TUNING PROTOCOLS TO IMPROVE THE ENERGY EFFICIENCY OF SENSORNETS

J. Tate, I. Bate and S. Poulding

Department of Computer Science,
University of York, York, YO10 5DD, United Kingdom
{jt | iain.bate | smp}@cs.york.ac.uk
Fax: +44 1904 434331

Keywords: sensornets, protocols, energy, design

Abstract

Energy efficiency is of critical importance in sensornets where the working life of wireless motes, and consequently the entire network, is limited by the finite energy capacity of batteries. Radio network activity typically dominates the energy consumption profile of motes running distributed applications, and hence represents the obvious target when attempting to use energy more frugally. Significant savings can be obtained by carefully tuning existing energy-ignorant protocols. Current practice in choosing parameters is generally based on experience, intuition, and trial and error. This approach rarely leads to the best choice. In this paper a novel method is presented through which the complex relationships between protocol parameters, network structure, application workload and observed network behaviour are understood and tuned.

1 Introduction

Energy efficiency is of critical importance in sensornets where the working life of wireless motes and consequently the entire network is limited by the finite capacity of batteries. The network must manage its consumption of this non-renewable resource, carefully balancing the competing demands of performance and durability. Radio networking components are typically the greatest consumers of energy, and consequently offer the most significant target for improved energy management efficiency.

Improvements in sensornet hardware will enable greater performance per unit cost. The sensornet designer may choose to “spend” this improvement by providing each mote with greater resources, or by using more motes in projects, or by reducing the cost of the project. In the latter two cases, hardware improvements will be useful to the sensornet designer but will not necessarily allow heavier, higher-overhead protocols to become viable.

Regardless of hardware improvements, network lifetime can be extended by using radio communications more efficiently. Most non-trivial protocols have sets of parameters which can be fine-tuned. One option is to introduce new energy-aware networking protocols which seek to reduce energy consumption by managing network traffic more efficiently. However without suitable parameters the benefits of doing so will not be fully realised. In this paper we explore how existing well-understood energy-ignorant protocols can be tuned for maximal energy efficiency. The protocol, flooding, has been chosen as it is well understood and relatively simple which means the technique and its benefits can be clearly understood. However, the tuning method we describe is equally applicable to both energy-ignorant and energy-aware protocols.

Sensornet motes resources are typically so highly constrained that any energy efficiency improvement is worthy of evaluation. Many small savings achieved over the sensornet lifetime accumulate into large savings. 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 is harder still.

Considering one factor at a time is unlikely to give a thorough treatment of the trade-offs and hence any solution found may differ significantly from the optimal. For instance, suppose we wish to maximise the probability that a packet reaches a given destination within a given deadline. If we increase the *gossip probability* are we helping timely delivery by ensuring that at least one short path from source to destination is traversed, or are we hindering timely delivery by congesting the network? How big should each node’s *waiting packet queue size* be to avoid retaining packets that will inevitably expire prior to delivery but without throwing away packets which should still be viable? Does increasing the *gossip probability* upset this delicate balance by delivering too many irrelevant packets, undermining careful tuning of the *waiting packet* *queue size*?

Where we have multiple controllable factors, each of which can take many values, combinatorial explosion renders exhaustive exploration through experimentation impossible. However, well-established pragmatic approaches can approximate exhaustive exploration in acceptable time. In this paper we propose a method based on full factorial design experiments [3]. For each of f factors we sample the defined range at g evenly-spaced points. A set of design points is established in which every setting of every factor is combined with every setting of every other factor.

Significantly, this rigorous approach takes account of unknown interactions and relationships between controlled factors which remain hidden under the traditional approach of varying only one factor at a time. We identify the most influential protocol factors, and show that the most energy-efficient tunings of energy-ignorant protocols still yield sub-optimal energy consumption. Finally, additional savings achievable by aggressive state management are calculated. To the best of our knowledge this method has not previously been applied to protocol tuning in sensornets.

The structure of the remainder of this paper is as follows. Section 2 examines related work. Section 3 defines the questions addressed by this paper. Section 4 describes the experimental method. Section 5 evaluates the experimental results. Conclusions are presented in section 6.

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* [19].

Optimising sensornets for energy efficient is complex. Raghunathan [19] 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 [23] apply a *systematic statistical design of experiments* approach to evaluate and model the complex tradeoffs in MANET design. This work considers the impact of varying network design with a fixed network application. In contrast, our work considers the impact of varying network protocol and application behaviour for a fixed network design.

Received signal power in sensor networks with short antenna heights falls rapidly, in inverse proportion to r^4 in distance r, due to partial cancellation from ground ray reflection [5]. An interesting consequence of this non-linear relationship is that network routes containing many short-distance hops may be more energy-efficient than routes containing few long-distance hops, albeit at the expense of requiring more intermediate nodes to be awake to forward traffic [13]. Energy savings must come throughout the protocol stack, influencing the design and operation of sensornet applications, networking protocols, network topologies and network tasking. Sending a single bit of information 100m may consume more energy than 1000-3000 CPU instructions [18]. This cost is incurred by the sender, the receiver, and any intermediate nodes along multi-hop paths, which may grow as network node count increases. Energy savings must be driven by energy-aware design throughout the network stack, rather than relying on improvements in hardware technology [19].

