Adaptive GPS Duty Cycling and Radio Ranging for Energy-efficient Localization

Raja Jurdak
CSIRO ICT Centre
Pullenvale QLD 4069 Australia
rjurdak@ieee.org

Peter Corke
Built Environment and Engineering
Queensland University of Technology, Brisbane, Australia
peter.corke@qut.edu.au

Dhinesh Dharman
CSIRO ICT Centre
Pullenvale QLD 4069 Australia
dhinesh.dharman@gmail.com

Guillaume Salagnac
INSA
Lyon, France
guillaume.salagnac@insa-lyon.fr

Abstract

This paper addresses the tradeoff between energy consumption and localization performance in a mobile sensor network application. The focus is on augmenting GPS location with more energy-efficient location sensors to bound position estimate uncertainty in order to prolong node lifetime. We use empirical GPS and radio contact data from a large-scale animal tracking deployment to model node mobility, GPS and radio performance. These models are used to explore duty cycling strategies for maintaining position uncertainty within specified bounds. We then explore the benefits of using short-range radio contact logging alongside GPS as an energy-inexpensive means of lowering uncertainty while the GPS is off, and we propose a versatile contact logging strategy that relies on RSSI ranging and GPS lock back-offs for reducing the node energy consumption relative to GPS duty cycling. Results show that our strategy can cut the node energy consumption by half while meeting application-specific positioning criteria.

1 Introduction

This paper addresses the tradeoff between energy consumption and localization performance for outdoor mobile sensor network applications. Mobile network proposals often assume that GPS can provide absolute location information [1], without considering the operational constraints and high energy profile of this technology, which causes rapid depletion of the node batteries.

SenSys'10, November 3–5, 2010, Zurich, Switzerland.

The obvious solution is to duty cycle the GPS module to prolong node lifetime. This comes at the cost of increased position uncertainty. Whenever the GPS is powered down, the uncertainty in the mobile node's position increases with time. Fortunately, most applications can tolerate a certain amount of localization uncertainty. A key challenge is that, even if the node is not moving, its uncertainty will continue to increase until the GPS is again turned on.

This paper proposes the use of more efficient sensor streams to infer node locations and to reduce position uncertainty while the GPS is off. In particular, we focus on radio ranging among neighbouring nodes. Nodes send radio messages to share their position and estimated uncertainty with spatially proximal neighbours, which enables sharing of GPS load among the nodes and an overall reduction in the use of GPS at each node. The challenge in using multiple sensors is to keep the position uncertainty within application-specific bounds while determining which sensor to sample and when to sample it.

To address this challenge, this paper proposes a strategy for managing this position uncertainty/energy tradeoff in mobile sensor networks through the use of multiple sensor streams with different energy cost and accuracy characteristics. Our initial analysis reveals that the simple combination of GPS and contact logging reduces the power consumption of the GPS module but incurs additional power consumption for increased radio activity. We then focus on modeling the evolution of position uncertainty with time and the energy implications of different GPS duty cycling strategies and radio contact logging. Our models use empirical data from long term animal tracking experiments [2] where the GPS is always on, which provides ground truth position data, to build models of node mobility and GPS performance.

This paper considers two classes of positioning applications: (1) applications that can tolerate a fixed uncertainty in node positions, which are representative of location-based mobile phone applications, vehicle tracking applications,

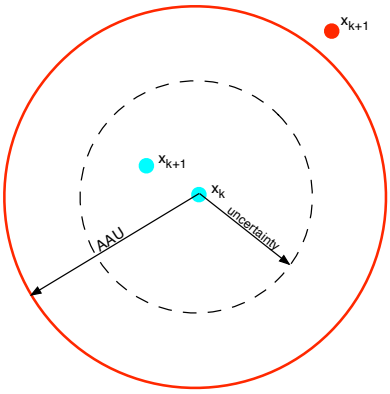


Figure 1. The growing uncertainty (dashed circle) and the maximum uncertainty bound (solid circle).

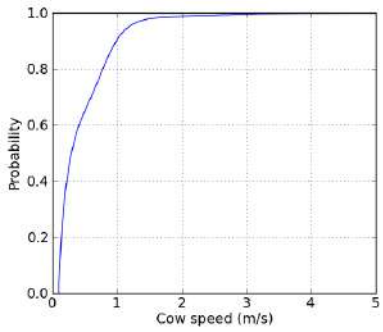


Figure 2. Cumulative distribution of observed animal speed. 95% of measurement are below 1m/s, 100% are below 3.5m/s

and animal tracking applications; and (2) applications where the tolerance for position uncertainty varies based on given landmarks, which are representative of virtual fencing [2] and collaborative gaming applications. While we propose a strategy for class 2 applications that exploits their variable uncertainty to minimise energy consumption, the core focus of the paper remains around the class 1 applications that have wider applicability. We devise an event-based contact logging strategy for class 1 applications that triggers beacon transmission whenever a node detects a significant change in its position state, resulting in significant energy savings.

