Heartbeat design for energy-aware IoT: are your sensors alive?

Gyamfi, S., Brusey, J., Gaura, E. & Wilkins, R.

Original citation & hyperlink:
Gyamfi, S, Brusey, J, Gaura, E & Wilkins, R 2019, 'Heartbeat design for energy-aware IoT: are your sensors alive?', Expert Systems with Applications, vol. 128, pp. 124-139. https://dx.doi.org/10.1016/j.eswa.2019.03.022

DOI 10.1016/j.eswa.2019.03.022
ISSN 0957-4174

Publisher: Elsevier

Heartbeat design for energy-aware IoT: are your sensors alive?

Kojo Sarfo Gyamfi[a,*], James Brusey[a], Elena Gaura[a], Ross Wilkins[b]

[a] *Faculty of Engineering and Computing, Coventry University, Coventry, CV1 5FB, United Kingdom*
[b] *SAIC Motor UK, Birmingham, B31 2BQ, United Kingdom*

Abstract

A number of algorithms now exist for using model-based prediction at the sensor node of a wireless sensor network (WSN) to enable a dramatic reduction in transmission rates, and thus save energy at the sensor node. These approaches, however, sometimes reduce the rate so substantially as to make the health state of the network opaque. One solution is to include a regular heartbeat transmission whose receipt or otherwise informs the sink about the health state of the node. However, given that a large period increases the probability that dead nodes go unnoticed at the sink, while a small period likely increases the energy cost of communication, what should be the period of the heartbeat transmission? In this paper, we examine the use of heartbeats in WSN design. We derive a general protocol for optimal and dynamic heartbeat transmission by minimising the Bayes risk, which is the expected cost of missing data from dead nodes plus the energy cost of heartbeat transmissions. Our proposed algorithm is dynamic in the sense that the heartbeat period is updated as time goes on and node failures become more probable. We validate our design experimentally using three real-world datasets, and show a 36% reduction in the total heartbeat operational cost over a heartbeat transmission with a fixed period; the results also highlight the superiority of our algorithm over arbitrarily chosen heartbeat periods in different WSN settings, thus promising significant cost savings in WSN

*Corresponding author:
Email addresses: ac9085@coventry.ac.uk (Kojo Sarfo Gyamfi), j.brusey@coventry.ac.uk (James Brusey), csx216@coventry.ac.uk (Elena Gaura), ross.wilkins87@gmail.com (Ross Wilkins)

applications.

Keywords: Internet of things, wireless sensor networks, failure detection, heartbeat transmission, edge mining

---

1. Introduction

The Internet of Things (IoT) is enabled to a large extent by wireless sensor networks (WSNs) deployed to monitor and potentially improve understanding of environments and phenomena (Jin et al., 2014; Bellavista et al., 2013). Domain scientists need robust and reliable data to allow them to find patterns and confirm hypotheses about how monitored phenomena evolve over time; moreover, industrial technicians need timely and reliable data to ensure process or system parameters remain within certain bounds.

The critical factor in the design of wireless sensor network systems is the energy cost of communicating the data (Rosset et al., 2017). Although other components such as sensing and processing play a part in the energy budget, these are typically much lower than the energy cost of communication. For example, a comparison of power requirements for a range of WSN motes and components by Polastre et al. (2005) shows that, for the commonly used Telos platform, the power required to operate the radio is approximately ten times greater than that required to operate the CPU. Therefore, reducing the energy cost associated with communication will substantially reduce the overall energy usage. Reducing the energy usage will in turn lengthen the time that a system can be left unattended (without battery changes), and thus enable many WSN applications that would otherwise be infeasible.

Many WSN applications now employ functional modes involving data compression at the node in order to reduce the energy cost of transmission (Barr & Asanović, 2006; Gaura et al., 2013; Tulone & Madden, 2006). Such compression can be done on a packet-by-packet basis. However, this is not worthwhile since, by and large, packets tend to be quite small, and thus compressing individual packets will yield only a small (if any) saving (Barr & Asanović, 2006). Alternatively, while data compression may be achieved by aggregation (over time) of several packets into one, this comes at the cost of reducing the timeliness of the data. Again, the energy saving may still be minor, especially if data has already been compressed at the application layer (Barr & Asanović, 2006).

One promising way of reducing the frequency of data transmission, and hence the energy cost of communication, is through predictive data reduction. This approach aims to transmit only unexpected (and thus informative) messages and to suppress the transmission of expected, predictable and thus uninformative ones. For a example, in the simple case of a WSN deployed to monitor the status of a normally-closed window, the node may transmit messages only when the window is opened, rather than transmit the status of the window at every sampling period; on the non-receipt of a message, the sink assumes that the window remains closed. This leads to an irregular and somewhat unpredictable frequency of transmission. In some cases, in particular with Bare Necessities (Gaura et al., 2011, 2013), there might be only a few transmissions per year.

While such an approach significantly reduces the energy cost of communication and increases the longevity of the network, it poses a significant risk for successful deployment of WSNs. Since sensor network deployments are subject to failure—batteries can be exhausted, hardware can be tampered with or stolen, electronics may fail —how does the end-user know that sensor nodes are no longer operational or that messages are merely being suppressed?

A common approach is to ensure each node transmits a regular *heartbeat* message, whose purpose is to inform the sink about the health state of the node. However, the context of wireless sensing poses some particular problems and opportunities for the design of heartbeat transmission. In particular, what information should a heartbeat contain? What happens if a heartbeat message is lost? How frequently should heartbeats be sent?

This paper primarily concerns itself with the latter issue, i.e., the frequency of heartbeat transmissions. This is important because a small heartbeat period likely increases the energy cost of heartbeat transmissions, while a large heartbeat period increases the probability that dead nodes will go unnoticed at the sink. In this regard, our main contributions in this paper are as follows:

1. We derive the optimal heartbeat period for monitoring IoTs with irregular transmission based on a Bayes risk minimisation in terms of two key parameters: the probability of node failure over time and the probability of environment change over time.
2. We propose a heartbeat transmission protocol based on the optimal heartbeat period.

The above contributions are detailed in Section 3. We experimentally validate our theoretical results in Section 4 using three real-world datasets obtained from temperature and humidity monitoring applications (Wilkins, 2015). The datasets are overlaid by Weibull failure processes (Azharuddin et al., 2015; Lee et al., 2008) under different failure rates and environmental change probabilities. The discussions of the experimental results are given in Section 5, and we conclude the paper and outline directions for future work in Section 6.

2. Related work

A number of algorithms employing predictive data reduction have been proposed, including Ken (Chu et al., 2006), Probabilistic Adaptable Query system (PAQ) (Tulone & Madden, 2006), Similarity-based Adaptive Framework (SAF) (Bakhtiar et al., 2012), Dual Kalman Filter (DKF) (Santini & Romer, 2006), Derivative-based Prediction (DBP) (Raza et al., 2012), Bare Necessities (BN) (Gaura et al., 2011, 2013), and Linear Spanish Inquisition Protocol (L-SIP) (Goldsmith & Brusey, 2010; Gaura et al., 2013).

These algorithms follow a general form similar to that shown in Algorithm 1. They begin each sensing cycle in Line 1 by obtaining a state vector $\mathbf{x}_{node}$, which is a vector of sensor readings of the phenomena being monitored, such as temperature or humidity. Some algorithms optionally include a filter to smoothen the sensed signal $\mathbf{x}_{node}$ prior to transmission: Line 2. Typically the filter is recursive, in that it uses the last state estimate as the summary of all past sensor measurements, and could be implemented as, for example, an exponentially weighted moving average (Gaura et al., 2011) or a discrete Kalman filter (Santini & Romer, 2006).

The node then makes an estimate of the state vector as known at the sink in Line 3; this estimate is denoted as $\mathbf{x}_{sink}$ and is referred to as the "sink state". The estimation procedure is carried out simultaneously at the sink, and it involves making a prediction on the evolution of the state vector $\mathbf{x}_{node}$ via methods such as naive prediction, linear extrapolation, least mean squares or ARIMA (Aderohunmu et al., 2013; Santini & Romer, 2006; Tulone & Madden, 2006; Gaura et al., 2013). For example, a constant (or naive) state model might be appropriate for monitoring whether a window is opened or closed, whereas a linear model might be more appropriate for representing a low-frequency signal such as the consumption of electricity over time. Often, either a constant state or a linear model is appropriate when the phenomenon

Algorithm 1 Generic algorithm for predictive data reduction

At each sensing cycle:

1: Obtain vector of sensor readings.
2: Filter the vector of sensor readings.
3: Update or predict sink's estimate of the vector of sensor readings.
4: **if** New state is significantly different from the sink estimate **then**
5: Transmit new state.
6: On acknowledgement, update local copy of sink state.
7: **else**
8: Do not transmit. Sink assumes that new state vector is not significantly different from its own estimate.
9: **end if**

is not well understood, as these models make few assumptions. On the whole, the choice of the prediction model, as well as the filter option, tends to be application-specific. The models are chosen to obtain an optimal trade-off between the computational burden on the node and the data accuracy required in the application.

In Step 4, the node compares the state vector $\mathbf{x}_{node}$ with the sink state $\mathbf{x}_{sink}$, and determines if they are significantly different or not. Whether $\mathbf{x}_{node}$ and $\mathbf{x}_{sink}$ are significantly different depends on the accuracy required by the application. Suppose that a maximum error tolerance of ϵ is pre-specified for a given application, then Step 4 essentially evaluates whether $\|\mathbf{x}_{node} - \mathbf{x}_{sink}\|^2 \geq \epsilon^2$. For example, ϵ can be 0.5 degrees Celsius for a temperature monitoring application or 2% for humidity monitoring. For event detection WSNs (Akyildiz et al., 2002; Yu et al., 2005; Sarigiannidis et al., 2015; Bahrepour et al., 2010), the data can be suppressed for most of the time until the process being monitored goes outside some predefined normal or acceptable range. Therefore, the condition in Line 4 evaluates whether or not $\mathbf{x}_{node}\| \geq x_u$ or $\mathbf{x}_{node}\| \leq x_l$, where x_l and x_u are lower and upper bounds respectively on $\|\mathbf{x}_{node}\|$.

If $\|\mathbf{x}_{node} - \mathbf{x}_{sink}\|^2 \geq \epsilon^2$ (for predictive data reduction) or $\mathbf{x}_{node}\| \geq x_u$ or $\mathbf{x}_{node}\| \leq x_l$ (for event detection), then $\mathbf{x}_{node}$ is transmitted, as indicated in Line 5. Once transmission has been acknowledged in Line 7, the local node's record of $\mathbf{x}_{sink}$ is updated.

It is important to note that updating the sink state should only occur after the transmission has been acknowledged, i.e., after the data has been

stored in a permanent datastore at the sink. This ensures that the signal can be reconstructed with minimal error by the sink. Key advantages of Algorithm 1 are genericity and simplicity. It is general in the sense that it is largely agnostic regarding the low-level wireless communication transmission protocol, avoids neighbour interaction, and can be applied to a wide variety of sensed phenomena types. It is also simple in the sense that the algorithm does not increase risk of failure by relying on neighbouring nodes.

Transmission suppression, such as that described in Algorithm 1 and particularly with Bare Necessities (Gaura et al., 2011), introduces the problem that the health of the network is less immediately apparent. This is because, when no message is received from a particular node, it is unclear whether a transmission was attempted (and lost), no transmission was attempted (i.e., it was suppressed), or some other failure has occurred. L-SIP (Goldsmith & Brusey, 2010) relies on end-to-end acknowledgement to ensure that data loss is minimised in the case of a dropped packet. This goes some way to resolving part of the ambiguity because, unless there is a significant problem with the network, dropped packets will cause retries and the packet will soon arrive. Nonetheless, without a heartbeat mechanism, there is no way to distinguish between suppressed transmissions and failures.

Failures are common and documented. Sensor networks fail for a variety of reasons such as: software defects, Byzantine failures, theft or tampering, server failure, hardware failure and battery depletion. Nevertheless, for modern sensor networks, battery depletion is becoming a less important problem (Akyildiz et al., 2002). This is due to the fact that battery life can be extended, e.g., by transmission suppression, better protocols, more efficient hardware, energy harvesting, etc. Moreover, battery exhaustion is generally predictable and easily monitored (e.g., along with other sensor parameters), and hence, battery replacement can be scheduled. How then are we to deal with unpredictable node failures other than battery depletion?

Lau et al. (2014) propose a failure detection method based on the Naive Bayes classifier by monitoring the end-to-end transmission time of the network. This approach involves neighbourhood interaction, and relieves the node of the computational burden, as the failure detection processing is carried out at the sink. However, the method is not developed in the framework of transmission suppression, and there is no distinction made of node failures and suppressed data, when no data is received at the sink.

Vigilance (Schoellhammer, 2010) is an example of an approach to deal with failure that imputes missing data and provides an estimate of uncer-

tainty for each measurement. This estimate provides a basis for reducing the amount of maintenance required and increasing the overall yield. Silberstein et al. (2007) provide one of the few approaches to handling failures alongside transmission suppression. This approach called BaySail (Silberstein et al., 2007), like Vigilance, provides estimates of the missing data at the sink (whether they are failures or suppressions), but does so using knowledge of the suppression scheme in Bayesian inference. While minimising maintenance interventions is often a worthwhile aim because it can be costly or destructive, Vigilance and BaySail do not focus on the detection of when node failures occur. Hence they are unable to identify when a maintenance intervention is necessary in a timely manner, especially in the monitoring of complex phenomena.

To detect whether a node failure has occurred, a heartbeat message can be incorporated in the transmission suppression protocol. For example, in Memento (Rost & Balakrishnan, 2006), the nodes send heartbeat beacons at a given frequency. Rost & Balakrishnan (2006) give the example of 16 heartbeats per network 'sweep' to achieve a false positive rate of 1% with the Direct-Heartbeat approach for a given WSN environment). The sink therefore has to wait for some threshold number of missed receipts, which is the heartbeat period, before considering the node dead, if the heartbeat is not received.

The concept of using heartbeat (HB) messages for fault detection and reliability assurance has been around for some time. For example, Foster (1995) developed the Globus Heartbeat Monitor in the context of network connected distributed processes. This mechanism uses heartbeat periods of around $2 - 9s$. Adapting the heartbeat period depending on conditions such as network and processing delay (Noor et al., 2012) helps reduce the time it takes to identify a fault while keeping network congestion low. For mobile systems, heartbeats can be used to keep track of which servers are in range of a mobile device, such as a tablet moving about a busy hospital (Johnsson et al., 2017). These HB periods are relatively small (about $2s$, by default).

Some IoT devices contain complete GNU/Linux servers and thus can consider fault tolerant approaches that might be more appropriate for larger servers (Celesti et al., 2017). For smaller devices, it is still possible to use a form of lightweight container, such as the *strips* approach used by Su et al. (2014). The *strips* approach allows failure recovery, say, on a neighboring node. However, lightweight devices tend to need specialist protocols, such as Zero Message Queue (ZMQ) (Meng et al., 2017), which is a lightweight

message queue protocol. In ZMQ, heartbeat transmissions are peer-to-peer and are used to identify when to restart a connection, on the assumption that any connection loss is only temporary. Despite being lightweight, ZMQ does not consider the energy cost of heartbeat transmissions.