Radio modules are usually the most power-hungry component of sensor nodes [5]. Surprisingly, operating radios in *idle* listening mode often provides little power advantage over actively transmitting or receiving, and receiving can consume more energy than sending [19]. Observing the full potential benefit requires radio state to be managed harmoniously with network activity. Significant energy savings can be observed by identifying low-activity periods and rebalancing the energy-performance tradeoff [9]. Reducing transmission power can also reduce energy consumption directly, by consuming less energy per bit transmitted, and indirectly, by localising network activity and reducing collisions and contention [13].

In classic flooding a node broadcasts a packet to each of its neighbours which in turn rebroadcast the packet and so on. Flooding converges in $O(d)$ rounds in the network diameter d [14] where each node has received the packet, with some exceptions [8]. Flooding is utilised by most non-geographical routing protocols [10] and often succeeds where more sophisticated protocols cannot react quickly to rapidly changing networks [17]. Counter-intuitively complex behaviour is observed [8] despite the protocol's simplicity. *Broadcast storms* [16] 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 [14], displaying bimodal behaviour where either hardly any nodes receive the packet, or almost all do [20]. Gossip probabilities in the range [0.6, 0.8] usually, but not always, ensure most nodes receive most packets [10]. Appropriate gossip probability selection is generally difficult, and may need to vary across nodes and time [15]. Other flooding variants include counter-bounded, distance-based and location-based types [16]. Energy-aware gossiping variants exist which turn nodes off at random [11] exploit the fact that overhearing irrelevant communications is a major source of energy consumption [21]. The physical topology of the sensornet strongly influences energy consumption [24], as exploited by the *Smart Gossip* variant [15].

More sophisticated energy-efficiency protocols can employ a variety of underlying techniques, in which nodes maintain online models of network activity and energy metrics. These approaches may attempt to find low-cost routes using links of fixed cost, rather than attacking the link cost itself. *Minimum transmitted energy* protocols select least-cost routes using the average energy consumed in transmitting packets between node pairs as the link cost [4]. *Maximum lifetime energy routing* protocols extend *minimum transmitted energy* protocols by introducing remaining node energy to the link cost function [4]. *Maximum lifetime data gathering* protocols implement energy load-balancing across nodes [12]. *Energy-Aware QoS Routing Protocol* finds least-cost paths conforming to given end-to-end latency requirements [2].

3 Research problem

3.1 Desired research outcomes

Given a set of typical and broadly comparable sensornet configurations, and a typical energy-ignorant network routing protocol, we define the following objectives.

- Obj 1: Identify which controllable factors are most significant in each solution quality metric.
- Obj 2: Identify the best set of values to assign to controlled factors, yielding the best performance for each solution quality metric.
- Obj 3: Identify the set of periods, and hence proportion of time overall, during which further energy savings are possible by jointly managing network traffic and mote radio state.

3.2 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 a single protocol considered representative of a class of similar 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.

TTL-bounded gossiping was selected to represent energy-ignorant, geography-ignorant, stateless protocols [16]. This class of protocols is important because of their simplicity. More complex protocols of higher sophistication 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 [10] which is wasteful if the source and destination are physically close.

Note that in selecting TTL-bounded gossiping for our experiments we make no claims as to the merit of this protocol for any given sensornet application. More specifically, we do not claim that an optimally-configured gossiping protocol will offer superior performance to more recent and more sophisticated alternative protocols.

3.3 Network design

The TTL-bounded gossiping protocol is considered within the context of an aggregated sensing application running on resource-constrained wireless nodes. It is noted though that the routing performed by the protocol has, and uses, no knowledge about the nature of the application nor specific information from the application during operation. That is, the protocol is essentially stateless and unintelligent in its operation. Most traffic consists of packets travelling short routes upward within the aggregation tree structure. To ensure fair comparison between protocol controlled factor configurations we must account for the influence of network design. We characterise networks by a set of network design factors in section 4.1. We implement blocking of these factors by defining a set of networks sharing identical design factors other than differing in physical node position. The specific network design factors described here and in sections 4.1 and 5.1 are not significant, and are selected merely for expedience in demonstrating the method.

A proportion of nodes are randomly selected as clusterhead nodes, periodically exchanging aggregated results with other clusterhead nodes. The remaining nodes periodically send observations to their geographically nearest clusterhead node. Other than this role allocation, each node is identical. Nodes are modelled on the MICA2 mote with similar performance, radio, and energy properties. Nodes are static and distributed randomly within a cube vacuum devoid of obstructions. Radio signal propagation is modelled by the Friis free space model with exponent 2.0 [7]. All antennas have unity gain and are considered isotropic sources.

