped at the recipient, a given node will enqueue the packet for later retransmission to its neighbours. Unitless. Defined in the range $[0, 1]$. Search range is $[0, 1]$.

5.5. Implicit Geographic Forwarding protocol

An adapted form of the *Implicit Geographic Forwarding* (IGF) [5], in which the backtracking support is removed for simplicity, is the second protocol under consideration. This protocol is ignorant of energy, network topology, and the host application, ensuring no bias in the results produced. Unlike flooding-derived protocols, IGF implements a three-phase handshaking sequence to moderate data packet broadcast. Consider a packet p with source A and destination D, currently at node S. Node S broadcasts a short *Request-To-Send* (RTS) received by neighbouring nodes $N_i \in N_{neighbours}$. Each RTS recipient N_i considers its geographic position relative to S and D, and if the angle $\angle DSN_i < \theta$ (where θ is a controlled factor X_7) N_i broadcasts a short *Clear-To-Send* (CTS). $\angle DSN_i$ is trivially $0°$ if $N_i = D$. If S receives one or more CTS replies, it selects the node N_i offering the smallest $\angle DSN_i$ and selects this as the next recipient. Packet p is then broadcast with this choice added to its header. All neighbours $N_i \in N_{neighbours}$ except the selected N_i can safely ignore p. When the selected N_i receives p it sends a short Acknowledgement (ACK) to S, completing this stage of the process. The process repeats, with the previous N_i becoming the new S, until the packet arrives at D or a node N_i for which there are no suitable forwarding candidate neighbours.

In addition to the protocol-independent controlled factors defined above in section 5.3 two additional controlled factors must be specified.

X_7: **CTS threshold angle** When node N_i receives a CTS message from S, it will not send an RTS unless $\angle DSN_i < X_7$. This factor is intended to prevent many low-quality or poorly located forwarding candidates sending RTS messages, and prevents packets being forwarded in the opposite direction to the destination if $X_7 < 90$. Measured in *degrees*. Defined in the range $[0, 180]$. Search range is $[5, 85]$.

X_8: **State timeout base** Complete IGF cycles imply several wait/timeout periods. To minimise the search space we define all as multiples of a single parameter X_8, such that $CTS\_WAIT = X_8$, $DATA\_WAIT = 2X_8$, and $ACK\_WAIT = X_8$. Measured in *seconds*. Defined in the range $(0, \infty)$. Search range is $[0, 1]$.

5.6. Network response metrics

The quality of a given set of controlled factor values was determined by measuring a set of metrics against a simulated network. For a given candidate solution, specified by a set of input controlled factor values for a given network protocol, we define the quality of this candidate solution in terms of a set of network response metrics derived in [16]. Three aspects of solution quality were measured; *performance*, *reliability*, and *efficiency*. Measurement of these metrics was performed through simulation as described in section 4.4.

For each metric $M_1 - M_5$ lower values imply more favourable behaviour. Zero represents optimal solution quality in a given metric, though this value is unlikely to be observed in practice. Where metrics are defined *per hop* or *per metre*, this is to normalise results in the size of the network. This is essential in order that results be comparable between networks of different node count, node distribution in the network space, or physical size. Where metrics are defined *per packet*, this is to normalise results in the volume of traffic handled by the network to enable fair comparison between relatively busy or quiet networks, a property which is not a controlled factor but for which we must account.

Performance metrics:

Network performance is defined in terms of *normalised latency*, which is the average time taken for a packet to traverse unit distance within the network. This is important because in most real-world applications it is not sufficient for a network to guarantee that a packet will eventually be delivered. In real-time applications, such as a typical sensornet application, it is important that packets are delivered within a given deadline. Knowledge of the average latency per unit distance allows the network designer to calculate the physical speed at which data traverses the network.

M_1: **Latency per hop** Mean time for a packet to travel 1 node-node hop. Measured in $hop^{-1}s$. Defined in the range $(0, \infty)$.

M_2: **Latency per metre** Mean time for a packet to travel 1 metre. Measured in $m^{-1}s$. Defined in the range $(0, \infty)$.

Reliability metrics:

Network reliability is defined in terms of *packet delivery*. Ideally, every packet generated by the simulated sensing application and queued for delivery at the source node would eventually reach the destination node within the delivery deadline. The source node and destination nodes are not interested in how this is achieved, or the route taken through the network; these are details that are delegated to the network middleware.