In sum, the contributions of this paper are three-fold:

- Proposal and evaluation of new GPS duty cycling algorithms that are verified with detailed simulations based on a large empirical dataset. The algorithms use a probabilistic model of the GPS lock time and empirical node mobility models from a cattle tracking experiment, to establish guidelines for how GPS duty cycling affects energy efficiency and position accuracy in mobile applications.
- Novel methods for combining radio ranging with GPS position measurements. We initially explore the performance of periodic radio beaconing with a fixed transmit power versus a less constrained approach with RSSI-based ranging and event-driven contact beaconing. Simulation results confirm that the RSSI- and event- based approaches can reduce energy consumption by nearly half, with a small reduction in a position accuracy.
- Proposal and evaluation of an adaptive GPS duty cycling algorithm for responding to extreme cases where energy or accuracy have to be compromised.

2 Cooperative localization

The notation that we will use in this paper is illustrated in Figure 1. At time k we obtain a GPS measurement x_k and then turn off the GPS. At the next measurement x_{k+1}, the mobile node may be either within or outside the user-set maximum uncertainty bound, indicated by the larger circle. Whenever x_{k+1} falls outside the larger circle, we denote this as an error, since the system would have failed to deliver a position estimate within the user-set uncertainty bound. A primary performance metric is the ratio of errors to the total number of GPS measurements (denoted by the error rate σ), which is a reliability measure of our algorithm. It is important to note that we are not attempting to estimate the exact node position while the GPS is off. Our aim is rather to estimate the bounded vicinity of the node, and to ensure that the next GPS measurement occurs within a defined radius of the previous measurement.

Our approach requires a means to estimate the uncertainty bound (shown as a dashed circle) at any given time. Since we do not know how fast nodes are moving or in what direction, the simplest model is to assume a speed and grow the estimated uncertainty linearly with time. When it approaches the maximum allowable uncertainty bound, taking into account the GPS's lock time, we turn on the GPS. If the node's speed exceeds our assumption, it could end up at the point marked in red when the GPS obtains a fix.

2.1 User Policy

Our approach relies on user policy inputs to identify relevant performance bounds. The operator of the system specifies the user policy, which can include: (1) the acceptable error in position for mobile entities; (2) the desired battery lifetime of the mobile devices; and (3) a preference for error or lifetime. The error comprises two components: (a) the absolute acceptable uncertainty (AAU) which is the maximum uncertainty in Figure 1 within which we expect to be when the GPS next turns on; and (b) the error tolerance (σ^*), which is upper bound on the error rate σ. Together, these error metrics set constraints on the time that the system can use low-end positioning sensors before the uncertainty demands switching on the energy-hungry GPS sensor. The desired battery lifetime sets a target for average power consumption for the mobile device, so that each device can gauge whether it is on track to meeting its target lifetime.

2.2 Node Mobility

We consider three attributes for modelling node mobility: (1) the distribution of past speed data of the mobile nodes, as shown in Figure 2 for cows; (2) the current speed; and (3) the clustering behaviour of the mobile entities.

The distribution of past speed data is useful for estimating the assumed speed of the mobile device, as well as the like-

lihood of variation from that average speed. However, the speed at successive samples is not an independent random variable and is strongly persistent, so we additionally use the current speed for future speed predictions.

Clustering is another important aspect of the node mobility. The clustering behavior builds on the time series of separation distances between every pair of mobile entities. Separation distance information can be obtained using any combination of received signal strength indicator (RSSI) or the difference in exchanged GPS coordinates. The clustering behavior can be described by statistics on the separation distance between the entities and also the duration for which the entities are in contact. These statistics enable determining expected node density and expected contact durations among mobile nodes, which in turn guides the decision on setting GPS duty cycle and radio beaconing metrics.

2.3 Energy Model

Our approach tracks node energy consumption by recording the duty cycles of the major node components [3]: GPS module, radio, MCU, and other components.

GPS duty cycling is a key factor in our approach. We simply compute the GPS module energy consumption at any time t during the deployment as $P_{gps}^{ON} \times DC_{gps} \times t$, where DC_{gps} is the GPS duty cycle, and P_{gps}^{ON} is the power consumption of the GPS module in active mode. The GPS module is completely powered off when not in use for a current draw of zero. We track the radio energy consumption as the sum of the transmission, reception, and listening energy values respectively [3].

The MCU power consumption is heavily dependent on the GPS and beaconing strategies, as the MCU must power on when either the GPS module or radio are active. In general, the MCU duty cycle and energy consumed are expressed as:

$$DC_{MCU} = DC_{gps} + DC_{radio} - overlap \quad (1)$$
$$E_{MCU} = DC_{MCU} \times P_{MCU} \times t$$

where DC_{radio} is the duty cycle of the radio, and $overlap$ denotes any overlap between the on-times for the GPS and radio. Sections 3 and 4 elaborate on MCU energy consumption further for specific duty cycling strategies. The remaining node components, including the regulators and audio board for emitting audio cues to the cattle in the virtual fencing application, have a constant energy.