Heartbeat transmissions have also been employed as a security measure (Pongle & Chavan, 2015; Mangelkar et al., 2018) to counter a selective-forwarding attack.

While heartbeats have become a common inclusion in the implementation of WSNs (Abbasi et al., 2007; Yu et al., 2007; Demirbas et al., 2004; Gupta & Younis, 2003; Fasolo et al., 2007; Wilkins, 2015), the design of an optimal heartbeat period is little studied, especially in the context of transmission suppression. Mainly, the existing approaches empirically determine the heartbeat period beyond which the non-receipt of a message indicates a node failure, for a given WSN environment. In terms of a principled procedure for determining the heartbeat period, we identify the works by Rost & Balakrishnan (2006) and Noor et al. (2012) as those closest to ours.

The method proposed by Rost & Balakrishnan (2006) is based on a one-tailed version of Chebyshev's inequality; this procedure, known as *Variance Bound* (VB), requires the user to specify a required false positive rate FP_{req}. Variance Bound involves the computation of the mean and standard deviations of the interval between consecutive heartbeat receipts; thus, it is a dynamic heartbeat transmission protocol, in the sense that, as new estimates of the mean and standard deviation are obtained, the optimal heartbeat period is updated. However, this procedure relies only on the statistical distribution of heartbeat intervals, rather than on the WSN characteristics including the node failure distribution or the transmission suppression scheme. Moreover, in order to guarantee the require false positive rate, Chebyshev's inequality does not give a tight bound, therefore increasing the length of time dead nodes go unnoticed.

The method proposed by (Noor et al., 2012) is another adaptive heartbeat transmission procedure, known as affirmative adaptive failure detector (AAFD). AAFD similarly considers the inter-arrival times between heartbeats. The methods keeps a list S of inter-arrival times to compute a metric to estimate the expected arrival time for the next heartbeat transmission. Similar to the Variance Bound detector, the AAFD heartbeat transmission is sub-optimal because no consideration is made of the characteristics of the WSN itself.

Nevertheless, the design of an optimal heartbeat period is important in

order to minimise the length of time dead nodes go unnoticed as well as minimise the frequency of heartbeat transmissions, since a high heartbeat transmission rate tends to offset the energy gains made by transmission suppression. Thus, our aim is to utilise the WSN characteristics to find a general heartbeat protocol applicable to any arbitrary WSN environment, that achieves an optimal trade-off between minimising the false positive rate and the missed detection probability.

3. Optimal heartbeat design

3.1. Sending heartbeat packets

A potential problem with reducing packet transmission is that a node may be silent *either* because it is working normally and suppressing messages *or* because it has failed (and is no longer capable of transmission). The end user does not know which one. A solution is to periodically transmit a "heartbeat" packet to let the sink know that the node is still operating. The sink is made aware of the periodicity of the heartbeat message, so that when no such message is received after the predefined period, the sink assumes a node failure has occurred. This leaves open the question of how frequently the heartbeat must be transmitted. If a heartbeat is transmitted too frequently, then the energy cost of communication increases, so that the node's energy budget is quickly depleted. Consequently, interventions may need to be taken at the node in order to replenish its energy budget. On the other hand, if a heartbeat is sent too infrequently, then, in the event that a node fails before the heartbeat period is up, the dead node may go unnoticed at the sink for a long time. Such a scenario may lead to the loss of important data from the phenomenon that is being monitored.

The optimal design of a heartbeat protocol should therefore minimise the frequency at which the heartbeat messages are sent while, at the same time, minimising the length of time that dead nodes go unnoticed. In this paper, we formulate this problem as a Bayes risk minimisation, where the Bayes risk is the expected cost of missing data from dead nodes and the energy cost of heartbeat transmissions.

3.2. Bayes risk minimisation

For any given node in the WSN, let R_n represent the event that a message is received at the sink at time step n, and $\bar{R}_n$ represent the event that no message is received at the sink at time step n. In the generic algorithm for

predictive data reduction as given in Algorithm 1, the node transmits data when the new state has changed significantly from the estimate of the sink state, assuming there is no node failure. Thus, knowledge of the change of state (or equivalently, the rate of transmission suppression) informs the sink as to the probability of R_n or $\bar{R}_n$. To this end, let C_n be the event that a significant change of system state (as given in Line 4 of Algorithm 1) has occurred, and $\bar{C}_n$, the event that such a change has not occurred. Also, let F_n be the event that a sensor node has failed by time step n, and $\bar{F}_n$, the event that no node failure has occurred. Then, the behaviour of the sensor network can be summarised by the following probability relationships:

1. If a node has failed, no message will be received at the sink. Therefore,

$$P(\bar{R}_n|F_n) = 1 \tag{1}$$

2. If the state has changed and the node has not failed, then a message will be received at the sink[1]. Thus,

$$P(\bar{R}_n|C_n, \bar{F}_n) = 0 \tag{2}$$

3. Conversely, if the state has not changed, then no message should be sent, and hence none received at the sink, whether or not the node has failed. This implies that:

$$P(\bar{R}_n|\bar{C}_n, \bar{F}_n) = 1, \quad \text{and} \quad P(\bar{R}_n|\bar{C}_n, F_n) = 1 \tag{3}$$

In order to determine the optimal heartbeat period, we first consider a WSN functional mode that does not involve heartbeat transmissions. We let $n = l$ be the last time step at which a message is received at the sink. Then, we seek to find the node silent time s, such that, following s periods of no message receipts, the sink is confident that a node failure has occurred by time step $n = l + s$. For the special case of $s = 1$, the sink decides that a failure has occurred every time it does not receive data; such a value of s would be impractical for a WSN employing transmission suppression, as the sink tends to consider all suppressions as node failures. This suggests that s has to be ideally greater than 1, but not so large that dead nodes go unnoticed for too long.

[1]For simplicity, it is assumed there are no other sources of failure, such as transmission loss.

The sink makes a decision denoted d_1, if a failure has occurred, and it makes the decision denoted d_0, if no failure has occurred. Moreover, the sink incurs a cost c_{01} when there is a failure, (i.e., when F_n is true) but it makes the decision d_0 that there is no node failure; this cost c_{01} is the cost of missing important data from the monitored phenomenon as dead nodes go unnoticed, and is likely to be incurred if s is too large. Similarly, the sink incurs a cost c_{10} when it decides d_1 that there is a node failure, when there is no node failure (i.e., when $\bar{F}_n$ is true); this cost c_{10} is associated with the unavailing cost of transportation or other such interventions required to verify the status of a perfectly operating node, and it is likely to be incurred when s is too small.

Having these costs now permit us to express the Bayes risk $\mathcal{B}$ as:

$$\mathcal{B} = c_{00}P(d_0, \bar{F}_n) + c_{10}P(d_1, \bar{F}_n) + c_{01}P(d_0, F_n) + c_{11}P(d_1, F_n) \qquad (4)$$

where the costs c_{11} and c_{00} are the costs of making correct decisions, i.e., deciding that there is a node failure after s periods of no receipts when F_n is true, or deciding that there is no node failure after s periods of no receipts when $\bar{F}_n$ is true. These costs (c_{11} and c_{00}) are often conveniently zero, so that correct decisions are not penalised. In general, however, the four cost variables should be such that $c_{01} - c_{11} > 0$ and $c_{10} - c_{00} > 0$, so that there is a higher penalty associated with making incorrect decisions than with making correct decisions.

It is worth noting that the operational costs $(c_{00}, c_{01}, c_{10}, c_{11})$ can only be appropriately defined by the end-users (possibly in terms of its monetary values), depending on the WSN application.

The general design procedure is then to find an optimal value of s, denoted as s^*, for which the Bayes risk is minimum. Intuitively, the penalty c_{01} stops s^* from being too large, while c_{10} stops s^* from being too small, if the Bayes risk is to be minimised. If we were now to consider a WSN functional mode where heartbeat messages are sent, this optimal value s^* should then correspond to the optimal heartbeat period, so that the node does not send heartbeat messages so frequently that it incurs a lot of energy cost due to communication, or so infrequently that dead nodes go unnoticed for a long time.

Notice that (4) can be rewritten as:

$$\mathcal{B} = \Big[c_{00}P(d_0|\bar{F}_n) + c_{10}P(d_1|\bar{F}_n)\Big]P(\bar{F}_n) + \Big[c_{01}P(d_0|F_n) + c_{11}P(d_1|F_n)\Big]P(F_n), \tag{5}$$

which can be further expanded as:

$$\mathcal{B} = \Big[c_{00}\big(1 - P(d_1|\bar{F}_n)\big) + c_{10}P(d_1|\bar{F}_n)\Big]P(\bar{F}_n) + \Big[c_{01}\big(1 - P(d_1|F_n)\big) + c_{11}P(d_1|F_n)\Big]P(F_n), \tag{6}$$

i.e., in terms of d_1 only, thus yielding the following expression for the Bayes risk:

$$\mathcal{B} = c_{00}P(\bar{F}_n) + (c_{10} - c_{00})P(d_1|\bar{F}_n)P(\bar{F}_n) + c_{01}P(F_n) + (c_{11} - c_{01})P(d_1|F_n)P(F_n). \tag{7}$$

Let $\mathcal{S}$ be the observation space within which the random variable s occurs. $\mathcal{S}$ is partitioned into two regions $\mathcal{S}_1$ and $\mathcal{S}_0$ such that, if $s \in \mathcal{S}_1$, we make the decision d_1 that there is a node failure at time $n = l + s$, and if $s \in \mathcal{S}_0$, we decide d_0 that there is no node failure at time $n = l + s$. Then, by expressing (7) in terms of the node silent time s that we seek to optimise, we obtain the following:

$$\mathcal{B} = c_{00}P(\bar{F}_{l+s}) + c_{01}P(F_{l+s}) + \int_{\mathcal{S}_1}\Big[(c_{10} - c_{00})P(s|\bar{F}_{l+s})P(\bar{F}_{l+s}) - (c_{01} - c_{11})P(s|F_{l+s})P(F_{l+s})\Big]ds, \tag{8}$$

The Bayes risk, as given by (8), can only be minimised (Cohn & Melsa, 1980) when the decision region $\mathcal{S}_1$ is chosen such that,

$$(c_{10} - c_{00})P(s|\bar{F}_{l+s})P(\bar{F}_{l+s}) - (c_{01} - c_{11})P(s|F_{l+s})P(F_{l+s}) < 0, \tag{9}$$

which can equivalently be rewritten as:

$$\frac{P(s|F_{l+s})P(F_{l+s})}{P(s|\bar{F}_{l+s})P(\bar{F}_{l+s})} > \frac{(c_{10} - c_{00})}{(c_{01} - c_{11})}. \tag{10}$$

Using Bayes rule, the relation given in (10) for choosing the decision region $\mathcal{S}_1$ can be expressed in terms of the posteriors thus:

$$\frac{P(F_{l+s}|s)P(s)}{P(\bar{F}_{l+s}|s)P(s)} > \frac{(c_{10}-c_{00})}{(c_{01}-c_{11})}, \tag{11}$$

since the node failure probability can be conveniently modelled after standard failure processes such as the Weibull distribution (Azharuddin et al., 2015; Lee et al., 2008).

Therefore, the decision rule for deciding whether or not a node failure has occurred after s periods of no message receipts can be expressed as:

$$\frac{P(F_{l+s}|s)}{P(\bar{F}_{l+s}|s)} \underset{\bar{F}_{l+s}}{\overset{F_{l+s}}{\gtrless}} \frac{(c_{10}-c_{00})}{(c_{01}-c_{11})}, \tag{12}$$

where $P(F_{l+s}|s)$ is the probability that a node has failed given that, after the last message is received at time $n = l$, no message is received for s time steps starting from time $n = l+1$ to time $n = l+s$, and $P(\bar{F}_{l+s}|s)$ is the probability that the node has not failed under the aforementioned conditional, i.e.,

$$P(F_{l+s}|s) = P(F_{l+s}|\bar{R}_{l+1}, ..., \bar{R}_{l+s}, R_l) \tag{13}$$

$$P(\bar{F}_{l+s}|s) = P(\bar{F}_{l+s}|\bar{R}_{l+1}, ..., \bar{R}_{l+s}, R_l) \tag{14}$$

It then remains to model the conditional probabilities in (13) and (14), taking into account the node failure probability and the rate of transmission suppression in the transmission suppression scheme being employed. This permits the computation of the optimal value of s, denoted as s^*.

3.3. Failure probability

From Bayes rule,

$$P(\bar{F}_{l+s}|\bar{R}_{l+1}, ..., \bar{R}_{l+s}, R_l) = \alpha_{l+s} P(\bar{R}_{l+1}, ..., \bar{R}_{l+s}|\bar{F}_{l+s}, R_l) P(\bar{F}_{l+s}|R_l), \tag{15}$$

and

$$P(F_{l+s}|\bar{R}_{l+1}, ..., \bar{R}_{l+s}, R_l) = \alpha_{l+s} P(\bar{R}_{l+1}, ..., \bar{R}_{l+s}|F_{l+s}, R_l) P(F_{l+s}|R_l), \tag{16}$$

where α_{l+s} is a normalising factor given by:

$$\alpha_{l+s} = \frac{1}{P(\bar{R}_{l+1}, ..., \bar{R}_{l+s}|R_l)}. \tag{17}$$

For brevity sake, we shall denote the sequence of no message receipts $\{\bar{R}_{l+1}, ..., \bar{R}_{l+s}\}$ simply as $\bar{R}_{l+1:l+s}$. Moreover, since the receipt of a message at time $n = l$ implies that the node has not failed at time $n = l$, R_l is equivalent to $\bar{F}_l$. Thus, (15) and (16) can be expressed succinctly as:

$$P(\bar{F}_{l+s}|\bar{R}_{l+1:l+s}, \bar{F}_l) = \alpha_{l+s} P(\bar{R}_{l+1:l+s}|\bar{F}_{l+s}, \bar{F}_l) P(\bar{F}_{l+s}|\bar{F}_l), \tag{18}$$

and

$$P(F_{l+s}|\bar{R}_{l+1:l+s}, \bar{F}_l) = \alpha_{l+s} P(R_{l+1:l+s}|F_{l+s}, \bar{F}_l) P(F_{l+s}|\bar{F}_l), \tag{19}$$

First, we consider the factor $P(\bar{F}_{l+s}|\bar{F}_l)$ in (18), which is the probability that the node does not fail at time $n = l + s$, given that it has not failed by time $n = l$. Within each sensing cycle, there is an associated probability of node non-failure (or the staying alive probability) β_n given by:

$$\beta_n = P(\bar{F}_n|\bar{F}_{n-1}), \tag{20}$$

Therefore,

$$\beta_{l+1} = P(\bar{F}_{l+1}|\bar{F}_l) \tag{21}$$

Furthermore, for all $i > 1$,

$$\begin{aligned} &P(\bar{F}_{l+i}|\bar{F}_l) = \\ &P(\bar{F}_{l+i}|\bar{F}_l, \bar{F}_{l+i-1}) P(\bar{F}_{l+i-1}|\bar{F}_l) + P(\bar{F}_{l+i}|\bar{F}_l, F_{l+i-1}) P(F_{l+i-1}|\bar{F}_l). \end{aligned} \tag{22}$$