All networks are simulated in *yass* (Yet Another Sensornet Simulator) [1]. However, the approach presented could be implemented with equivalent results in any sensornet in which protocol factors can be controlled, and solution quality metrics measured.

3.4 Protocol controlled factors

To explore the range of possible protocol configurations it was necessary to define a set of controlled factors against which the measured response could be observed. Each controlled factor represents some property which is independent of the network configuration, but which may be tuned by the network designer to achieve a desired behaviour or to implement some resource usage tradeoff.

1. *Gossip probability*. The probability that upon receiving a packet a given node will attempt to retransmit that packet. Unitless, defined in the range [0,1].
2. *Seen-packet buffer size*. The number of packets received or transmitted by a node of which knowledge is retained in a FIFO buffer. Nodes do not retransmit a previously-transmitted packet if the latter is held in this cache. Measured in *packets*.
3. *Waiting-packet buffer size*. The number of packets queued for transmission or retransmission in a FIFO buffer. Newer packets displace older packets when full. Measured in *packets*.
4. *Initial backoff*. When attempting to begin transmitting 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 *n*th term is the *n*th power of this base value. Measured in *seconds*.
5. *Packet lifetime*. The maximum permitted time for a packet in transit within a network between source and destination. If the lifetime is exceeded before reaching the destination, the packet is dropped. Measured in *seconds*.
6. *Intercluster TTL*. The total number of node-node hops permitted for packets travelling between aggregation clusters. If this TTL is exceeded prior to reaching the destination, the packet is dropped. Intracluster routes from data sources to clusterhead nodes always have a TTL of 1 and are unaffected by this factor. Measured in *hops*.

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.

3.5 Solution quality metrics

The quality of a given set of controlled factor values was determined by measuring a set of metrics against a simulated network. Three aspects of solution quality were considered and measured; *performance, reliability,* and *efficiency*. A fourth quality aspect, *robustness* to variation between networks, was not measured directly but instead addressed indirectly during analysis (see section 4.1).

For each aspect identified as being significant to a greater notion of quality, a set of metrics was defined against which to measure the extent to which a given protocol configuration satisfies this notion of quality when implemented within a simulated network. This set of metrics was reduced to the minimal set deemed sufficient to capture the characteristics of interest to minimise redundancy and experimental overhead [3].

Performance metrics

1. *Latency per hop*. Mean time for a packet to travel 1 node-node hop. Measured in $hop^{-1}s$.
2. *Latency per metre*. Mean time for a packet to travel 1 metre. Measured in $m^{-1}s$.

Reliability metrics

3. *Packet delivery failure ratio*. Proportion of packets created by source nodes that fail to reach their intended destination. Unitless.

Efficiency metrics

4. *Energy per packet per hop*. Mean energy for 1 packet to travel 1 node-node hop. Measured in $J packet^{-1} hop^{-1}$.
5. *Energy per packet per metre*. Mean energy for 1 packet to travel 1 metre. Measured in $J packet^{-1} m^{-1}$.
6. *Optimal energy per packet per hop*. Mean energy for 1 packet to travel 1 node-node hop, assuming optimal radio state management. Measured in $J packet^{-1} hop^{-1}$.
7. *Optimal energy per packet per metre*. Mean energy for 1 packet to travel 1 metre, assuming optimal radio state management. Measured in $J packet^{-1} m^{-1}$.

For each metric the lowest defined value is zero, and lesser values represent more favourable solutions. 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.

If we assume that maximal performance is attainable when all components operate in their highest-performance state, then we can find the upper bound of energy required to attain maximal network performance by assuming that all components operate at all times in their highest energy consumption state. However, if at any point an energy-consuming component could be operated in a lower energy consumption state without harming performance, the lower bound on energy required to attain maximal performance must be lower than the upper bound. Where component utilisation is lower than 100% in a given execution of the simulated network application, there must necessarily be periods when a component is operated at a higher energy consumption level than is necessary. In this paper we obtain this upper bound experimentally for a given activity trace for a given simulated network.

Suppose a perfect component state management policy existed, maintaining all components in minimal energy *off* states at all times, except when higher-energy *on* states are required to perform useful work. The energy consumed by a network whose components were managed by this perfect policy would be the lower bound on energy required to attain maximal performance. We simplify by assuming that the mote radio is the only significant energy consumer for which significant savings are feasible by managing state without degrading application performance.

Real component state management policies cannot predict future requirements with 100% accuracy so will yield energy consumption somewhere between these upper and lower bounds, with better policies attaining values close to the lower bound. In this paper we obtain this lower bound by analysing the activity trace for a given simulated network, and derive the activity trace which would be observed under a perfect component state management policy. Future work will build upon this paper to explore this important problem. Assuming that the most energy-efficient networking protocol configurations are employed, aggressive component state management policies can act yielding behaviour approaching that required to achieve this theoretical lower bound.