2.4 Performance Tradeoffs

The user policy in our approach targets both position accuracy and energy efficiency, which are dependent variables that are often in tension: setting a low error tolerance reduces the achievable lifetime of mobile devices for a given battery, because it limits opportunities for GPS duty cycling. In the simplest terms, the system configuration would ideally (1) maximize lifetime; and (2) minimize error rate, within the user set AAU value. Obviously, achieving both goals simultaneously is not possible; however, our approach aims at identifying the operating point that ensures that both metrics are performing within user-specific bounds. Whenever either of these conditions are violated, the system is not delivering on one of the user-set policies. In section 4.3, we propose a best-effort strategy for delivering on energy and accuracy within user-set bounds. However, if a situation arises that fails to meet targets for both metrics, then the system has reached a critical point. The third user policy metric from section 2.1 specifies the user preference for either favouring energy efficiency or position accuracy at such critical points. While in general the preference can be specified as weights for favouring constraints (1) and (2), the rest of this paper uses a boolean value that discretely favours either of the constraints at critical points in the deployment.

Figure 3. A cow with our smart collar

It is important to distinguish between the measured error rate, which is visible to nodes during deployment, and the actual error rate, which can only be tracked when ground truth data is available, such as for offline analysis. In Sections 3 and 4, we track the actual error rate as we conduct offline analysis of empirical data for more insight into system design. Section 4.3 uses measured error rate in developing an online strategy for managing performance.

3 GPS duty cycling

This section applies our framework to GPS duty cycling independently of any other sensor stream. The aim is to explore different GPS duty cycling algorithms and their effect on error rate and energy efficiency.

3.1 Motivating Application

To ground the discussion, we consider an outdoor location monitoring application for tracking cattle using smart collars, as shown in Figure 3 that contain wireless sensor nodes and GPS modules. The goal is to track cow positions, and in some cases, enforce exclusion zones within the paddock, in effect a virtual fence [4, 2]. Virtual fencing is useful for managing cattle in vast grazing lands, such as in Australia, where farms can reach the sizes of small countries, and it is not economically feasible to install physical fences in the whole area. The reader should note, however, that our approach is independent of the virtual fencing aspect and generally applicable to all mobile node tracking scenarios.

The target node lifetime in this application is 3 months, which is the interval at which the animals are brought in for health checks, treatment and sorting. Achieving this lifetime is a challenge because the GPS module on each node has a large current draw [1]. Our cattle monitoring application and

[1] Smart phones that include GPS modules involve similar considerations, albeit with more regular opportunities for recharging

Component	Power (mW)
MCU	18
Radio (TX)	79.2
Radio (RX)	29.7
GPS	165
Switch-mode Regulator	6
Linear Regulator	3.3
Audio	3

Table 1. Active mode power consumption for collar components

cow collar setup are fully described in Wark et al. [2]. We limit the discussion here to a high level description of the collar setup and the energy consumption of each of its components.

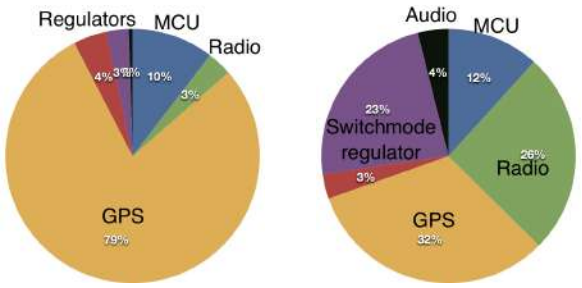
The collars incorporate a Fleck node, a GPS module, and an audio board for generating audio cues to indicate to animals that they are crossing a virtual fenceline. The Fleck node itself comprises an Atmel-1281 MCU and Atmel RF212 radio. Four D-Cell batteries in series provide power to all the collar components via several switch-mode regulators. The weight of these batteries yields a collar weight that approaches the upper limit set by an animal ethics committee in Australia, so larger batteries are not an option.

The power consumption of these components is shown in Table 1. The sleep mode power consumption for the radio, MCU, GPS, and audio is several orders of magnitude lower than the active mode current. The power consumption of the GPS board is the largest by far. To put things in perspective, Figure 4(a) illustrates a scenario with the GPS constantly active. Using our collars with a 10% radio duty cycle, based on a 6ms check interval and 60ms sleep interval, and an always-on GPS board, the GPS accounts for 88% of the power consumption of 209 mW and limits the nodes' lifetime to 19 days. Figure 4(b) sets the GPS duty cycle at 5%, while maintaining the same setup for all of the collar components. In this scenario, the GPS accounts for about a third of the overall node power consumption of 25mW, which extends the nodes' lifetime by a factor of 7.5. This confirms the importance of duty cycling the GPS on the mobile nodes. In both scenarios, the MCU remains on only as required to service to the GPS or radio, which couples its duty cycle to these components' duty cycles.

3.2 GPS lock time

The GPS module does not output a valid position estimate immediately after being turned on. It enters a state where the internal filter is initializing. This is referred to as *acquiring lock* and indicated by a flag in the output data packet. Once it has lock, the GPS reports horizontal dilution of precision (HDOP), which is an estimate of error in the horizontal plane. This value tends to start high, perhaps several tens of metres, and falls over consecutive samples to a few metres. The whole process of obtaining lock and an estimate with acceptable HDOP is referred to as obtaining a GPS *position lock*. We conducted an experiment over three days in which the GPS was turned off for a random amount of time,

batteries.



(a) Always on GPS (b) GPS with duty cycle of 5%

Figure 4. Impact of GPS duty cycling: Reducing GPS duty cycle from 100% to 5% reduces the node power consumption by 7.5 times. The radio has a 10% duty cycle in performing low power listening for both cases.