Note that, since nodes never recover once they have failed,

$$P(\bar{F}_n|F_{n-1}) = 0. \tag{23}$$

Thus, by substituting (20) and (23) into (22), we obtain:

$$P(\bar{F}_{l+i}|\bar{F}_l) = \beta_{l+i} P(\bar{F}_{l+i-1}|\bar{F}_l). \tag{24}$$

If we denote $P(\bar{F}_{l+i}|\bar{F}_l)$ by b_{l+i}, then we may express (24) as:

$$b_{l+i} = \beta_{l+i} b_{l+i-1}. \tag{25}$$

For $i = 1$, i.e., at time $n = l+1$ when there is the first instance of no message receipt at the sink, $b_l = P(\bar{F}_l|\bar{F}_l) = 1$, and $b_{l+1} = \beta_{l+1}$. For $i = 2$, $b_{l+2} =$

$\beta_{l+2}b_{l+1} = \beta_{l+1}\beta_{l+2}$. Moreover, for $i = 3$, $b_{l+3} = \beta_{l+3}b_{l+2} = \beta_{l+1}\beta_{l+2}\beta_{l+3}$. It follows that for $i = s$,

$$b_{l+s} = \prod_{i=1}^{s} \beta_{l+i}, \tag{26}$$

i.e.,

$$P(\bar{F}_{l+s}|\bar{F}_l) = \prod_{i=1}^{s} \beta_{l+i} \quad \text{and} \quad P(F_{l+s}|\bar{F}_l) = 1 - \prod_{i=1}^{s} \beta_{l+i}. \tag{27}$$

Next, we consider the posterior probability $P(\bar{R}_{l+1:l+s}|\bar{F}_{l+s}, \bar{F}_l)$ in (18); this is the probability that no message is received for the s time steps from $n = l + 1$ to $n = l + s$, given that the node has not failed by time $n = l + s$ as well as by time $n = l$. The fact that the node has not failed by $n = l + s$ implies that the non-receipt of messages for the s time steps could only be due to the transmission suppression. Since the suppression of a message at one time step does not depend on the suppression at any other time step, but on the evolution of the state of the phenomenon being monitored, it follows that:

$$P(\bar{R}_{l+1:l+s}|\bar{F}_{l+s}, \bar{F}_l) = \prod_{i=l+1}^{l+s} P(\bar{R}_i|\bar{F}_{l+s}, \bar{F}_l) = \gamma^s, \tag{28}$$

where γ is the constant rate of transmission suppression.

Similarly, the posterior probability $P(\bar{R}_{l+1:l+s}|F_{l+s}, \bar{F}_l)$ given in (19) is the probability that no message is received for the s time steps from $n = l+1$ to $n = l+s$, given that the node has failed by time $n = l+s$, but has not failed by time $n = l$; since the node could have failed any time between $n = l$ and $n = l + s$, the non-receipt of messages for the s time steps could either be due to node failure or transmission suppression. In this case, the probability $P(\bar{R}_{l+1:l+s}|F_{l+s}, \bar{F}_l)$ is given by:

$$P(\bar{R}_{l+1:l+s}|F_{l+s}, \bar{F}_l) = \frac{1}{1 - \prod_{k=1}^{s} \beta_{l+k}} \sum_{i=1}^{s} \gamma^{i-1}(1 - \beta_{l+i}) \prod_{j=1}^{i-1} \beta_{l+j} \tag{29}$$

The proof of this result is given in Appendix A.

Now, by substituting (27), (28) and (29) into (18) and (19), we obtain:

$$P(\bar{F}_{l+s}|\bar{R}_{l+1}, ..., \bar{R}_{l+s}, R_l) = \alpha_{l+s}\gamma^s \prod_{i=1}^{s} \beta_{l+i}, \tag{30}$$

and

$$P(F_{l+s}|\bar{R}_{l+1},...,\bar{R}_{l+s},R_l) =$$
$$\alpha_{l+s}\left[\frac{1}{1-\prod_{k=1}^{s}\beta_{l+k}}\sum_{i=1}^{s}\gamma^{i-1}(1-\beta_{l+i})\prod_{j=1}^{i-1}\beta_{l+j}\right]\left(1-\prod_{i=1}^{s}\beta_{l+i}\right). \quad (31)$$

The relation in (31) simplifies to:

$$P(F_{l+s}|\bar{R}_{l+1},...,\bar{R}_{l+s},R_l) = \alpha_{l+s}\sum_{i=1}^{s}\gamma^{i-1}(1-\beta_{l+i})\prod_{j=1}^{i-1}\beta_{l+j}, \quad (32)$$

Finally, by substituting (30) and (32) into the decision rule of (12), we obtain the following:

$$\sum_{i=1}^{s}\gamma^{i-1}(1-\beta_{l+i})\prod_{j=1}^{i-1}\beta_{l+j} \underset{\bar{F}_{l+s}}{\overset{F_{l+s}}{\gtrless}} \frac{(c_{10}-c_{00})}{(c_{01}-c_{11})}\gamma^{s}\prod_{i=1}^{s}\beta_{l+i}, \quad (33)$$

which represents the optimal rule for deciding whether or not a heartbeat message should be transmitted from the node, for any $s \geq 1$, in order to minimise the Bayes risk.

3.4. Optimal heartbeat transmission protocol

Except for specific values of γ, β_n and the operational costs, (33) generally has no closed-form solution. Moreover, unless the staying-alive probability β_n is constant, the optimal heartbeat period s^* varies for different l in the same run of the WSN application, where l is the last instance of a message receipt at the sink. Thus, the condition given in (33) has to be checked after the last message is received at $n = l$ for every value of s starting from $s = 1$ until the least value of s, denoted s_min, for which it can be decided that a failure has occurred. The value s_{min} is s^*. In order not to burden the node with the evaluation of the decision rule, s^* can be determined at the sink, so that the sink sends this value to the node as part of its acknowledgement after a message is received; this procedure is described in Algorithm 2.

If the sink receives the next heartbeat message at the time $n = l + s^*$, then it knows the node is operating fine. Else if no heartbeat message is received at that time, the sink assumes that a node failure has occurred.

Algorithm 2 Optimal heartbeat transmission

Initialise the optimal heartbeat period $s = s^*$ by evaluating the decision rule of (33) starting from $s = 1$ to $s = s^*$, for $l = 0$.

At each sensing cycle:

Node functions:

Obtain vector of sensor readings $\mathbf{x}_{node}$.
Filter the vector of sensor readings.
Update or predict sink's estimate of the vector of sensor readings.
if New state is significantly different from the sink estimate or heartbeat timer has expired **then**
    Transmit new state.
    On acknowledgement, update local copy of sink state and optimal heartbeat period s^*.
    Restart heartbeat timer.
else
    Do not transmit. Sink assumes that new state vector is not significantly different from its own estimate.
    Decrement the heartbeat (countdown) timer.
end if

Sink functions:

Update or predict sink state.
if Message is received **then**
    Store received data in a permanent datastore.
    Evaluate the decision rule of (33) starting from $s = 1$ to the least value of s, s_{min}, for which (33) indicates a failure.
    Update the optimal heartbeat period s^* as s_{min}.
    Send acknowledgement together with optimal heartbeat period s^*.
    Restart heartbeat timer.
else
    Store the predicted sink state in a permanent datastore.
    Decrement heartbeat timer.
    if Heartbeat timer has expired **then**
        Decide a failure has occurred.
    end if
end if

3.5. Short notes on transmission suppression rate, operational costs, and staying-alive probability

3.5.1. Transmission suppression rate

The transmission suppression rate γ depends on the transmission suppression protocol being employed as well as the phenomenon being monitored. An estimate of γ can practically be obtained by running the transmission suppression protocol in a WSN without any heartbeats for a short period of time. Then, the relative frequency of messages suppressed over time within the accepted error tolerance of the application is observed. A cumulative moving average of these relative frequencies can be used as γ. It is worth noting that, for use in the heartbeat transmission protocol, the estimate of γ obtained as described may be poor, for the very reason that it is obtained for only a short period of the WSN lifetime. Moreover, this estimation procedure for γ does not incorporate the transmission of heartbeats, which alters the suppression rate.

3.5.2. Operational costs

Once heartbeats are incorporated in the WSN functional mode, the operational costs have slightly different meanings, and are hence referred to as "heartbeat operational costs":

1. c_{00} is the cost of not sending any heartbeat when no node failure has occurred; this represents a correct decision as the node/sink correctly estimates that a failure has not occurred.
2. c_{11} is the cost of the sink deciding that a heartbeat must have been sent at time $n = l + s$ on the non-receipt of a message, when a node failure has actually occurred. This also represents a correct decision.
3. c_{10} is the cost of the node sending a heartbeat at $n = l + s$, which is acknowledged at the sink, implying no node failure has occurred; this is a false alarm cost associated with the cost of communicating the heartbeat message.
4. c_{01} is the cost incurred when the node does not send a heartbeat at $n = l + s$, when in fact a node failure has occurred; this is a missed detection cost. In many applications, it is possible to express c_{01} in terms of $c_{01,n}$, which is the cost of missing data in one sampling interval. Since the node may have failed any time after $n = l$ and up to $n = l+s$, the cost $c_{01,n}$ can be incurred up to s times. Thus, the expected cost

c_{01} in terms of $c_{01,n}$ can be given as:

$$c_{01} = \frac{\sum_{i=1}^{s} P(\bar{F}_{l+i-1}, F_{l+i}|F_{l+s}, \bar{F}_l)(s-i+1)c_{01,n}}{\sum_{i=1}^{s} P(\bar{F}_{l+i-1}, F_{l+i}|F_{l+s}, \bar{F}_l)}. \tag{34}$$

The probability $P(\bar{F}_{l+i-1}, F_{l+i}|F_{l+s}, \bar{F}_l)$ is given in Appendix A as part of the proof of (29); using this result, the expected cost of missed detection c_{01} can be given as:

$$c_{01} = \frac{\sum_{i=1}^{s}(1-\beta_{l+i})\prod_{j=1}^{i-1}\beta_{l+j}(s-i+1)c_{01,n}}{\sum_{i=1}^{s}(1-\beta_{l+i})\prod_{j=1}^{i-1}\beta_{l+j}} \tag{35}$$

3.5.3. Staying-alive probability

In order to obtain the staying alive probability, one would have to specify a node failure distribution. A common failure distribution in the context of wireless sensor networks is the Weibull distribution (Azharuddin et al., 2015; Lee et al., 2008), which is a generalisation of the memoryless exponential distribution.

Remark 1

Given a node whose failure rate is given by the Weibull distribution thus:

$$P(F_n) = 1 - e^{-(nt_{sample}/\lambda)^{\kappa}}, \tag{36}$$

where

1. t_{sample} is the sampling period,
2. $\lambda\Gamma(1+1/\kappa) = \tau$ is the expected lifetime of the node,
3. κ is the Weibull slope ($\kappa < 1$ yields a decreasing failure rate, $\kappa = 1$ gives the exponential failure distribution which has a constant failure rate, and $\kappa > 1$ yields an increasing failure rate),

the staying-alive probability β_n can be derived as follows:

$$\beta_n = P(\bar{F}_n|\bar{F}_{n-1}) = \frac{P(\bar{F}_n, \bar{F}_{n-1})}{P(\bar{F}_{n-1})} = \frac{P(\bar{F}_n)}{P(\bar{F}_{n-1})}. \tag{37}$$

Note that between time steps $n-1$ and n, the sampling period t_{sample} elapses, and thus,

$$P(F_n) = 1 - e^{-((n-1)t_{sample}/\lambda)^{\kappa}}, \tag{38}$$

Therefore, the staying alive probability β_n can be expressed as:

$$\beta_n = \frac{e^{-(nt_{sample}/\lambda)^{\kappa}}}{e^{-((n-1)t_{sample}/\lambda)^{\kappa}}} = e^{(t_{sample}/\lambda)^{\kappa}[(n-1)^{\kappa} - n^{\kappa}]} \tag{39}$$

4. Experimental validation

In this section, we experimentally validate the proposed optimal heartbeat transmission protocol in Algorithm 2 in terms of its performance in (a) reducing the energy cost of heartbeat transmission and, (b) reducing the length of time dead nodes go unnoticed at the sink. These performance measures are concisely expressed in terms of the total heartbeat operational cost given as:

$$\mathcal{C} = c_{10} \times \text{Number of false alarms} + c_{01,n} \times \text{Number of missed detections.} \tag{40}$$

where c_{10} (the cost of false alarm) is the energy cost of every heartbeat transmission, and $c_{01,n}$ is the cost of missing important data in one sampling interval as dead nodes go unnoticed. By (40), we have implicitly assigned values of zero to the costs of making correct decisions c_{00} and c_{11}, i.e., deciding that a node failure has occurred when, in fact, it has, or deciding that no node failure has occurred when indeed it has not.

For our experiments, we have employed experimental data obtained via Sense-and-Send as part of the Cogent-House project (Wilkins, 2015) which involved a deployment of 235 sensor nodes in 38 homes to monitor air temperature and relative humidity. We consider three different datasets, each of which is collected over a one-year period at a sampling period t_{sample} of 5 minutes. Our first dataset represents the humidity in a living room, and is denoted as "LivingHum"; the second dataset represents the temperature in a kitchen, and is denoted as "KitchenTemp"; the third dataset represents the temperature in a bedroom, and is denoted as "BedTemp". The choice of these datasets instead of the commonly used Intel Lab Data is due to the larger quantity of available data, the nature of the deployments within a real life (non-laboratory) application, and the availability of datasets with 100% yield allowing for an accurate baseline (Wilkins, 2015).