Suppose further that an imperfect state management policy switches a component into an *off* state where the maximal performance trace would require this component in an *on* state. Energy consumption would be reduced further, but performance would now be sub-maximal. State management policies may further reduce energy consumption, below the lower bound discussed above, but this will necessarily entail a tradeoff between energy consumption and performance; there is no longer an opportunity for *"something for nothing"* by exploiting redundancy. We do not consider tradeoffs of this type in this paper.

4 Experimental method

4.1 Obtaining time-to-convergence for network design configurations

When each simulation begins the network is in a pristine state with no packets queued for delivery and no packets in transit. Clearly this condition changes quickly as the simulation begins, but the issue remains of determining at which point the metrics can be sampled. If the value of a given metric varies unpredictably over time then it is meaningless to select a single point at which metric sampling occurs. If, however, the value of this given metric converges on a single value then we must determine at what time the metric is sufficiently converged to be considered steady within some margin for experimental noise. We cannot discard the possibility that convergence takes longer in some network designs, so network design and time-to-convergence are considered together.

Metrics are sampled periodically but are influenced by total simulated period from the start to the sampling point. Metrics therefore converge on the actual value, sample accuracy increasing monotonically in simulated time, until sampled values fall within experimental error margin at which point no further improvement is possible.

Assume the value of some convergent metric at time t is given by m_t. m_t approaches its converged value m as $t \rightarrow \infty$. At some time c the value m_c becomes sufficiently close to m such that for all $t > c$ the value m_t is within $\pm n$% of m_c. We define metrics as *converged* at time c. Any further variation, including that deriving from noise and unblocked nuisance factors, is within $\pm n$% experimental error margin. We set $n = 5$ such that measured metrics used in later analysis have ± 5% measurement error.

Assuming that all metrics are sufficiently convergent within some time c, then we need only run simulations for this same c. Running simulations for longer than c merely consumes resources without improving the quality of experimental results. However, it is reasonable to assume that some metrics may converge more quickly than other, as they measure orthogonal aspects of sensornet behaviour. In determining required simulation time we find the slowest-converging pairing of network design and quality metric, add a further safety margin, and round to the next highest integral number of seconds.

We ran a series of experiments in which a fixed set of protocol controlled factors was tested in a variety of network designs with different configurations of network design factors. Here we indirectly address the *robustness* solution quality aspect; if experimental results show similar behaviour in every dissimilar simulated network, then it is reasonable to assume that TTL-bounded gossiping has some degree of robustness to environmental conditions and network design. It follows that any subsequent findings cannot be explained solely as artefacts of a given network design.

For each network design factor considered here we assigned *high* and *low* values where appropriate. Evaluating each possible combination in a factorial design experiment ensures all network design factor interactions are accounted for. During simulation the metric values were sampled periodically, and analysed to determine at which point further variation is within experimental error. We repeated each design point with three networks. Each network was randomly-generated but reused for each design point.

1. *Node count*. Number of nodes in the network, selected from $\{50, 250\}$ *nodes*.
2. *Node density*. Average number of neighbouring nodes with which pairwise communications are possible, selected from $\{5, 25\}$ *nodes*.
3. *Traffic periodicity*. Rate of source node packet production, selected from $\{0.5, 5\}$ *packet* s^{-1}.

4. *Clusterhead ratio*. Proportion of nodes which have the *clusterhead* role, selected from $\{5, 100\}$%.
5. *Node type*. Selected from $\{$MICA2, MICAz$\}$.

Having established convergence time for each quality metric in each network design considered in the factorial design experiment, we define t as the slowest quality metric convergence time observed as described above. We consider t to be valid for any network characterised by a set of network design factor values falling within the boundaries defined above. All following experiments use network designs sharing a single configuration of network design factors, differing only in the physical location of nodes within the network space. This geographical node distribution is random and uniform.

4.2 Obtaining ranges over which controlled factors induce differentiated outcomes

It is necessary to define the ranges within which controlled factors are to be considered, such that any statistical models fitted to the results encompass the behaviour of the *interesting* regions of the landscape, that is those ranges which induce variation in measured response and include the values giving the best possible metric values.

We employed a trial-and-improvement approach, working one factor at a time. A fixed network design was simulated with all but one controlled factor fixed, starting with estimates of reasonable upper and lower bounds. We then iteratively moved the upper and lower bounds, recalculating the effects through simulation, until we arrived at a set of controlled factor ranges with the desired properties. Note that this process is not intended to discover optimal values for controlled values and does not take account of factor interactions. It is merely to define plausible ranges within which interesting values fall, to be explored more thoroughly in later experiments.

4.3 Screening for significant factors

It is reasonable to assume that some controlled factors may be more significant than others. To discover the relative influence of the controlled factors defined in section 3.4 we implemented a factorial design experiment. For each of the six controlled factor we defined three assignable values spread evenly throughout the intervals defined in section 4.2, totalling 3^6 design points. Each design point was assessed through simulation running for the period determined in section 4.1. Blocking of nuisance factors was implemented by randomly generating three sample networks, each displaying the same fixed characteristics within the boundaries defined in section 4.1, and simulating each design point with each network. This gives a total of $3^6 \times 3 = 2187$ data points for model fitting.