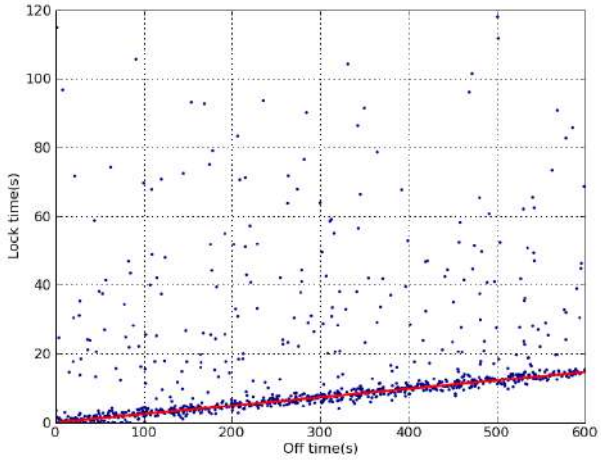


Figure 5. Observed GPS lock time as a function of off time. The vertical scale has been limited to allow the linear characteristic to be visible.

t_O. We trackied the lock time t_L , that time that it takes the GPS chip to achieve HDOP less than 10m. The data is shown as a scatter plot in Figure 5 and we observe a linear cluster of points that is represented by

$$t_L = 0.0245 t_O + 0.0051$$

with a superimposed Poisson random process. In some cases, it took more than 2 minutes to obtain a lock.

3.3 Design Considerations

While GPS provides the most reliable absolute location estimate outdoors, its high power consumption demands duty cycling in order to achieve the target lifetime. Whenever the GPS is active, the localization uncertainty is equal to the GPS error U_{gps}, which depends on the number of satellites currently visible.

Whenever the GPS is powered off at time t_k, and until it next acquires lock, the position uncertainty grows progressively according to

$$U(t) = \bar{s} \times [t - t_k]$$

where $\bar{s}$ is the assumed speed. Since we require $U < AAU$

and the GPS has a finite lock time we need to start the lock process when $t > T_{max}$ where

$$T_{max} = \frac{AAU - U_{gps}(t_k)}{\bar{s}} - t_L \quad (2)$$

is the GPS off time. The value of $\bar{s}$ can be specific to each mobile node, and can be a constant, or vary as a function of time and space. The lock time, t_L, has a weak dependence on the length of time that the GPS has been off, which we can control. It also depends on factors beyond our control, such as the time varying satellite configuration and the deployment environment. For example, some satellites will be obscured in a deep valley or under dense foliage. In practice, a "best estimate" of t_L is used based on data such as shown in Figure 5 or updated in real-time on the node based on the measured time taken to acquire lock. Nevertheless, long lock times will occur with low probability, which means that the estimated uncertainty will occasionally exceed AAU.

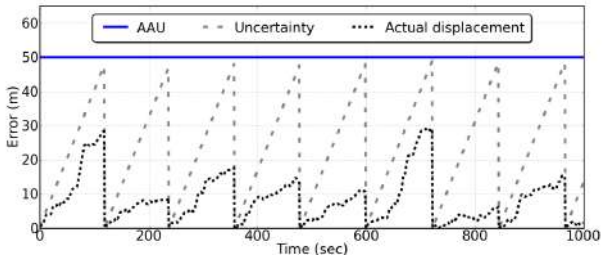
```
loop {t}
  choose_speed()
  if GPS == LOCK then
    position ← GPS position
    U ←U_gps
    lastlocktime = now()
    GPS = OFF
  else
    U ← U + (t * S_c)
    if GPS == OFF and U+(t_L*s̄)> AAU then
      GPS = ON
    end if
  end if
end loop
```

Figure 6. GPS duty cycling strategy

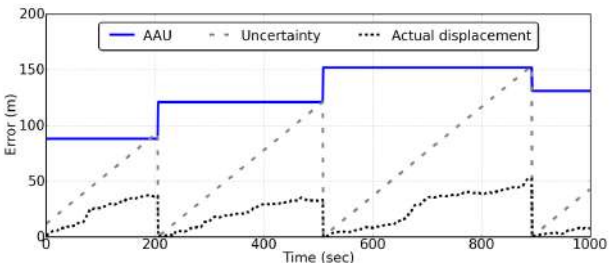
Figure 6 illustrates the algorithm for duty cycling the GPS module. The algorithm uses the choose_speed() function to select the speed model. At each time step the algorithm checks if the GPS has lock and if so it obtains the position, sets the uncertainty to U_{gps}, then powers off the GPS. Otherwise, it grows the uncertainty according to the current speed S_c. Once the uncertainty approaches AAU, it turns on the GPS and the process repeats.

3.4 Bounding the uncertainty

Most positioning applications require a fixed uncertainty bound. For a specific class of applications, such as virtual fencing, the AAU can dynamically vary during the deployment according to application state. For instance, in the virtual fence application, the AAU can vary as a function of the node's position and is the distance of the node from the virtual fence line. If the node is 1000 meters away from the fence, we simply require that the next time the GPS makes a measurement, it will not have crossed the fence line. If we assume the node moves at 1m/s (which is true 95% of the time), then a node can keep its GPS off for more than 985 seconds, with a conservative estimate of 15 seconds for GPS lock time. A node that was 100m away from the fence at the last lock can keep its GPS powered off for 85 seconds.