Our experimental methodology is outlined below:

1. First, we simulate a system where the sequence data given by LivingHum is transmitted from a single sensor node (not the entire network) using the L-SIP transmission protocol (Goldsmith & Brusey, 2010). The L-SIP implementation uses a predefined error of $\epsilon = 2\%$, which is the required error tolerance for the humidity monitoring application. Furthermore, the L-SIP implementation uses an exponential weighted moving average (EWMA) filter with a smoothing factor

$\alpha = 0.2$ for data smoothing and linear extrapolation for predicting the sink state. L-SIP on LivingHum results in a transmission suppression rate of $\gamma = 0.9705$, with a reconstruction error of 0.93% RMSE. The original and reconstructed signal at the sink are shown in Figure 1; a zoomed-in version showing the performance for the first week of data is given in Figure 2

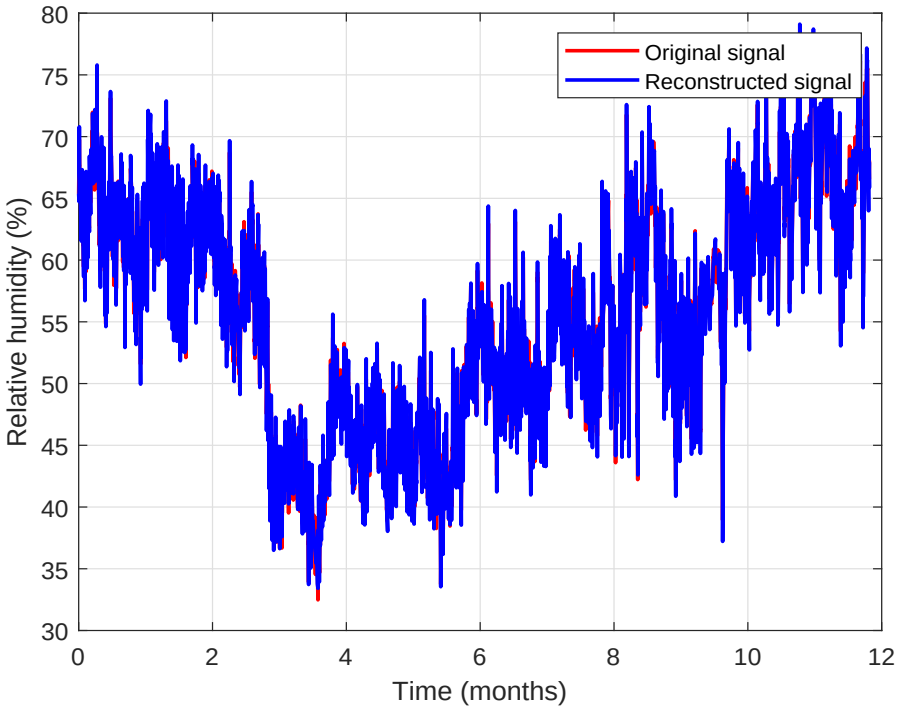

Figure 1: L-SIP transmission suppression on LivingHum dataset. The L-SIP implementation uses linear extrapolation for prediction and an EWMA filter with smoothing factor $\alpha = 0.2$, with an error tolerance $\epsilon = 2\%$.

2. Secondly, we simulate a system where the sequence data given by KitchenTemp is transmitted from a single sensor node using an event detection protocol. The event detection implementation uses upper and lower thresholds of 25 and 16 degrees Celsius respectively, such that the node transmits data only when the sensed temperature falls below 16 degrees or exceeds 25 degrees Celsius. In this implementation, no smoothing filter is applied, and we have used a constant (naive)

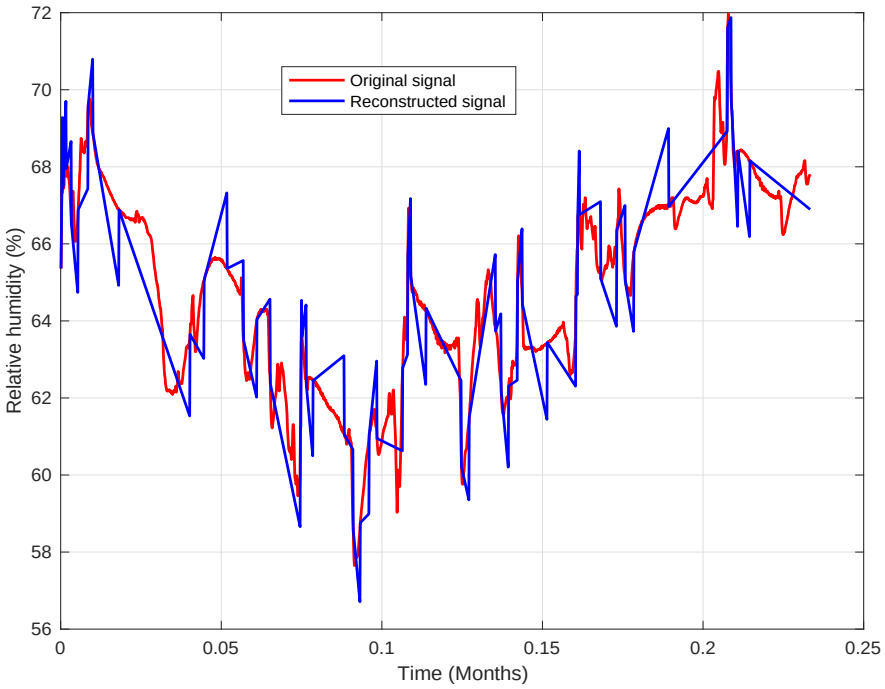

Figure 2: L-SIP transmission suppression on LivingHum dataset, zoomed in for the first week. The L-SIP implementation uses linear extrapolation for prediction and an EWMA filter with smoothing factor $\alpha = 0.2$, with an error tolerance $\epsilon = 2\%$.

(Aderohunmu et al., 2013) model for the prediction of the sink state. This event detection protocol results in a transmission suppression rate of $\gamma = 0.9969$. The original and reconstructed signals are given in Figure 3.

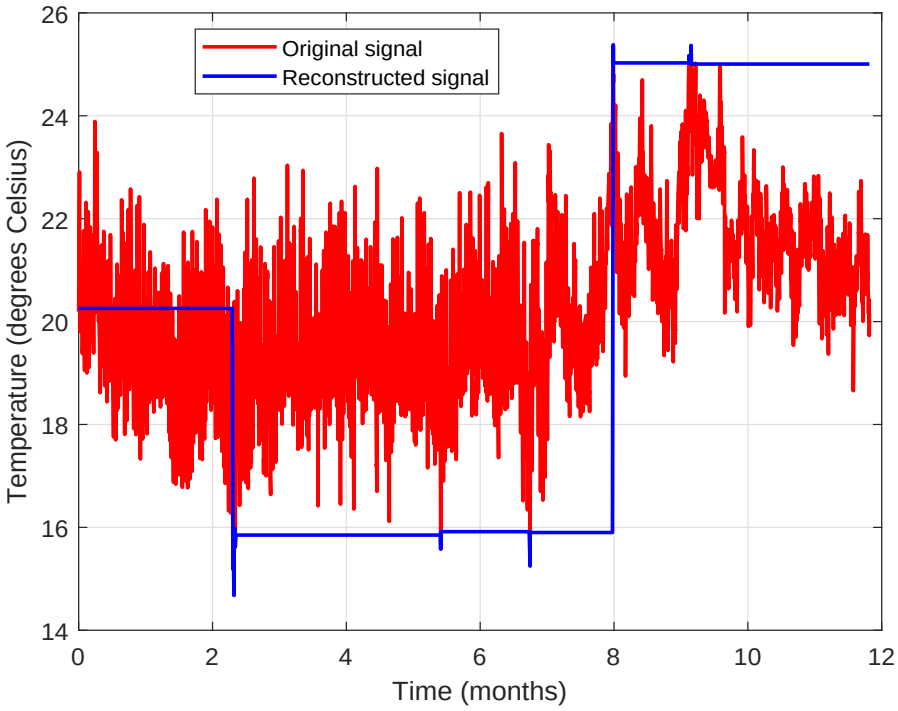

Figure 3: Event detection transmission protocol on KitchenTemp dataset. This event detection uses no smoothing filter and uses a naive model for prediction, with lower and upper thresholds of 16 and 25 degrees Celsius respectively.

3. Thirdly, we simulate a system where the sequence data given by BedTemp is transmitted from a single sensor node using the L-SIP transmission protocol. The L-SIP implementation uses a predefined error of $\epsilon = 0.5$ degrees Celsius, which is the required error tolerance for the temperature monitoring application. Again, the L-SIP implementation uses an exponential weighted moving average filter with a smoothing factor $\alpha = 0.2$ for data smoothing and linear extrapolation for predicting the sink state. L-SIP on BedTemp results in a transmission suppression rate of $\gamma = 0.9730$, with a reconstruction RMSE of 0.2348 degree Cel-

sius. The original and reconstructed signal at the sink are shown in Figure 4.

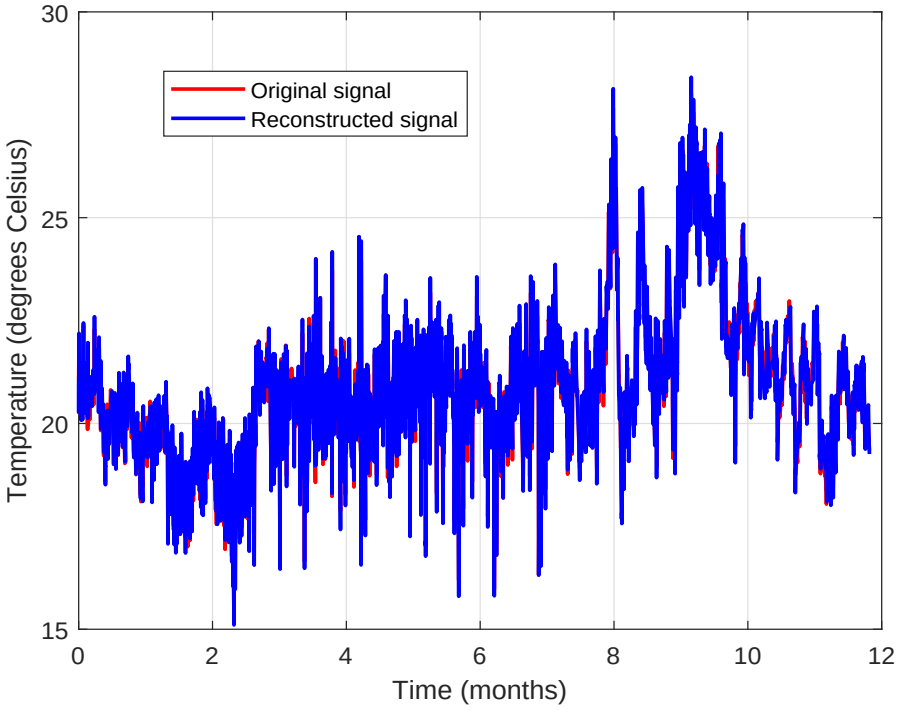

Figure 4: L-SIP transmission suppression on BedTemp dataset. The L-SIP implementation uses linear extrapolation for prediction and an EWMA filter with smoothing factor $\alpha = 0.2$, with an error tolerance $\epsilon = 0.5$ degrees Celsius.

4. Next, we overlay a Weibull failure distribution on the original data LivingHum, KitchenTemp and BedTemp, so that the node fails at some random time according to the Weibull failure rate, and thus no message is transmitted after this time. We run 1000 random trials.
5. We then employ the optimal heartbeat transmission protocol as given by Algorithm 2 on LivingHum, KitchenTemp and BedTemp (which are overlaid by the Weibull failure process). On LivingHum and BedTemp, we run L-SIP predictive data reduction, while we perform event detection on KitchenTemp, using the following baseline parameters:
   (a) Suppression rate of:
      i. $\gamma = 0.9705$ for L-SIP transmission suppression on LivingHum;

ii. $\gamma = 0.9969$ for event detection protocol on KitchenTemp;
iii. $\gamma = 0.9730$ for L-SIP transmission suppression on BedTemp.

Note that the above suppression rates are as given in steps 1-3 of this experimental methodology.

(b) sampling period of $t_{sample} = 5$ minutes —this is the sampling period of the Cogent-House monitoring application.

(c) Expected node lifetime of $\tau = 6$ months. While the actual lifetime of the nodes were about 2 years, we have empirically chosen 6 months for our simulation. This is because choosing a much higher lifetime reduces the Weibull probability of failure, so that it is impossible to reasonably simulate the occurrence of node failure on the one-year worth of data.

(d) Ratio of 1 between the cost of missed detection in one sampling period and the cost of false alarm, where the cost of false alarm c_{10} is kept at 1. This way, the energy cost of heartbeat transmissions and the cost of missing data due to dead nodes are equally important.

(e) Weibull shape parameter of $\kappa = 3$, in order to have an increasing failure rate (Azharuddin et al., 2015).

It is worth emphasising that if the node failure rate is not constant (i.e., if $\kappa > 1$ for the Weibull distribution), the proposed method does not yield a constant heartbeat period throughout the whole run of the WSN application. Instead, the heartbeat period is dynamically updated based on the instance of the last message receipt at the sink. This behaviour is illustrated in Figure 5, Figure 6 and Figure 7.

6. For comparison, we simulate the following heartbeat transmission protocols:

(a) A constant heartbeat transmission whose period is given as s_{init} and is obtained thus: suppose that, rather than the principled procedure given in Algorithm 2, one obtains a constant heartbeat period s_{init} empirically in order to minimise the Bayes risk, using WSN information available at the start of deployment. Then s_{init} can be set to the initial heartbeat period in the proposed algorithm. Note that in Figure 5 and Figure 7, $s_{init} = 522$, while in Figure 6, $s_{init} = 2372$.

(b) A constant heartbeat transmission whose period is given as s_{steady} and is obtained thus: suppose that, rather than the principled procedure given in Algorithm 2, one obtains a constant heartbeat

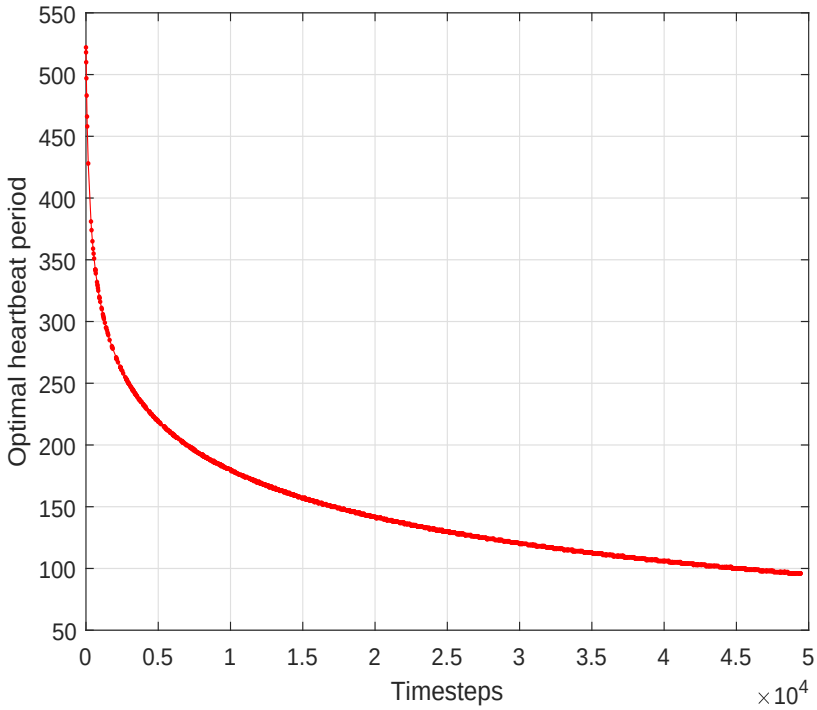

Figure 5: Update of optimal heartbeat period: L-SIP transmission suppression on LivingHum dataset, with the following parameter settings: $t_{sample} = 5$ minutes, $\gamma = 0.9705$, $\tau = 6$ months, $\kappa = 3$, $c_{10} = 1, c_{01,n} = 1$. Initial and final heartbeat periods are 522 and 96 respectively.