Screening experiment results were assessed by fitting linear interaction models for each quality metric. Analysis of variance allowed the most significant factors to be determined statistically. Fitted model relevance was determined examining residual distributions, and comparing observed and predicted values. Relatively insignificant factors were discarded, retaining only the three most significant.

4.4 Deriving fitted models from experimental results

Having discarded insignificant factors, we now focus our resources on those significant factors remaining. We repeat the experiments of section 4.3 but using six assigned values for each controlled factor and six network designs. Using more assigned values provides greater detail and increases the probability that an assigned value lies near any minima or maxima. Using more network designs provides greater blocking of nuisance factors. Each discarded controlled factor not considered in this stage of the experiment is assigned the midvalue of the range defined in section 4.2. Six values for each of three controlled factors gives 6^3 design points. With six network designs, this gives $6^3 \times 6 = 1296$ data points for model fitting.

4.5 Complexity of experiment design

Exhaustive exploration of the protocol configuration space defined in section 3.4 is 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 [3], samples the protocol configuration space at a finite set of points to render tractable the evaluation effort.

Consider the algorithmic complexity of this approach. Assume we define f controlled factors, and sample each at g evenly-spaced points. This sampling defines g^f design points, distributed evenly throughout the protocol configuration space. If we evaluate each design point for each of h networks then we define hg^f test cases.

The test suite size grows in f, g and h, but in a qualitatively different manner for each. Linear growth in h is observed, as the set of design points is simply repeated for each network. Polynomial and exponential growth in g and f respectively are observed because the design matrix defining the design points set can be represented as unit cells within a hypercube. Increasing g increases the length of the hypercube sides, whereas increasing f increases the dimensionality of the hypercube. Therefore, the test suite size is $O(n)$ in h, $O(n^c)$ in g, and $O(c^n)$ in f.

Although the experimental method is NP-hard in the number of controlled factors, f, this is not necessarily problematic in practice. Firstly, in the definition of a protocol there are a finite number of controllable factors, of which only a subset are likely to be of interest to the experimenter. For example, in section 3.4 we define 6 factors for TTL-bounded gossiping and thereby setting an upper bound of 6 for f in our experiments. Secondly, the screening phase of our experiments, defined in section 5.3, further reduces f by weeding-out insignificant controlled factors.

The polynomial growth in g is also managed in the experiment design. Recall from sections 4.3 and 4.4 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 [6]. Extending this to a linear relationship in f factors requires two values of each controlled factor to be represented in the set of design points [3]. We therefore require only that $g \geq 2$, with low and high values of each factor representing the range for which the model is required to predict metric values. Higher values of g obtain better fitted models but with decreasing gains for each additional sampling point, so small values of g work well [3] and minimise simulation cost. Higher-order linear models of order d would require $g \geq d$.

5 Results and evaluation

5.1 Time-to-convergence for network design configurations

Some combinations of network design and quality metric converged in less than 2 seconds, whereas the slowest took just under 7 seconds. We add a 25% safety margin to this longest convergence time and round up to the nearest integral number of seconds, and find that we require each simulation to run for 10 simulated seconds before sampling quality metrics. No combinations of network design and quality metric failed to converge, and thus we consider it appropriate and safe to curtail simulation after this finite simulated period, setting t to 10s.

We set network design characteristics to values falling within the ranges defined in section 4.1 for which this value of t was shown to be valid. Selected values were *node count* of 250 *nodes*, *node density* of 12 *nodes*, *traffic periodicity* of 1.0 *packet* s^{-1}, *clusterhead ratio* of 10%, and *node type* of *MICA*2.

5.2 Controlled factor ranges

Ranges for controlled factors defined in section 3.4 were derived by trial-and-improvement experiments as per section 4.2, with default values employed for discarded uninfluential factors.

1. *Gossip probability*. Experimental range is [0.2, 1], default 0.6.
2. *Seen-packet buffer size*. Experimental range is [1, 10] *packets*, default 5 *packets*.
3. *Waiting-packet buffer size*. Experimental range is [1, 10] *packets*, default 5 *packets*.
4. *Initial backoff*. Experimental range is [0.1, 1] *seconds*, default 0.5 *seconds*.
5. *Packet lifetime*. Experimental range is [0.1, 5] *seconds*, default 2.5 *seconds*.
6. *Intercluster TTL*. Experimental range is [2, 10] *hops*, default 6 *hops*.

5.3 Screening for significant factors

We defined the design points and sample network designs for the factorial design experiment as described in section 4, using preliminary findings described in sections 5.1 and 5.2. Each combination of design point and network design was simulated. For each solution quality metric a linear interaction model was fitted to the raw data points in MATLAB, yielding coefficients for an equation of the form indicated by Equation (1):

$$Y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \beta_{ij} x_i x_j + \varepsilon \qquad (1)$$

where β_0 is a constant, x_i is the ith controlled factor value, β_i is the coefficient for the i controlled factor, β_{ij} is the coefficient for the interaction between x_i and x_j, and ε is the noise term. The response is influenced linearly by each factor and each pairing of potentially interacting factors. The noise term ε distribution is approximately normal so our chosen model is not found inappropriate [22].