(a) Error behaviour for a static AAU



(b) Error behaviour for a dynamic AAU

Figure 7. Effect of AAU on error behaviour

Comparatively, a fixed AAU of 50m would result in a GPS off time of only 35 seconds. The sizeable increase in GPS off times relative to a fixed AAU can greatly reduces the average power consumption for dynamic AAU.

Figure 7(a) illustrates the effect of AAU on the growth of the estimated uncertainty. The actual displacement in this plot indicates the ground truth displacement of the mobile node since the last GPS lock. During deployment, the actual displacement is not visible to the nodes, as errors are only discovered when the node acquires the next GPS lock. The purpose in tracking the actual displacement here is to determine how well our uncertainty estimates model the nodes' mobility.

We choose an assumed speed $\bar{s}$ of 0.4 m/sec for the nodes, which is the average node speed in the cow dataset, and the estimated uncertainty grows linearly while the GPS module is powered off. When it approaches AAU, the node activates the GPS and sets its uncertainty to U_{gps} when a fix is obtained, resulting in periodic duty cycling of the GPS. A total of 9 GPS locks occurred in this time window.

In Figure 7(b) the AAU is varied according to the distance between the last observed position of the mobile node and the virtual fence line. While the user sets an initial AAU of 50m, the AAU rises to about 90m at the first GPS lock, reflecting how far the cow is from the virtual fence. The AAU increases to further to 125m and 150m at the next two GPS locks. The higher AAU allows the node to keep its GPS module powered off for longer reducing the number of GPS lock attempts to 4, with the actual displacement never exceeding the uncertainty.

3.4.1 Speed models

The assumed speed is a critical parameter. If it is too low, then the AAU bound will be violated if the animal moves quickly. If it is too high, the GPS off times will be very short which is costly in terms of power. Choosing the right speed

Static	*Dynamic*	*Probabilistic*
$S_c = \overline{s}$	if (s(t)> $\overline{s}$)	i=t-lastlocktime
	S_c=s(t)	if (i==0)
	else	S_c=s(t)
	$S_c = \overline{s}$	if ($S_c > \overline{s}$)
		$P = t_{22}$
		else
		$P = t_{11}$
		else
		$S_c = P \times S_c + \overline{s}(1-P)$
(a)	(b)	(c)

Figure 8. The choose_speed() function for each speed model

requires a speed model, which is highly dependent on the application. The data in Figure 2 indicates an average cow speed of only 0.4 m/sec, and 95% of the time the cows move at 1m/sec or less, with an absolute maximum of 3.5m/sec.

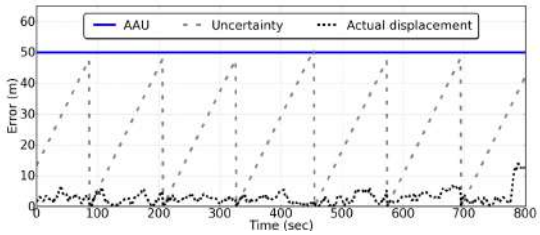
We explore three models for the assumed speed $\overline{s}$, which are summarised in Figure 8:

1. Static: based on a constant assumed speed
2. Dynamic: based on setting the assumed speed as the last observed speed of the mobile node, read from the GPS speed value from the GPS chip at last lock
3. Probabilistic: based on setting the assumed speed on the basis of the last observed speed and a state model of the mobile node

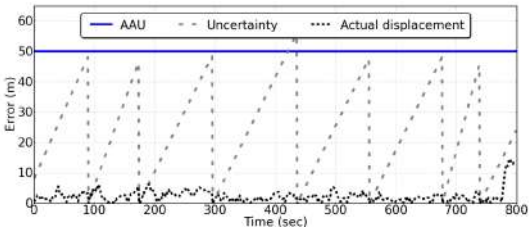
The static and dynamic speed models are application-independent. The static speed model assumes a low variance from an average speed and uses this estimate for the entire deployment. The dynamic speed model assumes a high correlation between the most recent speed measurement and the current speed estimate, i.e. persistence. The probabilistic speed model also relies on persistence, but assigns application-specific probabilities for decaying speed estimates. For instance, previous work has modeled animal mobility as a 3-state Markov process [5]. Unlike some other mobility models in the literature for mobile phone or car applications [6], the animals do not follow specific routes, rather they spend a lot of time in random foraging behaviour.

The probabilistic model in this paper is a simplification of the Markov model of cow speed in [5] where the animal has a slow- and a fast-moving state, states 1 and 2 respectively. State transitions from state i to state j are referred to as t_{ij}. Using data from a 2-day experiment we determine the probabilities that at each time step the animal will transition out of that state. The assumed speed is set to the last observed speed, as for the dynamic model, but at each subsequent time step it moves toward a constant assumed speed using a first-order filter. The time constant is a function of the transition probability of the initial state. The reasoning behind this is that as the time since the last speed observation grows, its significance decreases until at some point, and without any other data we revert to a constant assumed speed.