M_3: **Packet delivery failure ratio** Proportion of packets created at source nodes by the simulated application which the network attempted to deliver, but were lost before reaching their intended destination. Unitless. Defined in the range $[0, 1]$.

Efficiency metrics:

Network efficiency is defined by the average *energy* consumed to move data packets within the network. It is generally impossible to define the energy consumed in delivering a specific packet, so an average is obtained for all delivered packets. Unsuccessful delivery attempts also consume energy until all potential delivery branches terminate prior to reaching the destination. For a given network and a given network loading from the distributed application, lowering the energy required to move each packet through unit distance will increase the usable lifetime of the sensornet.

M_4: **Energy per packet per hop** Mean energy for 1 packet to travel 1 node-node hop. Measured in $Jpacket^{-1}hop^{-1}$. Defined in the range $(0, \infty)$.

M_5: **Energy per packet per metre** Mean energy for 1 packet to travel 1 metre. Measured in $Jpacket^{-1}m^{-1}$. Defined in the range $(0, \infty)$.

5.7. Measuring solution quality

The metrics M_1 to M_5 defined above are all mutually independent and may be targeted as individual objectives by sensornet designers. However, real sensornet designs are likely to require an acceptable compromise between multiple competing objectives. It is therefore necessary to define a mechanism by which the relative quality of two or more candidate solutions can be compared to determine which offers the best compromise.

Assume we have n controlled factors X_1-X_n and m metrics M_1-M_m. A candidate solution $S_\alpha = \{X_{\alpha 1}, \ldots, X_{\alpha n}\}$ maps to a set of metrics $T_\alpha = \{M_{\alpha 1}, \ldots, M_{\alpha m}\}$. The mapping of $S \mapsto T$ is not known ***a priori*** but instead is evaluated experimentally as described in section 4 for specific values of S. A perfect solution $S_{perfect}$ would yield a set of metrics $T_{perfect}$ such that $\forall M_i \in T_{perfect} \bullet M_i = 0$. Although $S_{perfect}$ does not necessarily exist, we define the quality measure E in Equation 3 of any given candidate solution S_α based on the Euclidean distance from the point in solution phase space defined by T_α to the point defined by $T_{perfect}$.

$$E = \sqrt[2]{\sum_{i=1}^{m} w_i (s_i M_i)^2} \quad (3)$$

Some network performance attributes may be of greater importance than others to a sensornet designer. We therefore define the weighting w_i for metric M_i such that a larger weighting value indicates a greater importance attached to the associated network behaviour attributes.

Each of the metrics M_1-M_m may be defined over a different range, so it is inappropriate to compare the absolute measured values directly. We define a scaling factor s_i for metric M_i such that all possible values of $s_i M_i$ are found in the range $[0, 1]$, noting that the the ideal value of any given metric is also the lowest possible value, 0. We therefore observe that the maximum Euclidian distance, E_{MAX} for a given set of weightings is given by Equation 4 representing the worst candidate solution. All experimental values of E can be compared against E_{MAX}.

It is only meaningful to compare two E values if all scaling values s_i are equal for each E. If for a given metric M_i is defined over a finite range then the value of s_i is well-defined and does not vary between network configurations under consideration. However, if a given metric M_i is defined over an infinite range then there does not exist a single well-defined value of s_i. Instead, we define s_i in the context of a given set of experimental results by setting $s_i = \frac{1}{MAX(M_i)}$ where $MAX(M_i)$ is the largest value of metric M_i observed during all experiments.

$$E_{MAX} = \sqrt[2]{\sum_{i=1}^{m} w_i} \quad (4)$$

In the experimental work that follows we set all $w_i = 1$ to give equal weighting to all metrics, and set all s_i using the second definition above as some metrics defined in section 5.6 are defined over an infinite range. It follows that all values of E are defined in the range $[0, \sqrt{m}]$ for m metrics where 0 implies the theoretically perfect solution and $\sqrt{m}$ implies the worst solution derivable from observed values.

6. Three-phase experiment implementation

In this section we apply the experimental method defined in section 4 to the protocol tuning problem described in section 5. We label the *TTL-Bounded Gossip* protocol as *A* and the *Implicit Geographic Forwarding* protocol as *B*.