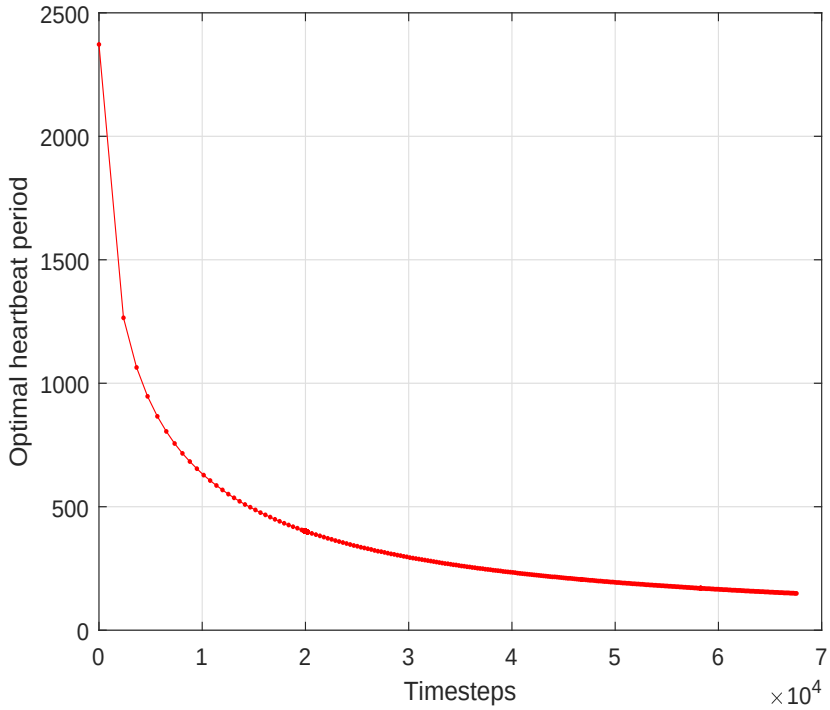

Figure 6: Update of optimal heartbeat period: event detection transmission on Kitchen-Temp dataset, with the following parameter settings: $t_{sample} = 5$ minutes, $\gamma = 0.9969$, $\tau = 6$ months, $\kappa = 3$, $c_{10} = 1, c_{01,n} = 1$. Initial and final heartbeat periods are 2372 and 149 respectively.

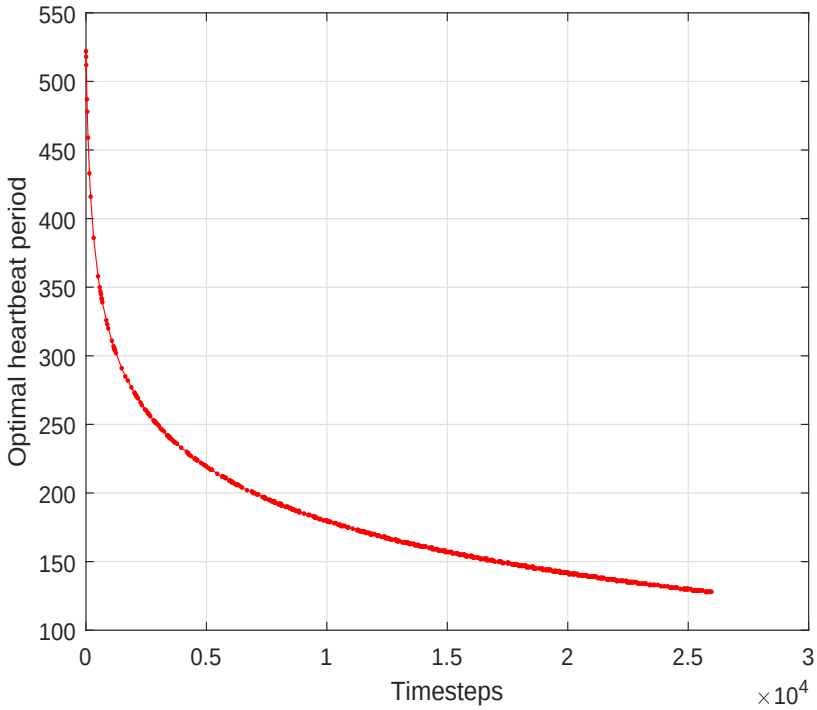

Figure 7: Update of optimal heartbeat period: L-SIP transmission suppression on BedTemp dataset, with the following parameter settings: $t_{sample} = 5$ minutes, $\gamma = 0.9730$, $\tau = 6$ months, $\kappa = 3$, $c_{10} = 1, c_{01,n} = 1$. Initial and final heartbeat periods are 522 and 128 respectively.

period s_{steady} empirically in order to minimise the Bayes risk using knowledge of the steady-state characteristics of the WSN. Then s_{steady} can be set to the steady-state heartbeat period of the proposed algorithm. Note that in Figure 5, $s_{steady} = 96$; in Figure 6, $s_{steady} = 149$; and in Figure 7, $s_{steady} = 128$.

(c) A constant heartbeat transmission whose period is given as $s_{avg} = 0.5(s_{init} + s_{steady})$, which is the midpoint between s_{steady} and s_{init}.

(d) Variance Bound (Rost & Balakrishnan, 2006) with a required false positive rate $FP_{req} = 0.5$, since the energy cost of heartbeat transmissions and the cost of missing data due to dead nodes are equally important.

(e) Affirmative Adaptive Failure Detection (AAFD) (Noor et al., 2012).

The results of the above experiments are given in Figure 8 to Figure 13.

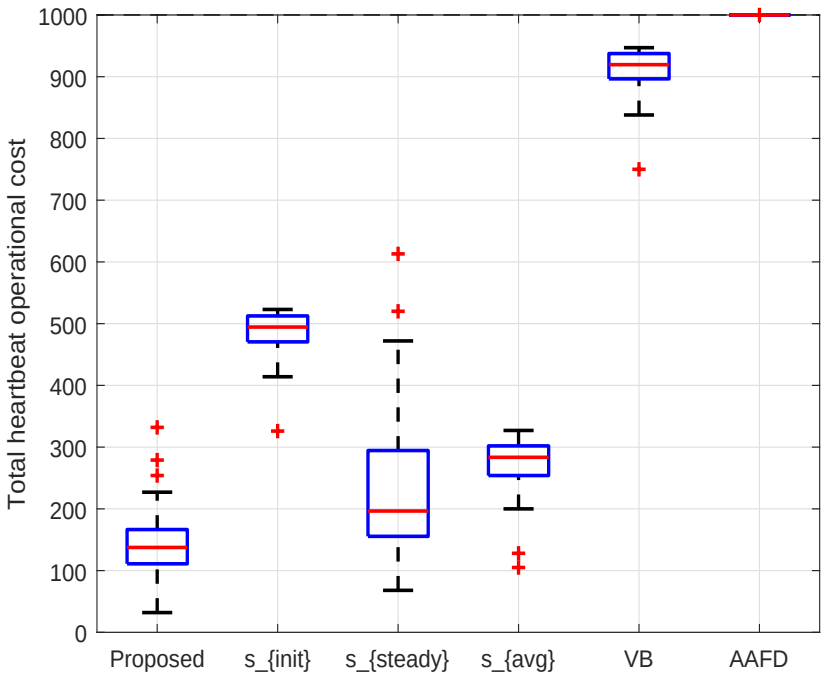

Figure 8: Total heartbeat operational cost: L-SIP transmission suppression on LivingHum dataset. $t_{sample} = 5$ minutes, $\gamma = 0.9705$, $\tau = 6$ months, $\kappa = 3$, $c_{10} = 1, c_{01,n} = 1$.

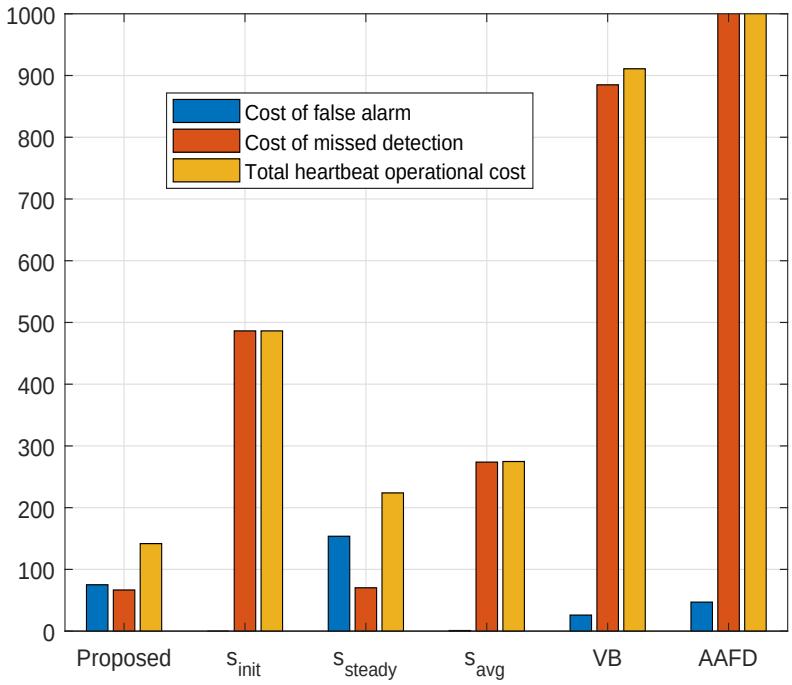

Figure 9: Average cost of missing dead nodes and energy cost of heartbeat transmission: L-SIP transmission suppression on KitchenTemp dataset. $t_{sample} = 5$ minutes, $\gamma = 0.9705$, $\tau = 6$ months, $\kappa = 3$, $c_{10} = 1, c_{01,n} = 1$.

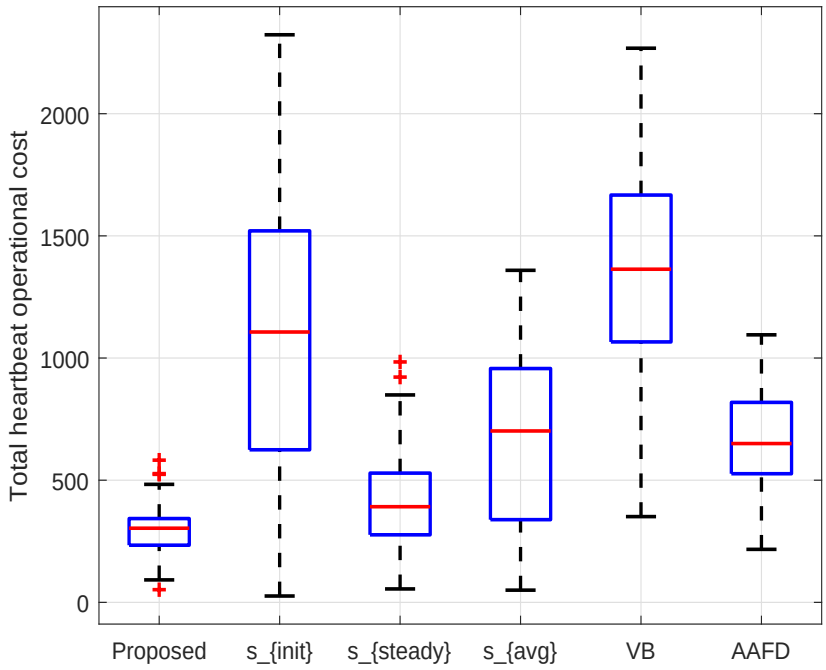

Figure 10: Total heartbeat operational cost: event detection on KitchenTemp dataset. $t_{sample} = 5$ minutes, $\gamma = 0.9969$, $\tau = 6$ months, $\kappa = 3$, $c_{10} = 1, c_{01,n} = 1$.

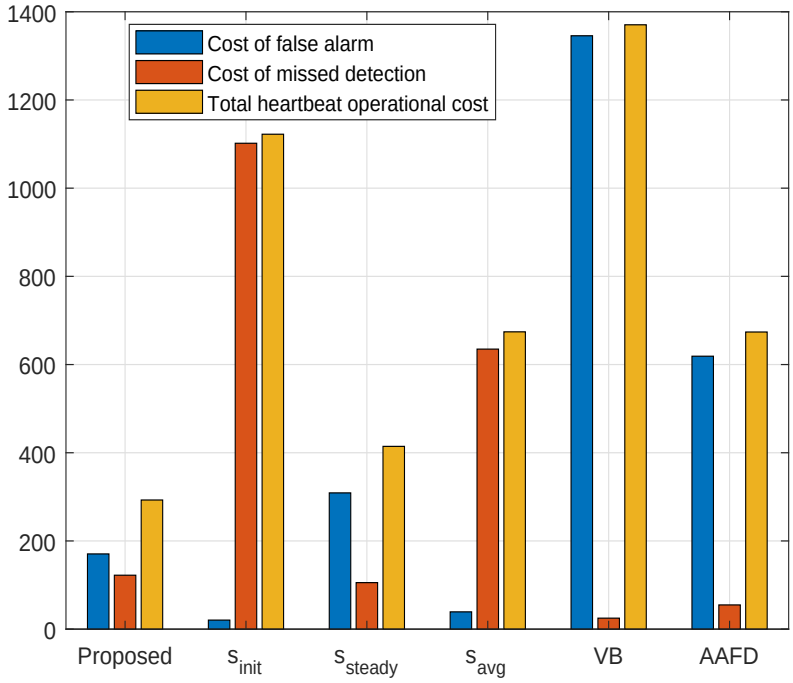

Figure 11: Average cost of missing dead nodes and energy cost of heartbeat transmission: event detection on KitchenTemp dataset. $t_{sample} = 5$ minutes, $\gamma = 0.9969$, $\tau = 6$ months, $\kappa = 3$, $c_{10} = 1, c_{01,n} = 1$.

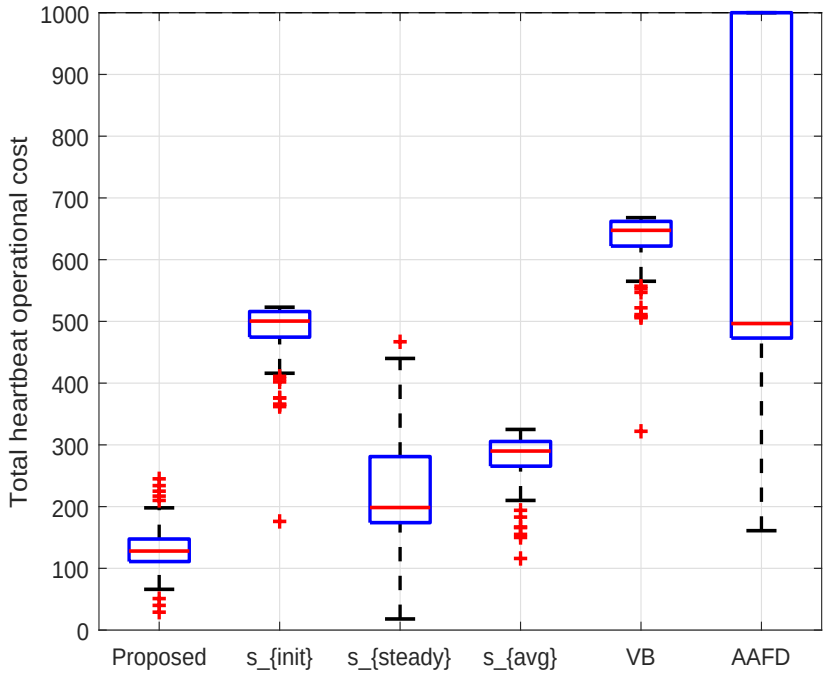

Figure 12: Total heartbeat operational cost: L-SIP transmission suppression on BedTemp dataset. $t_{sample} = 5$ minutes, $\gamma = 0.9730$, $\tau = 6$ months, $\kappa = 3$, $c_{10} = 1, c_{01,n} = 1$.