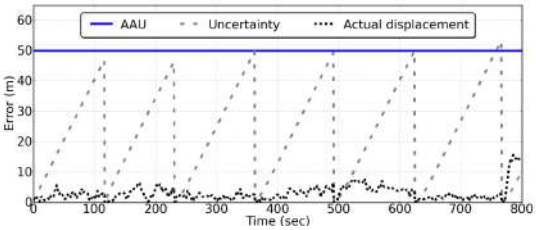
Figure 9 illustrates the effect of each speed model on uncertainty estimation for a constant AAU. All three models use the same $\overline{s}$ of 0.4 m/sec. The uncertainty in the constant



(a) Error behaviour for the static speed model



(b) Error behaviour for the dynamic speed model



(c) Error behaviour for the probabilistic speed model

Figure 9. Effect of speed model on errror behaviour

speed model (Figure 9(a)) grows regularly according to $\overline{s}$, resulting in 6 GPS locks and an error rate of 7.5%. For the dynamic speed model in Figure 9(b), the uncertainty growth rate follows the changes in $\overline{s}$. For instance, at 680 seconds, the node acquires lock and observes that the current speed is higher than $\overline{s}$, and as a result, it increase $\overline{s}$ to better model the actual motion. At the following GPS lock, the same node observes a drop in the measured speed, and reverts to $\overline{s}$ again for its speed estimate. The dynamic model improves the error rate to 3% while increasing the GPS locks to 7. Finally, the probabilistic model results in 7 lock attempts with an error rate of 3.8%.

3.5 Performance Evaluation

To evaluate the impact of our GPS duty cycling strategy on position accuracy and energy efficiency, we have implemented the duty cycling strategy in Figure 6 in a Python-based simulator. We use a dataset which contains GPS data from 30 cow collar nodes collected continuously (once per second) over 2 days (100% duty cycle) from a herd at the Belmont Research Station in central Queensland. This dataset of cow positions represents the ground truth for all the analysis and results in this paper. Each GPS module is represented is represented by a Collar class, with a simple state machine that includes the probabilistic lock time model of Figure 5 and the preceding off time. When the GPS is turned off, the simulator evolves the uncertainty estimate according to the speed model. The simulator keeps

the GPS module off for T_{max} seconds then turns it on to start the locking process. When the simulated GPS obtains lock, the true position is used to determine if the AAU constraint has been violated and to update the error rate σ. The simulator also updates the average power consumption for the GPS, radio, MCU and regulators based on the energy model in Section 2.3, which enables the computation of the node's overall power consumption.

We initially fix the AAU to 50m and explore the impact of changing the assumed speed for each of the three speed models. The results are shown in Figure 10. As expected the GPS power consumption dominates the node power consumption in all case. The GPS power consumption increases with assumed speed, since the faster growth in estimated uncertainty triggers more frequent GPS fixes. Because the GPS and radio timers fire completely independently, the *overlap* parameter for computing the MCU duty cycle is zero. The MCU power consumption increases with increasing GPS activity, as the MCU has to remain in active state for interacting with the GPS module. The radio power consumption, based on a low power listening duty cycle of 5% with 6ms check interval and 128ms sleep interval, is a small contributor to the overall node power consumption, since it only sends short data packets every minute back to the base station.

The GPS power consumption of the probabilistic speed model has the lowest dependence on the assumed speed, as it uses the most recent observed speed estimate for most of the time. For low assumed speeds, the dynamic speed model has higher power consumption than the static speed model, because it occasionally uses a much higher assumed speed when it happens that the speed observed over the previous interval is high. As the assumed speed increases, both the static and dynamic speed models exhibit similar behaviour, as there are fewer instances where the dynamic model switches to current speed values. The dynamic speed model performs best in terms of accuracy, with its error rate σ decreasing steadily with increasing assumed speed, from about 5% at an assumed speed of 0.2m/sec to less than 1% for assumed speeds above 1.2 m/s. The probabilistic and static speed models have similar power consumption at low speeds, with a slight advantage in error rate for the probabilistic model. At higher speeds, the static model achieves lower error rate at the cost of higher power consumption, since it always maintains a conservative assumed speed, while the probabilistic model can often use a lower assumed speed.

Figure 11 shows the effect of enabling dynamic AAU in a virtual fencing application according to distance from the fence. The variance of results here, indicated by error bars, is noticeably higher than for static AAU, because of the disparity among nodes' distance to the fence during the deployment. The dynamic AAU reduces the GPS power consumption by more than half for low assumed speeds, and more than 70% for higher assumed speeds. For the lowest assumed speed of 0.2 m/s, the GPS power consumption is reduced to about the same level as the MCU power consumption and the regulators. The overall node power consumption increases more linearly with assumed speed for dynamic AAU. At the highest assumed speed of 1.4 m/s, the overall node power consumption is about 40% of the static AAU case. The reduced power consumption of dynamic AAU comes at the cost of a higher error rate. This stems from the longer off times for the GPS modules, which increases the expected lock times and the instances where the GPS takes a very long time to obtain a fix. While dynamic AAU can yield significant energy savings at the cost of slightly higher error rates, dynamically changing AAU according to the distance from the fence is a specific feature of the virtual fencing application and similar tracking applications with known boundaries. The remainder of the paper focuses on static AAU, which is more representative of a wider class of GPS-enabled mobile applications.