6.1. Phase 1: Variance analysis

We can reduce the cost of the most expensive component, the execution of network simulations, by reducing the period within which the network simulation executes. However, if this period is too small we risk unacceptable levels of experimental error leading to meaningless results. We mitigate this risk by analysing the variance of network metrics

	M_1	M_2	M_3	M_4	M_5
A	19	43	58	49	46
B	27	78	63	51	61

Table 1. Phase 1: τ_{Ci} values for metrics $M_1 - M_5$

with respect to simulated time, calculating the minimum simulated period required for an acceptable and defined level of experimental error.

Metrics are sampled periodically but are influenced by total simulated period from the start to the sampling point. Assuming that the network eventually reaches a steady state, measured metrics converge on the actual value with sample accuracy increasing with simulated time, until sampled values fall within experimental error margin at which point no further improvement is possible. We measured this point for each metric by experiment. It is possible that a metric might appear to converge, but then later diverge. We mitigate this risk by running simulations for at least double the period from the start to the apparent convergence point.

Assume the value of some convergent metric M_α at simulated time τ is given by $M_\alpha(\tau)$. $M_\alpha(\tau)$ approaches its limiting value $M_\alpha(\infty)$ as $\tau \to \infty$. At some simulated time τ_α the value $M_\alpha(\tau_\alpha)$ becomes sufficiently close to $M_\alpha(\infty)$ such that for all $\tau > \tau_\alpha$ the value $M_\alpha(\tau)$ is within $\pm\eta\%$ of $M_\alpha(\infty)$. We define metric M_α as *converged* at this simulated time τ_α. Any further variation, including that deriving from noise and unblocked nuisance factors, is within $\pm\eta\%$ experimental error margin. We set $\eta = 5$ such that measured metrics used in later analysis have $\pm 5\%$ measurement error.

Consider $M = \{M_1, M_2, M_3, M_4, M_5\}$, the set of metrics defined in section 5.6. Table 1 presents τ_α measured experimentally for metrics $M_\alpha \in M$ with each value rounded as $\lceil \tau_\alpha \rceil$. For protocol A, *TTL-Bounded Gossip*, $\forall M_\alpha \in M \bullet \tau_\alpha < 60s$. We therefore select simulation length $\tau_{sim1} = 120s$ for P_1 to allow a safety margin for any anomalous solution instability. For protocol B, *Implicit Geographic Forwarding*, $\forall M_\alpha \in M \bullet \tau_{Ci} < 120s$. We select simulation length $\tau_{sim2} = 240s$ for P_1 to allow a safety margin for any anomalous solution instability.

6.2. Phase 2: Factor significance screening

In Phase 2 we identify which of the protocol controlled factors are the best predictors of the network performance metrics. This requires a small number of points in the parameter space to be sampled in the axis corresponding to each controlled factor, and a set of simulation experiments to be run to measure network performance under each combination. The n-way ANOVA method is applied to assess which controlled factors are significant to the experimental outcomes [14]. Any factors which are deemed statistically insignificant are dropped at this stage.

	M_1	M_2	M_3	M_4	M_5
X_1	0.0565	0.5262	0.2753	0.8355	0.5196
X_2	0.7422	0.3218	0.4093	0.8370	0.9509
X_3	0.3925	0.2663	0.6711	0.7004	0.6048
X_4	0.0000	0.3789	0.0000	0.6036	0.3521
X_5	0.6881	0.0000	0.0000	0.0000	0.0000
X_6	0.0056	0.0315	0.0000	0.1947	0.3795
$X_1 \times X_2$	0.8716	0.5711	0.1924	0.4157	0.9098
$X_1 \times X_3$	0.8779	0.4139	0.9825	0.4967	0.3630
$X_1 \times X_4$	0.0189	0.1029	0.7474	0.7598	0.4974
$X_1 \times X_5$	0.9491	0.7164	0.4740	0.9856	0.5221
$X_1 \times X_6$	0.9787	0.6021	0.7412	0.1371	0.5291
$X_2 \times X_3$	0.2412	0.6825	0.9027	0.9031	0.9039
$X_2 \times X_4$	0.4802	0.7602	0.5331	0.1796	0.1470
$X_2 \times X_5$	0.3899	0.2331	0.3729	0.9500	0.9823
$X_2 \times X_6$	0.7407	0.9634	0.0166	0.4551	0.3499
$X_3 \times X_4$	0.4156	0.4089	0.0737	0.4733	0.5438
$X_3 \times X_5$	0.7441	0.5886	0.7504	0.8016	0.6377
$X_3 \times X_6$	0.8538	0.4826	0.8753	0.5632	0.6450
$X_4 \times X_6$	0.5187	0.5028	0.0000	0.7627	0.8542
$X_4 \times X_5$	0.1707	0.8840	0.0000	0.7280	0.3555
$X_5 \times X_6$	0.0014	0.1860	0.0000	0.3845	0.5717