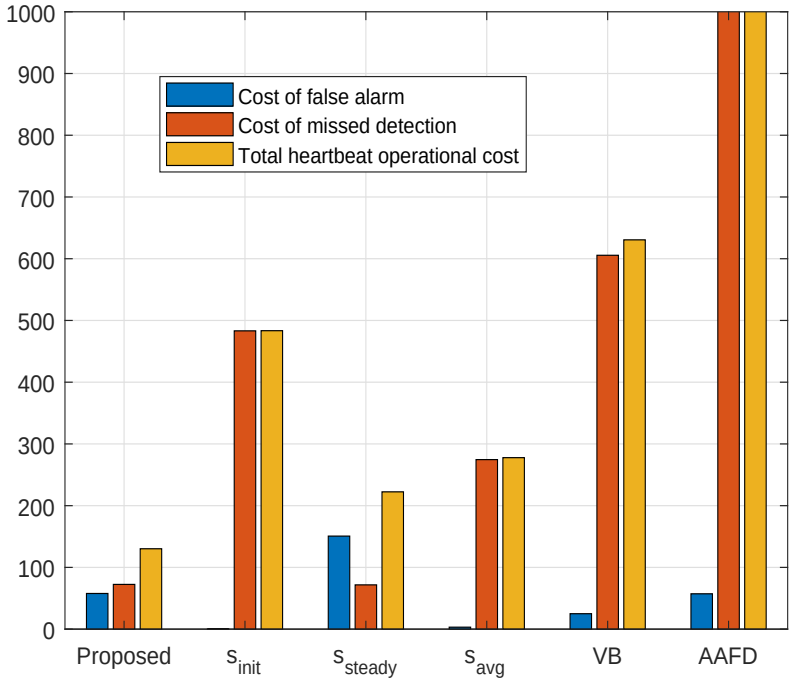

Figure 13: Average cost of missing dead nodes and energy cost of heartbeat transmission: L-SIP transmission suppression on BedTemp dataset. $t_{sample} = 5$ minutes, $\gamma = 0.9730$, $\tau = 6$ months, $\kappa = 3$, $c_{10} = 1, c_{01,n} = 1$.

7. To evaluate the robustness of our approach under different WSN settings, we also simulate the L-SIP transmission suppression protocol on the dataset LivingHum for different values of:
   (a) Sampling time: $0.1, 1, 2, 5, 60, 120, 360, 720, 1440$ minutes.
   (b) Expected node lifetime: $1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13$ months.
   (c) Weibull shape parameter: $0.5, 1, 1.5, 2, 2.5, 3, 3.2, 3.25$.
   (d) The ratio of the cost of missed detection in one sampling period $c_{01,n}$ to the cost of false alarm c_{10}: $0.1, 0.2, 0.5, 1, 2, 5$, where the cost of false alarm c_{10} is kept at 1. The required false positive rate FP_{req} for VB is computed from the ratios.
   (e) Transmission suppression rate:
   $0.16, 0.23, 0.47, 0.64, 0.77, 0.89, 0.94, 0.97, 0.99$.
   Note that these rates are obtained by running the transmission suppression protocol at different error tolerance levels, without any heartbeat transmissions, and for only a small amount of the WSN lifetime. For this reason, the estimate of γ is imperfect and has an effect on the performance of the proposed heartbeat transmission protocol. Thus, we also simulate our proposed algorithm by incorporating perfect knowledge of the suppression rate, i.e., where data transmissions are suppressed with a probability exactly given by the estimate of γ; we refer to this modification of the proposed algorithm as "Proposed*".

   The above values are chosen empirically based on the baseline parameters in step 4. The results for the experiments are given in Figure 14, Figure 15, Figure 16, Figure 17 and Figure 18.

5. Discussion of results

The results in Figure 8 show that the proposed algorithm significantly reduces the total heartbeat operational cost as compared to the existing adaptive heartbeat transmission protocols, VB and AAFD. Figure 9 shows that while VB and AAFD achieve lower costs of false alarm, they take a long time to detect failures, thus increasing the cost of missed detection, and consequently the total heartbeat operational cost. Our algorithm, on the other hand, achieves a better trade-off and results in a reduction of at least 84.44% in the average total heartbeat operational cost over VB and AAFD as shown in Figure 9. The performance gain of our proposed algorithm is as a result of the fact that the proposed algorithm is based on knowledge of

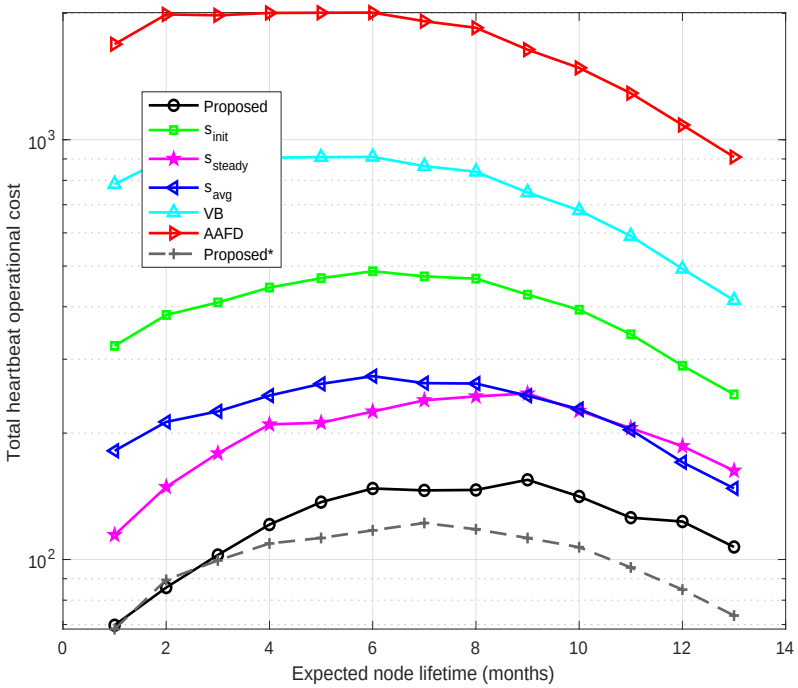

Figure 14: Expected lifetime: L-SIP transmission suppression on LivingHum dataset. $t_{sample} = 5$ minutes, $\gamma = 0.9730$, $\kappa = 3$, $c_{10} = 1, c_{01,n} = 1$.

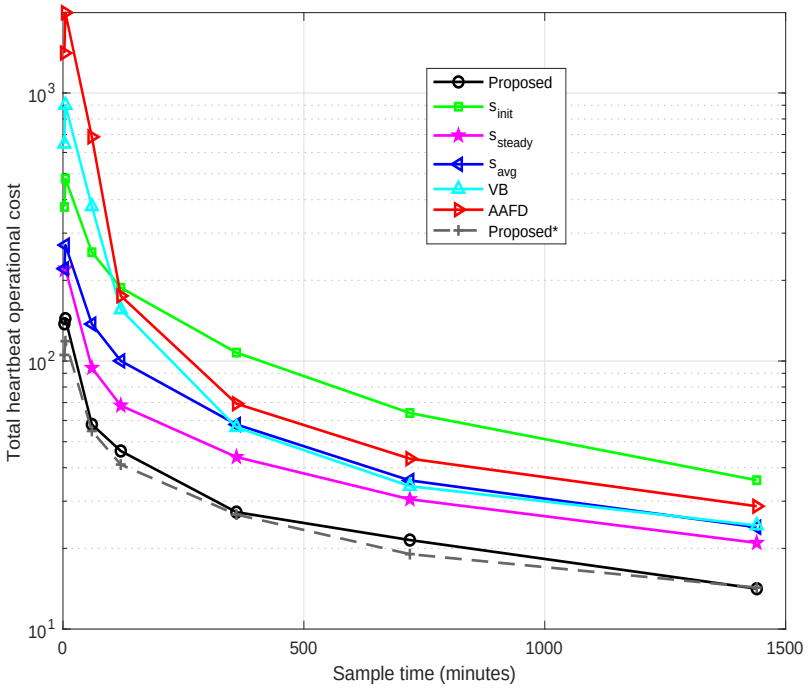

Figure 15: sampling period: L-SIP transmission suppression on LivingHum dataset. $\tau = 6$ months, $\gamma = 0.9730$, $\kappa = 3$, $c_{10} = 1, c_{01,n} = 1$.

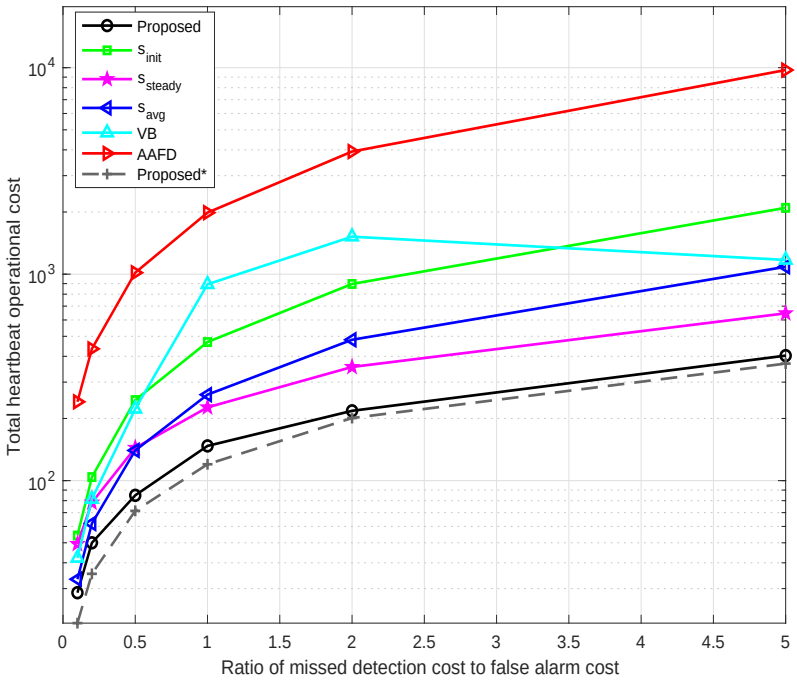

Figure 16: Ratio of missed detection cost to false alarm cost: L-SIP transmission suppression on LivingHum dataset. $t_{sample} = 5$ minutes, $\gamma = 0.9705$, $\tau = 6$ months, $\kappa = 3$, $c_{10} = 1$.

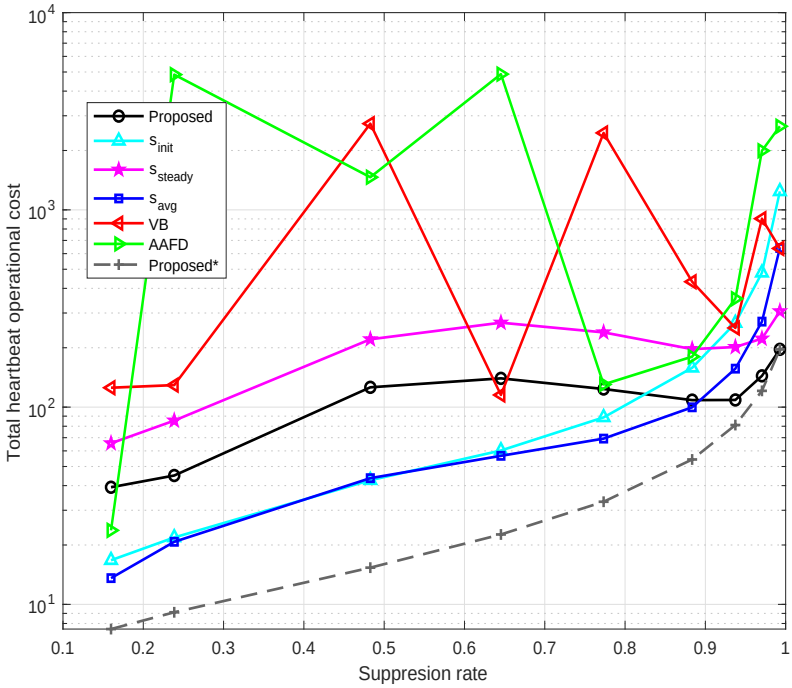

Figure 17: Suppression rate: L-SIP transmission suppression on LivingHum dataset. $t_{sample} = 5$ minutes, $\tau = 6$ months, $\kappa = 3$, $c_{10} = 1, c_{01,n} = 1$.

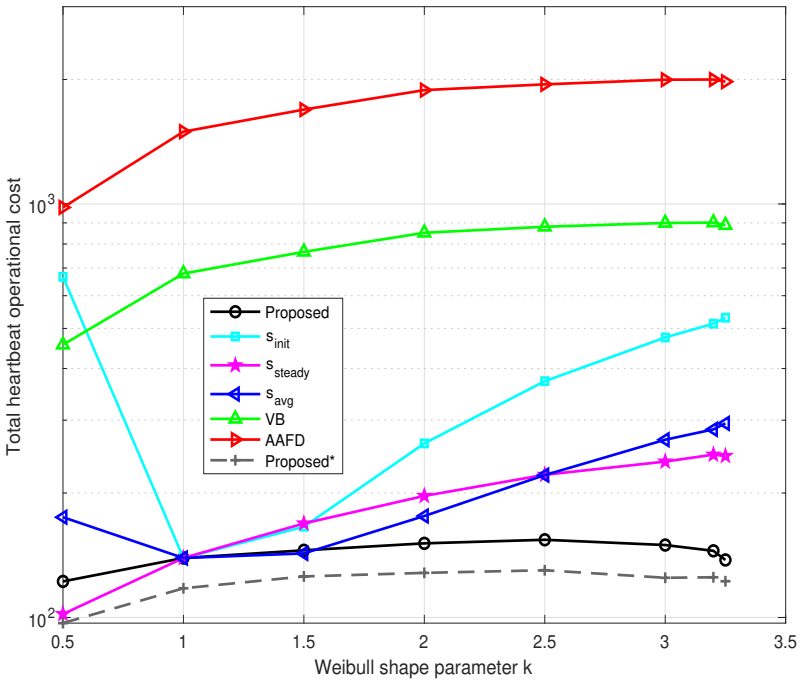

Figure 18: Weibull shape parameter: L-SIP transmission suppression on LivingHum dataset. $t_{sample} = 5$ minutes, $\gamma = 0.9705$, $\lambda = 6.7191$ months, $c_{10} = 1, c_{01,n} = 1$. $\kappa > 3.5$ was not simulated due to numerical instability.

the heartbeat operation costs, the node failure distribution, as well as the transmission suppression scheme employed. The existing adaptive heartbeat transmission methods do not incorporate the aforementioned information, and only estimate the heartbeat period using the inter-arrival times of the heartbeats.