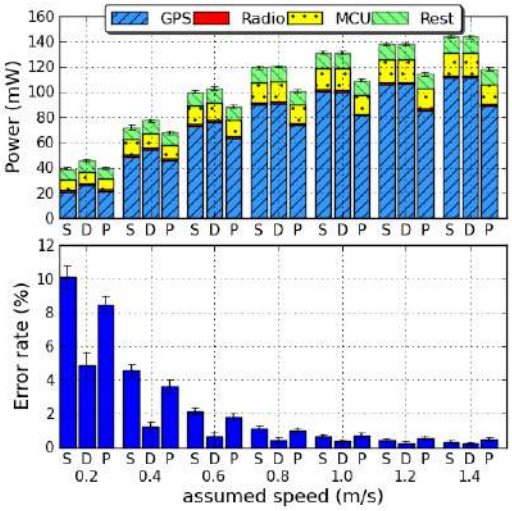


Figure 10. Power consumption and error rate for the 3 speed models (fixed AAU). S: static, D: dynamic, and P: probabilistic.

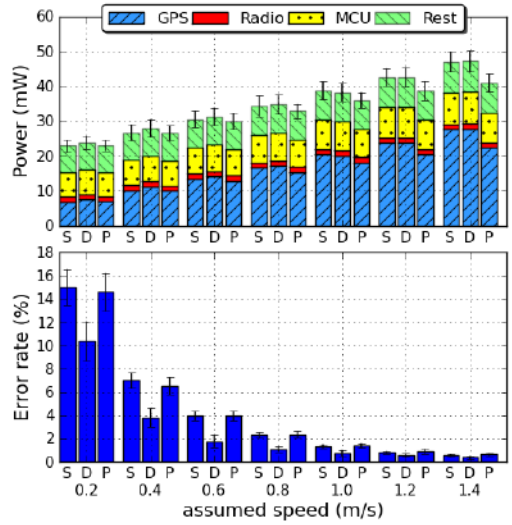


Figure 11. Power consumption and error rate for the 3 speed models (dynamic AAU). S: static, D: dynamic, and P: probabilistic.

In summary, GPS duty cycling clearly reduces the node

energy consumption at the cost of positioning error rates ranging from 0% to 15%. The dynamic speed model provides the best balance between error rate and power consumption, which is why we adopt this model for the remainder of the paper. Although the dynamic AAU yields sizeable benefits for the virtual fencing application, the remainder of this paper adopts the static AAU model that is more representative of general mobile sensor network applications. The next sections investigate how the use of contact logging can further improve the energy/accuracy tradeoff.

4 Coupling GPS and contact logging

Mobile nodes can use non-GPS relative localization signals to reduce position uncertainty while the GPS module is powered off. In many positioning applications, several mobile nodes may cluster together making short range RF contact logging attractive. Consider that node A has powered off its GPS and is growing its estimated uncertainty as a function of time. Receiving a beacon from node B only implies that the two nodes are within a certain distance $\hat{r}$, which can be conservatively estimated as the contact radius R, since the distance between the nodes must be $\hat{r} \leq R$.

The beacon from B also includes its last measured position x_k^B and its current uncertainty estimate U^B. If $U^B + \hat{r} < U^A$ then node B is nearby and has a lower uncertainty than node A. In this case node A lowers its uncertainty estimate

$$U^A := U^B + \hat{r} \tag{3}$$

Lowering the estimated uncertainty enables node A to keep its GPS off for longer, which reduces its energy consumption. If all nodes run this algorithm, the energy-expensive GPS position lock is shared across the nodes. The fairness of the algorithm stems from the fact that when node A relies on node B for its position estimate, $U^A > U^B$. If the two nodes use the same assumed speed to grow their uncertainty, then node A will decide to turn on its GPS before node B, which allows B to rely on A for its position estimate in the next cycle. Using a variable speed model can result in exceptions to this trend, but contact logging will result in long-term sharing of GPS load if the assumed speed follows a random distribution.

If a node's estimated uncertainty approaches AAU, it turns the GPS on. If the GPS is acquiring lock when U^A is updated and if $U^A < AAU - \bar{s} \times \bar{t}_{lock}$ then node A should turn off its GPS , even without acquiring lock.

Figure 13 illustrates the impact of contact logging on GPS duty cycling for two nodes that are sending contact beacons every second. Without contact logging (Figure 13(a)), each node independently tracks its uncertainty estimate and acquires a GPS position fix whenever its uncertainty approaches AAU, resulting in 7 fixes for node 1 and 6 fixes for node 2. Using contact logging, the nodes can reduce their GPS fixes to 5 and 4 locks in the same time window. Consider the plot at 480 seconds. The uncertainty for node 2 approaches AAU, so it powers on its GPS and obtains a position fix. Node 1, which is in the vicinity of node 2, can rely on the latter's recent GPS fix to reduce its uncertainty to the 10m contact radius. This example illustrates the fairness feature in our contact logging strategy, since nodes that recently acquire lock will have a smaller uncertainty estimate than neighbours, forcing another node to turn on its GPS in the next round.