Table 2. Phase 2: R^2 values for controlled factors X_1-X_6 and their interactions for metrics M_1-M_5 for the TTL-Bounded Gossip protocol

	M_1	M_2	M_3	M_4	M_5
X_2	0.0000	0.9115	0.0097	0.4277	0.5216
X_3	0.7991	0.2797	0.9275	0.9520	0.9890
X_4	0.0000	0.0000	0.0000	0.0000	0.0000
X_5	0.2715	0.2275	0.7656	0.8296	0.7311
X_7	0.0000	0.0000	0.0000	0.0000	0.0000
X_8	0.0000	0.0000	0.0000	0.0000	0.0000
$X_2 \times X_3$	0.9996	0.9892	0.7497	0.9969	0.9936
$X_2 \times X_4$	0.0254	0.9622	0.4034	0.6460	0.7493
$X_2 \times X_5$	0.8217	0.6784	0.8072	0.0623	0.0516
$X_2 \times X_7$	0.7903	0.2650	0.0997	0.9839	0.9658
$X_2 \times X_8$	0.0392	0.6358	0.0907	0.7985	0.8451
$X_3 \times X_4$	0.9386	0.2415	0.9718	0.9952	0.9996
$X_3 \times X_5$	0.8137	0.7883	0.8711	0.4870	0.4245
$X_3 \times X_7$	0.5805	0.3202	0.4790	0.9800	0.9800
$X_3 \times X_8$	0.9058	0.8431	0.8309	0.9626	0.9022
$X_4 \times X_5$	0.7270	0.1178	0.9223	0.9520	0.8842
$X_4 \times X_7$	0.0000	0.0000	0.0000	0.0000	0.0000
$X_4 \times X_8$	0.0000	0.0009	0.0000	0.0000	0.0000
$X_5 \times X_8$	0.8240	0.7159	0.7740	0.9712	0.9335
$X_5 \times X_7$	0.3849	0.0218	0.9782	0.8474	0.5175
$X_7 \times X_8$	0.0000	0.0000	0.0000	0.0000	0.0000

Table 3. Phase 2: R^2 values for controlled factors X_2-X_5 and $X_7 - X_8$ and their interactions for metrics M_1-M_5 for the Implicit Geographic Forwarding protocol

Controlled factors $\{X_1 - X_6\}$ were considered at this stage for protocol *A*. The test suite size was calculated using the formula given in section 4.3 with $p = 6$, $q = 3$, $r = 3$ and $s = 3$. This gives a test suite size of $3 \times 3 \times 3^6 = 6561$, hence 6561 points in the factor-response phase space are available for model fitting. Table 2 presents the R^2 values (the F-value of the ρ-statistic) for each controlled factor, and first-order pairwise interaction between factors. Factors $\{X_4, X_5, X_6\}$ are significant in isolation with 95% confidence ($R^2 < 0.05$) for at least two of the metrics M_1-M_5, and at least one of $\{X_4, X_5, X_6\}$ is evident in almost all interaction pairs deemed significant with 95% confidence. Factors $\{X_1, X_2, X_3\}$ are not significant in isolation for any metric, or as a member of an interaction pair which does not include any of $\{X_4, X_5, X_6\}$. Notably, the protocol-specific factor X_6 is statistically significant indicating that attempts to tune this protocol are appropriate.

Controlled factors $\{X_2 - X_5, X_7 - X_8\}$ were considered at this stage for protocol *B*. The test suite size was calculated using the formula given in section 4.3 with $p = 6$, $q = 3$, $r = 3$ and $s = 3$. This gives a test suite size of $3 \times 3 \times 3^6 = 6561$, hence 6561 points in the factor-response phase space are available for model fitting. Table 3 presents the R^2 values (the F-value of the ρ-statistic) for each controlled factor, and first-order pairwise interaction between factors. Factors $\{X_4, X_7, X_8\}$ are significant in isolation with 99% confidence ($R^2 < 0.01$) for all metrics $M_1 - M_5$. The controlled factor X_2 is significant with 99% confidence ($R^2 < 0.01$) for metric M_1 and significant with 90% confidence ($R^2 < 0.1$) for metric M_3. At least one of $\{X_2, X_4, X_7, X_8\}$ is evident in all interaction pairs deemed significant with at least 95% confidence ($R^2 < 0.05$).