Each interaction model was subjected to analysis of variance (ANOVA) [22] tests in MATLAB to determine which factors and interactions were most responsible for response variability. The same three controlled factors, *gossip probability*, *seen-packet cache size*, and *packet lifetime*, were found most significant for each quality metric. This is not unexpected, as these factors restrict network saturation which would otherwise adversely impact all metrics. The p-values for the F-statistics are relatively high and are generally well above 0.05 though below 0.25, suggesting a relatively weak though still significant influence [22]. However, at this stage we are interested only in the relative importance of the controlled factors to decide which to discard for the next stage. Interestingly, the most significant factors are independent of resource provision. This is important in highly resource-constrained sensornets. For example, little benefit would be observed by equipping motes with more memory to enable larger packet buffers.

Numerous nonzero pairwise interactions also gave p-values for F-statistics in this range. Protocol tuning efforts must take account of these interactions, which would otherwise remain hidden under the simplistic approach of considering each factor in isolation. All significant pairwise interactions involved at least one, and usually two, of the controlled factors deemed significant when not part of an interaction term. This reinforces the significance of these factors to the measured responses.

5.4 Deriving fitted models from experimental results

We defined the design points and sample network designs for the factorial design experiment as described in section

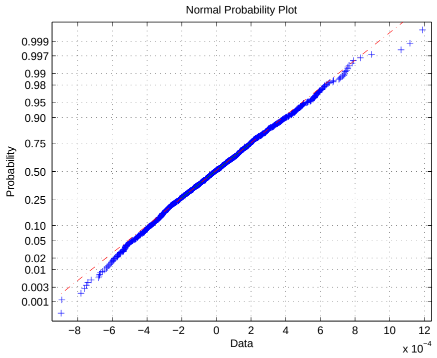

Figure 1: Normal probability plot for the latency per hop metric

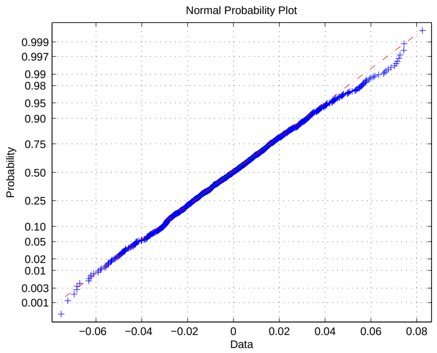

Figure 2: Normal probability plot for the packet delivery failure ratio metric

4, using the preliminary findings described in sections 5.1, 5.2 and 5.3. Using fewer controlled factors but greater detail in those remaining influential factors removes irrelevant and confusing detail, while allowing experimental resources to be deployed more efficiently.

We fitted a linear interaction model to the datapoints in a process similar to that described in section 5.3 for each quality metric. We find that in each case the noise term ε distribution is approximately normal suggesting that our choice of model is not inappropriate, for example as illustrated in Figures 2 and 1. Deviation from the normal distribution is most pronounced at the extremes, and is small for most residuals. We conclude the interaction model is a reasonable approximation for any more detailed model which may better address these extreme values, and removing the less significant controlled factors does not reduce the relevance of this model to this dataset.

Each interaction model was subjected to *analysis of variance* tests to determine factors and interactions most responsible for response variability. Summarised findings of p-values for the F-statistic are presented in Table 1. x_1 represents *gossip probability*, x_2 represents *seen-packet cache size*, and x_3 represents *packet lifetime*. Interactions between factors are represented by multiplication. Most 95% confidence intervals for factors and interactions include zero suggesting that observed results were possible, though not necessarily likely, without these factors and interactions [22]. Interestingly, interactions do not appear as significant in this more-detailed dataset.

The p-value for the x_3 metric in the *latency per hop* factor is given by MATLAB as being exactly zero to within machine precision. The measured response range was small; although the result may be statistically significant, the factor exerts very little actual influence on network behaviour. No controlled factors are greatly influential in *latency per hop*, whereas *latency per metre* appears influenced by *seen-packet cache size* and *packet lifetime* at the 25% and 10% levels respectively. As both latency metrics measure broadly the same effect, we conclude *performance* metrics are unpredictably and weakly influenced by controlled factors [22], such that experimental noise explains most variability.

The situation for *reliability* and *efficiency* metrics is somewhat different. The *gossip probability* and *packet lifetime* factors are more significant than in the screening experiment, whereas the *seen-packet cache size* factor is less significant and can be discarded. We conclude that the earlier apparent significance of *seen-packet cache size* was an artefact of lesser experimental detail of the screening experiments. *Gossip probability* and *packet lifetime* are together sufficient to explain most variation in the *reliability* and *efficiency* metrics not resulting from non-controlled factors and experimental noise.