Moreover, the proposed algorithm outperforms the constant heartbeat transmissions with heartbeat periods s_{init}, s_{steady} and s_{avg} as shown in Figure 8. In terms of the average total heartbeat operational cost in Figure 9, our algorithm results in at least 36.68% reduction over all the constant heartbeat transmissions. The cost reduction of our algorithm over the constant heartbeat transmission is due to the fact that, as time increases, the nodes become increasingly likely to fail, and thus, the optimal heartbeat protocol adaptively reduces the heartbeat period to match the increased failure rate.

Among the constant heartbeat transmissions, s_{steady} gives the best performance in terms of the average total heartbeat operational cost. Since the periods s_{init} and s_{avg} are larger than s_{steady}, they result in less false alarm costs but at the expense of higher costs of missed detection. The period s_{steady}, however, yields a better trade-off. This is because s_{steady} is set as the steady-state heartbeat period of the proposed optimal heartbeat protocol which minimises the Bayes risk. However, without the proposed algorithm, s_{steady} will have to be determined empirically, and will therefore require exhaustive trial and error.

In Figure 10 to Figure 13, the proposed algorithm exhibits performance gains over the existing heartbeat transmission approaches similar to that shown in Figure 8.

Figure 14 to Figure 18 illustrate the robustness of the proposed algorithm to different WSN environments. The results indicate that the proposed algorithm performs well not only on specific datasets with specific WSN settings, but also over a wide range of parameter settings.

5.1. Expected node lifetime (Figure 14)

As the expected node lifetime increases, the staying-alive probability β_n increases, and therefore the probability that a node failure will occur decreases. Thus, the total heartbeat operational cost becomes dominated by the cost of false alarms, since it becomes unlikely that a node failure may have been missed. Therefore, increasing the expected lifetime increases the total heartbeat operational cost. However, as the expected lifetime increases further (beyond $\tau = 3$ months in the figure), the staying-alive probability

increases slowly, as it approaches one, and there is not much increase in the cost of false alarms. Consequently, the total heartbeat operational cost tends to reduce, since the cost of missed detection decreases with increase in the expected lifetime.

As shown in Figure 14, the proposed optimal heartbeat period transmission protocol achieves the minimum total heartbeat operational cost under both small and large expected node lifetimes. Similarly, in the next sections, the proposed algorithm is shown to automatically adapt the optimal heartbeat period to variations in the sampling period, heartbeat operational costs, Weibull shape parameter and the suppression rate to minimise the total heartbeat operational cost.

5.2. sampling period (Figure 15)

As sampling period increases, the staying-alive probability β_n decreases, and hence, the probability that a node failure will occur increases. Therefore, the total heartbeat operational cost becomes dominated by the cost of missed detection, rather than the cost of false alarms, since it becomes more and more likely that a failure has occurred at the time a heartbeat is transmitted. Thus, increasing the sampling period increases the total heartbeat operational cost. However, the staying alive probability decreases exponentially with increase in the sampling period. Therefore, as the sampling period increases further (beyond $t_{sample} = 5$ minutes in the figure), the staying-alive probability decreases slowly, as it approaches zero. As a result, there is not much increase in the cost of missed detection beyond this point. Consequently, there is an eventual decrease in the total heartbeat operational cost, as the sampling period increases, since the cost of false alarm decreases.

5.3. Cost of missed detection (Figure 16)

For a given probability of node failure, increasing the cost of missed detection over the cost of false alarms implies that it becomes increasingly more costly to miss any node failures than to send out a false alarm heartbeat. Thus, if the cost of false alarm is kept constant, then increasing the cost of missed detection directly increases the total heartbeat operational cost.

5.4. Weibull shape parameter (Figure 18)

As the Weibull shape parameter increases, the staying-alive probability β_n decreases, and hence, the probability of failure increases. Therefore, the total heartbeat operational cost becomes dominated by the cost of missed

detection, rather than the cost of false alarms, since it becomes more and more likely that a failure has occurred at the time a heartbeat is transmitted. Thus, increasing the Weibull shape parameter increases the total heartbeat operational cost. However, the staying alive probability decreases exponentially with increase in the Weibull shape parameter. Therefore, as κ increases further (beyond $\kappa = 2.5$ in the figure), the staying-alive probability decreases slowly, as it approaches zero. As a result, there is not much increase in the cost of missed detection beyond this point. Consequently, there is an eventual decrease in the total heartbeat operational cost, as Weibull shape parameter increases, since the cost of false alarm decreases.

5.5. Suppression rate (Figure 17)

For a given probability of node failure, increasing the suppression rate implies that fewer messages are transmitted, and there is a longer period of no message receipts. Therefore, it becomes increasingly likely that the non-receipt of messages at the sink is due to transmission suppression instead of node failure. Thus, the total heartbeat operational cost is dominated by the cost of false alarms, rather than the cost of missed detection. Therefore, with increase in the suppression rate, the total heartbeat operational cost increases, as indicated by "Proposed*". However, because of an imperfect estimate of the suppression rate, Proposed is not able to exhibit the characteristics of "Proposed*".

It will be noted that the constant heartbeat transmission methods do not perform well in all WSN settings. For example, the constant heartbeat transmission with period s_{avg} achieves a good performance when $\kappa = 1, 1.5$, but performs poorly in other regions of κ (Figure 18). In the same manner, the existing adaptive heartbeat transmission methods AAFD and VB perform poorly, because they do not adapt the heartbeat periods to variations in the WSN characteristics, but rather on the statistical distribution of heartbeat inter-arrival times. Variance Bound (VB), on the other hand, introduces the required false positive rate (FP_{req}) in the computation of the heartbeat period. Thus, by appropriately selecting FP_{req} depending on the relative cost of missed detection, VB outperforms AAFD in Figure 16.

In contrast, our proposed optimal heartbeat period shows robustness under all scenarios and achieves superior performance in terms of the total heartbeat operational cost for both transmission suppression and event detection. Thus, the proposed approach eliminates the trial and error that may

be required to select an appropriate heartbeat period for any given WSN environment, while minimising the Bayes risk. Since the cost of missing important data because dead nodes go unnoticed and the cost of transmitting false alarm heartbeat messages can often both be given in terms of their monetary values (e.g., in dollars), minimising the Bayes risk yields significant savings in IoT applications.

Even with imperfect knowledge of the suppression rate in Figure 17, the proposed approach "Proposed" outperforms existing heartbeat transmissions in terms of the total heartbeat operational cost in the region of high suppression rates ($\gamma \geq 0.9$). However, for low suppression rates, "Proposed" is outperformed by "Proposed*" (which has perfect knowledge of the suppression rate), as well as the constant heartbeat transmissions with periods s_{init} and s_{avg}. This suggests that a good estimate of the suppression rate is required for applications and transmission suppression protocols with low suppression rates. This is, however, not much of an issue, since most event detection and existing transmission suppression protocols are able to achieve suppression rates well above 90% (Gaura et al., 2011, 2013). In the case of applications with low suppression rates, the constant heartbeat transmission with period s_{init} (which is the initial heartbeat period of the proposed algorithm) is recommended.

6. Conclusions

This paper has shown that the heartbeat period for transmission suppression algorithms as well as event detection protocols can be chosen so as to achieve an optimal trade-off between minimising the energy cost of heartbeat transmission and reducing the incidence of missing out important phenomenological data due to dead nodes going unnoticed. In this regard, our contribution in the paper is two-fold: first, we derive the optimal heartbeat period in terms of the environmental change probability, node failure rate and heartbeat operational costs, by following a Bayes-risk minimisation; secondly, we propose an optimal heartbeat transmission protocol from the result of the Bayes-risk minimisation.

Extensive experimental validation of the proposed algorithm shows that the Bayes risk, and thus, the cost of maintaining and running the WSN, can be significantly reduced (as much as 84.4% in our simulations) by employing the optimal heartbeat transmission protocol, as compared to using existing heartbeat transmission protocols. Our proposed algorithm shows robustness

and applicability to different WSN environments with different sampling frequency, heartbeat operational costs, transmission suppression rate or node failure distribution. It is straightforward to apply this approach in practice, by initialising the optimal heartbeat period at installation time based on available information, and to dynamically calculate the optimal heartbeat at the sink and to distribute this information to the rest of the network.

A limitation of our proposed approach is that it is sub-optimal when a good estimate of the suppression rate is not available for applications with low suppression rates. Thus, our future work aims to investigate methods of obtaining better estimates of the transmission suppression rate in order to improve the performance of our algorithm. Moreover, while this work assumed, for the sake of simplicity, that the non-receipt of a message at the sink is due only to sensor node failure, future work will be dedicated to incorporating other sources of failures such as transmission loss in the optimal design of heartbeat transmission. Finally, we aim to employ reinforcement learning to the Markov decision process of whether or not to transmit a heartbeat at any given time step in order to maximise some notion of expected long-term rewards related to the Bayes risk.

Appendix A.

Proof of (29):

$$P(\bar{R}_{l+1:l+s}|F_{l+s},\bar{F}_l) = \frac{1}{1-\prod_{k=1}^{s}\beta_{l+k}}\sum_{i=1}^{s}\gamma^{i-1}(1-\beta_{l+i})\prod_{j=1}^{i-1}\beta_{l+j}$$

Proof. First, we prove by mathematical induction that:

$$P(\bar{R}_{l+1:l+s}|F_{l+s},\bar{F}_l) = \sum_{i=1}^{s}\gamma^{i-1}P(\bar{F}_{l+i-1},F_{l+i}|F_{l+s},\bar{F}_l). \tag{A.1}$$

1. Base case:
   Suppose $s = 1$, then

   $$P(\bar{R}_{l+1:l+s}|F_{l+s},\bar{F}_l) = P(\bar{R}_{l+1}|F_{l+1},\bar{F}_l) \tag{A.2}$$

   Given that the node has failed by time $n = l+1$, then no message will be received at time $n = l+1$. Thus, from (A.2),

   $$P(\bar{R}_{l+1}|F_{l+1},\bar{F}_l) = 1. \tag{A.3}$$

Also, the right-hand side of (A.1) evaluates to:

$$P(\bar{R}_{l+1}|F_{l+1},\bar{F}_l) = \sum_{i=1}^{1}\gamma^{i-1}P(\bar{F}_{l+i-1},F_{l+i}|F_{l+1},\bar{F}_l) = P(\bar{F}_l,F_{l+1}|F_{l+1},\bar{F}_l) = 1. \quad \text{(A.4)}$$

2. Induction:
Suppose (A.1) holds true for $s = q$ (where $q \geq 1$ is a natural number) as:

$$P(\bar{R}_{l+1:l+q}|F_{l+q},\bar{F}_l) = \sum_{i=1}^{q}\gamma^{i-1}P(\bar{F}_{l+i-1},F_{l+i}|F_{l+q},\bar{F}_l), \quad \text{(A.5)}$$

then for $s = q+1$, we show that:

$$P(\bar{R}_{l+1:l+q+1}|F_{l+q+1},\bar{F}_l) = \sum_{i=1}^{q+1}\gamma^{i-1}P(\bar{F}_{l+i-1},F_{l+i}|F_{l+q+1},\bar{F}_l). \quad \text{(A.6)}$$

Given that the node has failed by time $n = l+q+1$, then it is either that the node has failed by $n = l+q$ or it fails after that. This implies that:

$$\begin{aligned} &P(\bar{R}_{l+1:l+q+1}|F_{l+q+1},\bar{F}_l) = \\ &P(\bar{R}_{l+1:l+q+1}|F_{l+q+1},\bar{F}_l,F_{l+q})P(F_{l+q}|F_{l+q+1},\bar{F}_l) \\ &+ P(\bar{R}_{l+1:l+q+1}|F_{l+q+1},\bar{F}_l,\bar{F}_{l+q})P(\bar{F}_{l+q}|F_{l+q+1},\bar{F}_l). \end{aligned} \quad \text{(A.7)}$$

(a) Suppose the node has failed by time $n = l+q$, then it fails at $n = l+q+1$ also, since nodes never recover once they have failed; thus,

$$P(\bar{R}_{l+1:l+q+1}|F_{l+q+1},\bar{F}_l,F_{l+q}) = P(\bar{R}_{l+1:l+q+1}|\bar{F}_l,F_{l+q}). \quad \text{(A.8)}$$

Moreover, if the node has failed by time $n = l+q$, then no message should be received at time $n = l+q+1$ with a probability of 1. Therefore,

$$P(\bar{R}_{l+1:l+q+1}|\bar{F}_l,F_{l+q}) = P(\bar{R}_{l+1:l+q}|\bar{F}_l,F_{l+q}). \quad \text{(A.9)}$$

(b) Suppose the node has not failed by time $n = l+q$, then no message will be received at every time step from $n = l+1$ to $n = l+q$ each with a probability equal to the transmission suppression rate γ, while no message should be received at time $n = l+q+1$ with a probability equal to 1 since the node fails by time $n = l+q+1$. Thus,

$$P(\bar{R}_{l+1:l+q+1}|F_{l+q+1}, \bar{F}_l, \bar{F}_{l+q}) = \gamma^q \tag{A.10}$$

Substituting (A.8), (A.9) and (A.10) into (A.7), we obtain:

$$\begin{aligned} &P(\bar{R}_{l+1:l+q+1}|F_{l+q+1}, \bar{F}_l) = \\ &P(\bar{R}_{l+1:l+q}|F_{l+q}, \bar{F}_l)P(F_{l+q}|F_{l+q+1}, \bar{F}_l) + \gamma^q P(\bar{F}_{l+q}|F_{l+q+1}, \bar{F}_l). \end{aligned} \tag{A.11}$$

Furthermore, substituting (A.5) into (A.11) yields:

$$\begin{aligned} P(\bar{R}_{l+1:l+q+1}|F_{l+q+1}, \bar{F}_l) &= \gamma^q P(\bar{F}_{l+q}|F_{l+q+1}, \bar{F}_l) \\ &+ \sum_{i=1}^{q} \gamma^{i-1} P(\bar{F}_{l+i-1}, F_{l+i}|F_{l+q}, \bar{F}_l) P(F_{l+q}|F_{l+q+1}, \bar{F}_l). \end{aligned} \tag{A.12}$$

Notice that, for random variables A, B, $p(A|B) = p(A, B|B)$, therefore,

$$P(\bar{F}_{l+q}|F_{l+q+1}, \bar{F}_l) = P(\bar{F}_{l+q}, F_{l+q+1}|F_{l+q+1}, \bar{F}_l). \tag{A.13}$$

Again, given that a node has failed at $n = l+q$, then it fails at $n = l+q+1$, since nodes never recover once they have failed; thus:

$$P(\bar{F}_{l+i-1}, F_{l+i}|F_{l+q}, \bar{F}_l) = P(\bar{F}_{l+i-1}, F_{l+i}|F_{l+q}, F_{l+q+1}, \bar{F}_l) \tag{A.14}$$

Now, using the chain rule of probability, observe that:

$$\begin{aligned} &P(\bar{F}_{l+i-1}, F_{l+i}|F_{l+q}, F_{l+q+1}, \bar{F}_l) P(F_{l+q}|F_{l+q+1}, \bar{F}_l) = \\ &P(\bar{F}_{l+i-1}, F_{l+i}, F_{l+q}|F_{l+q+1}, \bar{F}_l) \end{aligned} \tag{A.15}$$

For all $i \in \{1, .., q\}$,

$$P(\bar{F}_{l+i-1}, F_{l+i}, F_{l+q}|F_{l+q+1}, \bar{F}_l) = P(\bar{F}_{l+i-1}, F_{l+i}|F_{l+q+1}, \bar{F}_l) \tag{A.16}$$

Substituting (A.13), (A.14), (A.15) and (A.16) into (A.12), we obtain:

$$\begin{aligned} P(\bar{R}_{l+1:l+q+1}|F_{l+q+1}, \bar{F}_l) &= \gamma^q P(\bar{F}_{l+q}, F_{l+q+1}|F_{l+q+1}, \bar{F}_l) \\ &+ \sum_{i=1}^{q} \gamma^{i-1} P(\bar{F}_{l+i-1}, F_{l+i}|F_{l+q+1}, \bar{F}_l), \end{aligned} \tag{A.17}$$

which can be simplified as:

$$P(\bar{R}_{l+1:l+q+1}|F_{l+q+1}, \bar{F}_l) = \sum_{i=1}^{q+1} \gamma^{i-1} P(\bar{F}_{l+i-1}, F_{l+i}|F_{l+q+1}, \bar{F}_l), \quad \text{(A.18)}$$

which is exactly as given by (A.6).