6.3. Phase 3: High resolution modelling

In Phase 3 we sample the parameter space along the corresponding axis in a greater number of points for each statistically significant controlled factor. Again, a set of simulation experiments was performed to measure network performance under each configuration. Phase 2 identifies controllable factors not significant to the outcome for which we can justifiably select any value falling within the ranges explored experimentally. We select the midpoint value of the boundaries defined in section 5 for each controllable factor deemed insignificant by section 6.2.

Factors $\{X_4, X_5, X_6\}$ were considered at this stage for protocol *A*. The test suite size was calculated using the formula given in section 4.3 with $p = 3$, $q = 10$, $r = 3$ and $s = 3$. This gives a test suite size of $3 \times 3 \times 10^3 = 9000$, hence 9000 points in the factor-response phase space are available for model fitting. Factors $\{X_2, X_4, X_7, X_8\}$ were considered at this stage for protocol *B*. The test suite size was calculated using the formula given in section 4.3 with $p = 4$, $q = 7$, $r = 3$ and $s = 3$. This gives a test suite size of $3 \times 3 \times 7^4 = 21609$, hence 21609 points in the factor-response phase space are available for model fitting.

Linear interaction models of the form given by Equation 1 were fitted to the sampled points for protocols A and B, yielding sets of model coefficients C_A and C_B in controlled factors $X_1 - X_8$. Solving these simultaneous equations yields the best-known input protocol tuning values corresponding to A and B are labelled I_A and I_B respectively, given to 4 decimal places in table 4.

We conduct 100 repeats of each network design $\xi_i \in \Xi$ using parameter sets I_A and I_B as defined in table 4, taking the arithmetic mean of metrics M_1-M_5 to obtain results sets O_A and O_B as given in Table 5 to 4 decimal places. Relative quality of O_A and O_B is measured by the E value defined in the interval $[0, \sqrt{5}]$ as described in section 5.7. We normalise E to the range $[0, 1]$ in the rightmost column for convenience. $f(O_B, O_A)$ gives O_B as proportion of O_A for comparison of relative solution quality.

At this stage it is worth highlighting that there is no default or initial tuning against which to compare any other given tuning. A wide range of values were observed for each of the metrics M_1-M_5 during the experiments from which the values given in table 5 were derived, covering the entire spectrum of behaviour from highly effective to highly deficient. It is against these observed extremes that we compare all other observed values as they provide the only meaningful baseline for comparison.

The best tunings of protocols A and B display significantly different behaviour. A outperforms B by an order of magnitude in metrics M_1 and M_2. This is unsurprising as A is stateless and implements its actions as soon as possible, whereas B is based on a state machine with minimum-time guards on some transitions defined in terms of factor X_8. As the performance of network components increases it might be expected that this performance margin will increase unless X_8 is decreased appropriately.

However, B outperforms A in the remaining metrics M_3–M_5. In each case the performance of both protocols is within the same order of magnitude, but is nevertheless significantly different. As we weighted all metrics equally in section 5.7 this is sufficient for $E_B < E_A$ by a significant margin, such that B is found to perform considerably better than A in this experiment. Had we weighted metrics differently this condition would not necessarily hold. For example, placing more emphasis on metrics M_1 and M_2 would favour A over B, eventually reaching the condition $E_A < E_B$.

Consider the classes of metrics defined in section 5.6. If *performance* is the most important issue in a given network then A is a better choice, as the best compromise tuning of B cannot outperform the best compromise tuning of A in metrics $M_1 - M_2$. If *reliability* or *efficiency* is the most important issue in a given network then B is a better choice, as the best compromise tuning of A cannot outperform the best compromise tuning of B in metrics $M_3 - M_5$.

We conclude that the selection of protocol, and the tuning of that protocol, is dependent on the required performance characteristics of a given network. The method we describe in this paper allows the designer to efficiently and fairly compare a selection of candidate protocols, defining the relative priority of each measurable network response metric by setting appropriate weightings in the solution quality metric E, such that the most appropriate tuning of the most appropriate protocol can be obtained.