Problem indicator factors were added to the linear interaction model to determine the importance of network design on measured responses [3]. Regression analysis gave non-zero coefficients for problem indicator factors, with most 95% confidence intervals excluding zero. Models fitted for a given network are good at predicting absolute quality metrics within that network, but may be poor at predicting absolute quality metrics in other networks. However, models fitted for a given network can predict trends across similar networks. A model derived from network N_1 may not predict absolute quality metrics accurately for network N_2 under protocol configuration C_1, but will nevertheless be valid in deciding whether protocol configuration C_1 or C_2 is better in network N_2.

Figures 3 and 4 illustrate the interaction model fitted for *latency per hop energy per packet per metre* metrics respectively, plotted against *gossip probability* and *packet lifetime*. These plots encapsulate the simulated behaviour deemed significant to the derived interaction model.

This experiment stage was repeated with the same 6^3 design points, but using six new networks differing only in node spatial arrangement. Linear interaction models were produced from the resulting data points. Comparing models fitted to the new and original training data sets finds great similarity in coefficients and constant terms, and

Metric	x_1	x_2	x_3	$x_1 \times x_2$	$x_1 \times x_3$	$x_2 \times x_3$
Latency per hop	0.5560	0.7310	0.0000	0.3715	0.7191	0.8622
Latency per metre	0.8162	0.2304	0.1043	0.5469	0.9869	0.7740
Packet delivery failure ratio	0.0818	0.4590	0.1063	0.8840	0.7152	0.8767
Energy per packet per hop	0.0549	0.4644	0.1558	0.9332	0.6480	0.8852
Energy per packet per metre	0.0961	0.5347	0.1786	0.9636	0.7600	0.8679
Energy per packet per hop (optimal policy)	0.0581	0.4698	0.1428	0.9403	0.6672	0.8871
Energy per packet per metre (optimal policy)	0.1033	0.5356	0.1643	0.9678	0.7811	0.8673

Table 1: p-values for the F-statistic for the interaction model fitted to the 3-factor variant

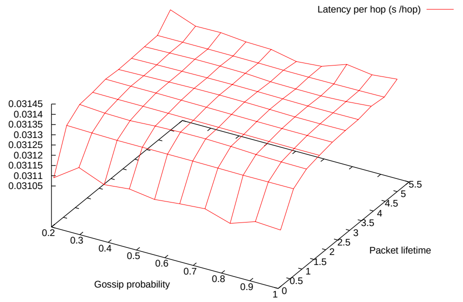

Figure 3: Observed latency per hop vs gossip prob. and packet lifetime

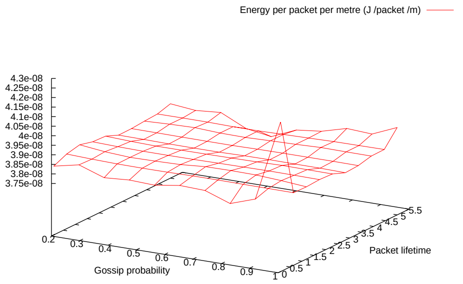

Figure 4: Observed energy per packet per metre vs gossip prob. and packet lifetime

analysis of variance finds similar levels of significance for factors and interactions. From this similarity we conclude that the linear interaction models should remain relevant when applied to other similar networks that do not form part of the original training set.

5.5 Potential energy savings through coordinated radio state management

We compare the *efficiency* metrics for energy consumption under the two extremes of state management policy. Our metrics provide measures of energy consumed per packet in travelling either 1 *hop* or 1 *metre*, under the two extremes of radio state management policy; a null policy in which the radio is never asleep, and an ideal policy with perfect information where the radio is switched off at all times except when useful work is performed. For each of the *per-hop* and *per-metre* measures, if p is the *energy per packet* under the null policy, and q is the *energy per packet*

PSEP	Min	Max	Mean	St. Dev.
r_h	0.4402	0.4548	0.4475	0.0022
r_m	0.4402	0.4548	0.4475	0.0022

Table 2: Summarised results for r_h and r_m

under the ideal policy, we define the measure of potential energy saving, $r = \dfrac{p-q}{p}$. Here, r represents the proportion of observed energy consumption under the null policy which could be saved by applying the perfect policy; r_h is the potential saving *per-hop*, and r_m is the potential saving *per-metre*. Higher r values indicate protocol configurations with greater potential for energy savings.

We calculated r_h and r_m for each combination of design point and network design considered in section 5.4, summarising the results in Table 2 where PSEP refers to Potential Saved Energy Proportion. We observe that r_h and r_m are identical for any given configuration as the TTL-bounded gossiping protocol is geography-ignorant, and consequently optimal radio state at any given time is not dependent on network distances measured in *hops* or *metres*. r values occupy a very small range, with very little variation between values obtained across the set of network configurations. We conclude that r values are largely independent of TTL-bounded gossip controlled factors in networks of the type considered in this paper.