3. Hence, (A.1) is true for all $s \geq 1$.

The relation in (A.1) can now be expanded as:

$$P(\bar{R}_{l+1:l+s}|F_{l+s}, \bar{F}_l) = \sum_{i=1}^{s} \gamma^{i-1} P(\bar{F}_{l+i-1}|F_{l+i}, F_{l+s}, \bar{F}_l) P(F_{l+i}|F_{l+s}, \bar{F}_l) \quad \text{(A.19)}$$

Note that:

$$P(F_{l+i}|\bar{F}_l) = P(F_{l+i}|\bar{F}_l, F_{l+s}) P(F_{l+s}|\bar{F}_l) + P(F_{l+i}|\bar{F}_l, \bar{F}_{l+s}) P(\bar{F}_{l+s}|\bar{F}_l) \quad \text{(A.20)}$$

Given that a node has not failed at time $n = l + s$, then the probability that it has failed at time $n = l + i$ is zero, for all $i \in \{1, ..., s\}$. Thus,

$$P(F_{l+i}|\bar{F}_l, F_{l+s}) = \frac{P(F_{l+i}|\bar{F}_l)}{P(F_{l+s}|\bar{F}_l)} \quad \text{(A.21)}$$

Furthermore,

$$\begin{aligned} P(\bar{F}_{l+i-1}|F_{l+i}, \bar{F}_l) &= P(\bar{F}_{l+i-1}|F_{l+i}, \bar{F}_l, F_{l+s}) P(F_{l+s}|F_{l+i}, \bar{F}_l) + \\ & P(\bar{F}_{l+i-1}|F_{l+i}, \bar{F}_l, \bar{F}_{l+s}) P(\bar{F}_{l+s}|F_{l+i}, \bar{F}_l) \end{aligned} \quad \text{(A.22)}$$

Again, given that a node has failed at time $n = l + i$, then the probability that it has not failed at time $n = l + s$ is zero, for all $i \in \{1, ..., s\}$, since nodes never recover once they have failed. Thus,

$$P(\bar{F}_{l+i-1}|F_{l+i}, \bar{F}_l, F_{l+s}) = P(\bar{F}_{l+i-1}|F_{l+i}, \bar{F}_l) \quad \text{(A.23)}$$

Moreover, observe that:

$$P(\bar{F}_{l+i-1}|\bar{F}_l) = P(\bar{F}_{l+i-1}|\bar{F}_l, F_{l+i}) P(F_{l+i}|\bar{F}_l) + P(\bar{F}_{l+i-1}|\bar{F}_l, \bar{F}_{l+i}) P(\bar{F}_{l+i}|\bar{F}_l). \quad \text{(A.24)}$$

Given that a node has not failed at $n = l + i$, then the probability that it has not failed at time $n = l + i - 1$ is one. Therefore,

$$P(\bar{F}_{l+i-1}|\bar{F}_l, F_{l+i}) = \frac{P(\bar{F}_{l+i-1}|\bar{F}_l) - P(\bar{F}_{l+i}|\bar{F}_l)}{P(F_{l+i}|\bar{F}_l)}. \tag{A.25}$$

By substituting (A.21), (A.23) and (A.25) into (A.19), it becomes:

$$P(\bar{R}_{l+1:l+s}|F_{l+s}, \bar{F}_l) = \sum_{i=1}^{s} \gamma^{i-1} \frac{P(\bar{F}_{l+i-1}|\bar{F}_l) - P(\bar{F}_{l+i}|\bar{F}_l)}{P(F_{l+i}|\bar{F}_l)} \frac{P(F_{l+i}|\bar{F}_l)}{P(F_{l+s}|\bar{F}_l)} \tag{A.26}$$

which can be simplified to:

$$P(\bar{R}_{l+1:l+s}|F_{l+s}, \bar{F}_l) = \frac{1}{P(F_{l+s}|\bar{F}_l)} \sum_{i=1}^{s} \gamma^{i-1} \big[P(\bar{F}_{l+i-1}|\bar{F}_l) - P(\bar{F}_{l+i}|\bar{F}_l)\big] \tag{A.27}$$

Recall from (27) that:

$$P(\bar{F}_{l+s}|\bar{F}_l) = \prod_{i=1}^{s} \beta_{l+i},$$

Therefore,

$$P(\bar{R}_{l+1:l+s}|F_{l+s}, \bar{F}_l) = \frac{1}{1 - \prod_{k=1}^{s} \beta_{l+k}} \sum_{i=1}^{s} \gamma^{i-1} \Big[\prod_{j=1}^{i-1} \beta_{l+j} - \prod_{j=1}^{i} \beta_{l+j} \Big] \tag{A.28}$$

which gives,

$$P(\bar{R}_{l+1:l+s}|F_{l+s}, \bar{F}_l) = \frac{1}{1 - \prod_{k=1}^{s} \beta_{l+k}} \sum_{i=1}^{s} \gamma^{i-1} (1 - \beta_{l+i}) \prod_{j=1}^{i-1} \beta_{l+j} \tag{A.29}$$

□

References

Abbasi, A. A., Akkaya, K., & Younis, M. (2007). A distributed connectivity restoration algorithm in wireless sensor and actor networks. In *Local computer networks, 2007. LCN 2007. 32nd IEEE conference on* (pp. 496–503). IEEE.

Aderohunmu, F. A., Paci, G., Brunelli, D., Deng, J. D., Benini, L., & Purvis, M. (2013). An application-specific forecasting algorithm for extending wsn lifetime. In *Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on* (pp. 374–381). IEEE.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer networks*, *38*, 393–422.

Azharuddin, M., Kuila, P., & Jana, P. K. (2015). Energy efficient fault tolerant clustering and routing algorithms for wireless sensor networks. *Computers & Electrical Engineering*, *41*, 177–190.

Bahrepour, M., Meratnia, N., Poel, M., Taghikhaki, Z., & Havinga, P. J. (2010). Distributed event detection in wireless sensor networks for disaster management. In *Intelligent Networking and Collaborative Systems (INCOS), 2010 2nd International Conference on* (pp. 507–512). IEEE.

Bakhtiar, Q. A., Makki, K., & Pissinou, N. (2012). Data reduction in low powered wireless sensor networks. In *Wireless Sensor Networks-Technology and Applications*. InTech.

Barr, K. C., & Asanović, K. (2006). Energy-aware lossless data compression. *ACM Transactions on Computer Systems (TOCS)*, *24*, 250–291.

Bellavista, P., Cardone, G., Corradi, A., & Foschini, L. (2013). Convergence of manet and wsn in iot urban scenarios. *IEEE Sensors Journal*, *13*, 3558–3567.

Celesti, A., Carnevale, L., Galletta, A., Fazio, M., & Villari, M. (2017). A watchdog service making container-based micro-services reliable in IoT clouds. In *Proc. - 2017 IEEE 5th Int. Conf. Futur. Internet Things Cloud, FiCloud 2017* (pp. 372–378). volume 2017-Janua. URL: `https://ieeexplore.ieee.org/document/8114506/`. doi:`10.1109/FiCloud.2017.57`.

Chu, D., Deshpande, A., Hellerstein, J. M., & Hong, W. (2006). Approximate data collection in sensor networks using probabilistic models. In *null* (p. 48). IEEE.

Cohn, D. L., & Melsa, J. L. (1980). *Decision and estimation theory*. McGraw-Hill.

Demirbas, M., Arora, A., & Mittal, V. (2004). Floc: A fast local clustering service for wireless sensor networks. In *Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS/DSN 2004)* (pp. 1–6).

Fasolo, E., Rossi, M., Widmer, J., & Zorzi, M. (2007). In-network aggregation techniques for wireless sensor networks: a survey. *IEEE Wireless Communications*, *14*.

Foster, I. (1995). The Globus project: a status report. *Futur. Gener. Comput. Syst.*, *15*, 607–621. URL: https://ieeexplore.ieee.org/abstract/document/666541/http://linkinghub.elsevier.com/retrieve/pii/S0167739X99000138. doi:10.1016/S0167-739X(99)00013-8.

Gaura, E. I., Brusey, J., Allen, M., Wilkins, R., Goldsmith, D., & Rednic, R. (2013). Edge mining the internet of things. *IEEE Sensors Journal*, *13*, 3816–3825.

Gaura, E. I., Brusey, J., & Wilkins, R. (2011). Bare necessities—knowledge-driven wsn design. In *Proc. IEEE Sensors* (pp. 66–70).

Goldsmith, D., & Brusey, J. (2010). The spanish inquisition protocol—model based transmission reduction for wireless sensor networks. In *Sensors, 2010 IEEE* (pp. 2043–2048). IEEE.

Gupta, G., & Younis, M. (2003). Fault-tolerant clustering of wireless sensor networks. In *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE* (pp. 1579–1584). IEEE volume 3.

Jin, J., Gubbi, J., Marusic, S., & Palaniswami, M. (2014). An information framework for creating a smart city through internet of things. *IEEE Internet of Things journal*, *1*, 112–121.

Johnsson, B. A., Nordahl, M., & Magnusson, B. (2017). Evaluating a Dynamic Keep-Alive Messaging Strategy for Mobile Pervasive Systems. In *Procedia Comput. Sci.* (pp. 319–326). volume 109. URL: https://www.sciencedirect.com/science/article/pii/S187705091731027X. doi:10.1016/j.procs.2017.05.358.

Lau, B. C., Ma, E. W., & Chow, T. W. (2014). Probabilistic fault detector for wireless sensor network. *Expert Systems with Applications*, *41*, 3703–3711.

Lee, J.-J., Krishnamachari, B., & Kuo, C.-C. J. (2008). Aging analysis in large-scale wireless sensor networks. *Ad Hoc Networks*, *6*, 1117–1133.

Mangelkar, S., Dhage, S. N., & Nimkar, A. V. (2018). A comparative study on RPL attacks and security solutions. In *Proc. 2017 Int. Conf. Intell. Comput. Control. I2C2 2017* (pp. 1–6). volume 2018-Janua. URL: https://ieeexplore.ieee.org/abstract/document/8321851/. doi:10.1109/I2C2.2017.8321851.

Meng, Z., Wu, Z., Muvianto, C., & Gray, J. (2017). A Data-Oriented M2M Messaging Mechanism for Industrial IoT Applications. *IEEE Internet Things J.*, *4*, 236–246. URL: https://ieeexplore.ieee.org/iel7/6488907/6702522/07801816.pdf. doi:10.1109/JIOT.2016.2646375.

Noor, A. S. M., Deris, M. M., Herawan, T., & nor Hassan, M. (2012). On affirmative adaptive failure detection. In *Algorithms Archit. Parallel Process. ICA3PP 2012. Lect. Notes Comput. Sci.*. URL: https://link.springer.com/chapter/10.1007/978-3-642-33065-0{_}13.

Polastre, J., Szewczyk, R., & Culler, D. (2005). Telos: enabling ultra-low power wireless research. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on* (pp. 364–369). IEEE.

Pongle, P., & Chavan, G. (2015). A survey: Attacks on RPL and 6LoWPAN in IoT. In *2015 Int. Conf. Pervasive Comput. Adv. Commun. Technol. Appl. Soc. ICPC 2015*. URL: https://ieeexplore.ieee.org/abstract/document/7087034/. doi:10.1109/PERVASIVE.2015.7087034.

Raza, U., Camerra, A., Murphy, A. L., Palpanas, T., & Picco, G. P. (2012). What does model-driven data acquisition really achieve in wireless sensor networks? In *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on* (pp. 85–94). IEEE.

Rosset, V., Paulo, M. A., Cespedes, J. G., & Nascimento, M. C. (2017). Enhancing the reliability on data delivery and energy efficiency by combining swarm intelligence and community detection in large-scale wsns. *Expert Systems with Applications*, *78*, 89–102.

Rost, S., & Balakrishnan, H. (2006). Memento: A health monitoring system for wireless sensor networks. In *Secon* (pp. 575–584).

Santini, S., & Romer, K. (2006). An adaptive strategy for quality-based data reduction in wireless sensor networks. In *Proceedings of the 3rd international conference on networked sensing systems (INSS 2006)* (pp. 29–36). TRF Chicago, IL.

Sarigiannidis, P., Karapistoli, E., & Economides, A. A. (2015). Detecting sybil attacks in wireless sensor networks using uwb ranging-based information. *Expert Systems with Applications*, *42*, 7560–7572.

Schoellhammer, T. (2010). Failure is inevitable: The trade-off between missing data and maintenance. In *Wireless Sensor Networks* (pp. 159–192). Springer.

Silberstein, A., Puggioni, G., Gelfand, A., Munagala, K., & Yang, J. (2007). Suppression and failures in sensor networks: A bayesian approach. In *Proceedings of the 33rd international conference on Very large data bases* (pp. 842–853). VLDB Endowment.

Su, P. H., Shih, C., Hsu, J. Y., Lin, K., & Wang, Y. (2014). Decentralized Fault Tolerance Mechanism for Intelligent IoT / M2M Middleware. In *2014 IEEE World Forum Internet Things* (pp. 45–50). URL: `https://www.computer.org/csdl/proceedings/wf-iot/2014/3459/00/06803115-abs.html`.

Tulone, D., & Madden, S. (2006). Paq: Time series forecasting for approximate query answering in sensor networks. In *European Workshop on Wireless Sensor Networks* (pp. 21–37). Springer.

Wilkins, R. (2015). *Approaches to transmission reduction protocols in low-frequency wireless sensor networks deployed in the field*. Ph.D. thesis.

Yu, L., Wang, N., & Meng, X. (2005). Real-time forest fire detection with wireless sensor networks. In *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on* (pp. 1214–1217). IEEE volume 2.

Yu, M., Mokhtar, H., & Merabti, M. (2007). Fault management in wireless sensor networks. *IEEE Wireless Communications*, *14*.