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
I_A	*5.5000*	*5.5000*	*0.5500*	4.9506	7.5390	0.9999	-	-
I_B	*5.5000*	2.4202	*0.5500*	8.9954	*5.500*	-	41.0607	0.2701

Table 4. Best-known protocol tunings: controlled factors

	M_1	M_2	M_3	M_4	M_5	E	$E \div \sqrt{5}$
O_A	1.4624×10^{-2}	7.2833×10^{-6}	3.5242×10^{-1}	2.2086×10^{-4}	9.6196×10^{-8}	0.35613	0.15927
O_B	1.2388×10^{-1}	5.9283×10^{-5}	2.3837×10^{-1}	7.5106×10^{-5}	3.6026×10^{-8}	0.25881	0.11574
$f(O_B, O_A)$	0.1180	0.1229	1.4785	2.9406	2.6702	1.3761	
Best protocol	A	A	B	B	B	B	

Table 5. Best-known protocol tunings: measured responses

7. Conclusions

In section 3 a set of desired research objectives was defined, against which we now state our findings.

Objective 1: *Define a reusable methodology for addressing the protocol tuning problem*

Section 4 defines a three-phase *Design Of Experiments* methodology based on the *Factorial Design* paradigm. Section 5 defines the format in which controlled factors and measured responses must be specified.

Objective 2: *Identify the significant factor-response relationships for the protocol tuning problem*

Section 6 identifies statistically significant controlled factors for the two dissimilar protocols considered in this paper, and summarises the trends governing the relationships between controlled factors and measured responses.

Objective 3: *Obtain near-optimal compromise solutions to the protocol tuning problem*

Section 6 shows that the experimental method discussed in this paper finds near-optimal solutions to the protocol tuning problem for both protocols considered in this paper and states the values of controlled factors under these tunings.

References

[1] H. Karl and A. Willig, *Protocols And Architectures For Wireless Sensor Networks*. Hoboken, NJ: Wiley, 2005.

[2] A. Baker, S. Dixon, F. Drabble, J. Gibbings, A. Lewkowicz, D. Moffat, and R. Shaw, *The Systematic Experiment*, J. Gibbings, Ed. Cambridge: Cambridge University Press, 1986.

[3] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proceedings of ICASSP 2001*, May 2001, pp. 2033–2036.

[4] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceeedings of ACM/IEEE MobiCom*, 1999, pp. 151–162.

[5] T. He, B. Blum, Q. Cao, J. Stankovic, S. Son, and T. Abdelzaher, "Robust and timely communication over highly dynamic sensor networks," *Real-Time Systems*, vol. 37, no. 3, pp. 261–289, 2007.

[6] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava, "Energy aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–50, March 2002.

[7] M. Totaro and D. Perkins, "Using statistical design of experiments for analyzing mobile ad hoc networks," in *Proceedings of Modeling, Analysis And Simulation Of Wireless And Mobile Systems*, 2005, pp. 159–168.

[8] Z. Haas, J. Halpern, and L. Li, "Gossip-based ad hoc routing," *IEEE/ACM Transactions on Networking*, vol. 14, no. 3, pp. 479–491, 2006.

[9] V. Park and M. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *INFOCOM (3)*, 1997, pp. 1405–1413.

[10] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "Complex behavior at scale: An experimental study of low-power wireless sensor networks," UCLA, Tech. Rep. CSD-TR 02-0013, Feb 2002.

[11] Y. Sasson, D. Cavin, and A. Schiper, "Probabilistic broadcast for flooding in wireless mobile ad hoc networks," in *Proceedings of IEEE Wireless Communications and Networking Conference*, March 2003, pp. 1124 – 1130.

[12] P. Kyasanur, R. Choudhury, and I. Gupta, "Smart Gossip: An adaptive gossip-based broadcasting service for sensor networks," in *Proceedings of IEEE Mobile Ad-hoc and Sensor Systems*, 2006, pp. 91–100.

[13] W. Wood and D. Martin, *Experimental Method*. London: Athlone Press, 1974.

[14] M. Spiegel and L. Stephens, *Theory and Problems of Statistics*. New York: McGraw-Hill, 1998.

[15] J. Tate and I. Bate, "YASS: A scaleable sensornet simulator for large scale experimentation," in *Proceedings of Communicating Process Architectures*, 2008, pp. 411–430.

[16] J. Tate, I. Bate, and S. Poulding, "Tuning protocols to improve the energy efficiency of sensornets," in *Proceedings of the Fourth IET UK Embedded Forum*, 2008.