We conclude that any controlled factor configuration taking values from the ranges obtained in 5.2 will yield potential energy savings r_h and r_m within the ranges illustrated in Table 2 when applied to networks of the type described in section 4.1.

6 Conclusions

In this paper an experimental method has been presented for providing near-optimal configurations for routing protocols in wireless sensor networks. The method provides a more thorough consideration of the problem's trade-offs than so-called *one factor at a time* approaches without the computational explosion associated with a full design space exploration. We also believe the a similar method can be applied to other complex problems that may benefit from parameter tuning. We now re-visit the set of desired research outcomes defined in section 3, against which we now state our findings.

Obj 1: *Identify which controllable factors are most signif-*

icant in each solution quality metric.

The method established that *gossip probability* and *packet lifetime* are the most important of the controlled factors considered in this paper, with relatively modest but statistically significant influence. The work also showed the method is also valid for the identification and prediction of trends, and determining an ordering of protocol factor configurations, but is not valid for accurate prediction of absolute metric values due to the stochastic nature of sensornets.

Obj 2: *Identify the best set of values to assign to controlled factors, yielding the best performance for each solution quality metric.*

For each quality metric a three-factor linear interaction model was obtained by regression analysis on simulation experiment results, from which the best set of parameter values can be deduced. Although tuning techniques find best configurations of a given networking protocol in a given network, sensornet designers must ensure that the model coefficients are relevant to the network under consideration.

Obj 3: *Identify the set of periods, and hence proportion of time overall, during which further energy savings are possible by jointly managing network traffic and mote radio state.*

Results suggest a perfect routing policy (with perfect state management and clairvoyance of network loads) could achieve energy savings of around 45% compared to the tuned gossiping protocol. Realistic non-perfect state management policies should realise energy savings between these bounds.

Future work will apply this method to other lightweight sensornet protocols to derive similar models and then use this information in the design of aggressive energy management policies.

References

[1] yass: Yet Another Sensornet Simulator project web pages. http://www.cs.york.ac.uk/rts/yass/, 2008.

[2] K. Akkaya and M. Younis. An energy-aware QoS routing protocol for wireless sensor networks. In *International Conference on Distributed Computing Systems*, pages 710–715, 2003.

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

[4] J. Chang and L. Tassiulas. Maximum lifetime routing in wireless sensor networks. In *Advanced Telecommunications and Information Networks Distribution Research Program, College Park, MD*, pages 609–619, March 2000.

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

[6] Euclid. *The Thirteen Books of The Elements*, volume 1. Dover, 2000.

[7] H. Friis. A note on a simple transmission formula. *Proceedings of the IRE*, 34(5):254–256, May 1946.

[8] 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. Technical Report CSD-TR 02-0013, UCLA, Feb 2002.

[9] R. Golding, P. Bosch, C. Staelin, T. Sullivan, and J. Wilkes. Idleness is not sloth. In *USENIX Winter*, 1995.

[10] Z. Haas, J. Halpern, and L. Li. Gossip-based ad hoc routing. *IEEE/ACM Transactions on Networking*, 14(3):479–491, 2006.

[11] X. Hou, D. Tipper, D. Tupho, and J. Kabara. Gossip-based sleep protocol (GSP) for energy efficient routing in wireless ad hoc networks. In *IEEE Wireless Communications and Networking Conference*, volume 3, pages 1305 – 1310, March 2003.

[12] K. Kalpakis, K. Dasgupta, and P. Namjoshi. Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks. *Computer Networks*, 42(6):697–716, 2003.

[13] V. Kawadia and P. Kumar. Power control and clustering in ad hoc networks. In *INFOCOM*, 2003.

[14] J. Kulik, W. Heinzelman, and H. Balakrishnan. Negotiation-based protocols for disseminating information in wireless sensor networks. *Wireless Networks*, 8(2-3):169–185, 2002.

[15] P. Kyasanur, R. Choudhury, and I. Gupta. Smart Gossip: An adaptive gossip-based broadcasting service for sensor networks. In *Mobile Adhoc and Sensor Sysetems (MASS)*, pages 91–100, 2006.

[16] 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*, pages 151–162, 1999.

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

[18] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Commun. ACM*, 43(5):51–58, 2000.

[19] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava. Energy aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19(2):40–50, March 2002.

[20] 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 (WCNC 2003)*, pages 1124 – 1130, March 2003.

[21] S. Singh and C. Raghavendra. PAMAS - power aware multi-access protocol with signalling for ad hoc networks. *SIGCOMM Computer Communications Review*, 28(3):5–26, 1998.

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

[23] M. Totaro and D. Perkins. Using statistical design of experiments for analyzing mobile ad hoc networks. In *MSWiM ’05: Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 159–168, 2005.

[24] D. Yupho and J. Kabara. The effect of physical topology on wireless sensor network lifetime. *Journal of Networks*, 2(5):14–23, September 2007.