Our contact logging strategy depends on the settings of two variables: (1) the contact beacon period; and (2) the contact radius. The beacon period is the time between beacons at each node. It involves a tradeoff: short contact periods increase the likelihood of useful contact but incur a higher energy cost.

The contact radius specifies the value of R, which must balance the need to cover larger areas with a higher R and to reduce uncertainty with a lower R. Section 4.1 analytically explores this balance for the choice of R, followed by simulations based on empirical data to determine the operating points for beacon period and contact radius that balance these tradeoffs. Section 4.2 then investigates the impact of beacon period on the energy/accuracy tradeoff.

4.1 Contact Radius

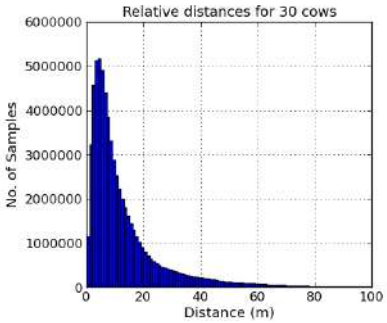


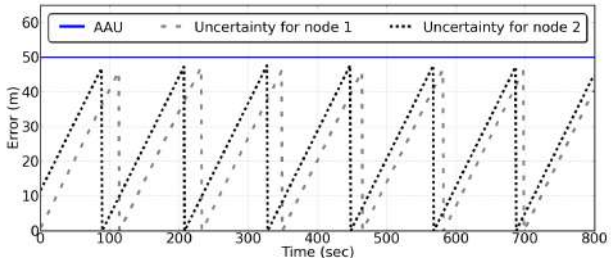
Figure 12. Relative distances between pairs of cows .

4.1.1 Analysis

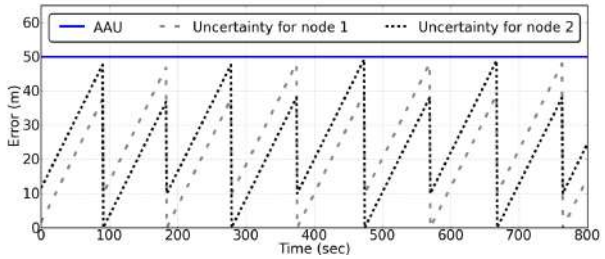
Instead of conservatively estimated $\hat{r} = R$, we could choose $\hat{r} < R/2$, since $R/2$ is the expected value of the inter-node distances if they are uniformly distributed. The choice of $\hat{r}$ can also rely on: offline statistics on inter-node distances from the 2-day experiment, shown in Figure 12; or RSSI values, as discussed in Section 4.1.3. For the moment, we focus on determining the features and dependencies of favourable contact radii.

In contact logging, the relative localization uncertainty between any two nodes that are in contact is $\hat{r}$, and it depends on the radio propagation factors and on the application requirements. It also relates to the value of AAU. Intuitively, $\hat{r}$ has to be less than AAU by at least $\bar{s} \times t_{\text{lock}}$. In other words, detecting that two mobile entities are in contact within a large distance is not useful if $\hat{r} > AAU - (\bar{s} \times t_{\text{lock}})$ is small.

Let $P(\hat{r})$ be the probability that the distance between two nodes is $\hat{r}$. From a single mobile node's perspective in an N-node network, the probability that at least one other node is closer than $\hat{r}$ is $P(contact) = 1 - (1 - P(\hat{r}))^{(N-1)}$. The probability that a neighboring node has a lower uncertainty is a function of the local node's uncertainty: when the local node's uncertainty approaches AAU, it is highly likely that a neighboring node's uncertainty is lower, since



(a) Error behaviour for GPS duty cycling



(b) Error behaviour for GPS duty cycling coupled with contact logging

Figure 13. Impact of contact logging on GPS duty cycles

the neighboring node runs the same GPS duty cycling algorithm and the expected value of $U = AAU/2$. Therefore, the probability that a neighboring node has a lower uncertainty $P(lower)$ than the local node is simply equal to $U(t)/AAU$. The likelihood that a neighbour with a lower uncertainty estimate will come into a node's radio range is simply $P_{use} = P(contact)P(lower)$. The selection of the best contact radius should maximize P_{use}:

$$\begin{aligned} P_{use} &= [1-(1-P(\hat{r}))^{(N-1)}] \times \frac{U(t)}{AAU} \\ where\ U(t) &= \hat{r} + (t - t_{contact}) \times \overline{s} \\ s.t.\ U(t) &< AAU \end{aligned} \tag{4}$$

and where $t_{contact}$ is the time of contact with a neighbor. The product of $(t - t_{contact}) \times \overline{s}$ quantifies the growth of local uncertainty since the initial contact and indicates an additional dependence of P_{use} on $\overline{s}$. P_{use} is a linear function of time represented by segment starting at $t_{contact}$ and ending at $\frac{AAU-\hat{r}}{\overline{s}}$. The segment length is shorter for larger $\hat{r}$, but $P_{contact}$ is also higher for larger $\hat{r}$, which suggests an optimal point exists. The best value for $\hat{r}$ should maximize the area below the trapezoid bounded by the P_{use} and the time axis:

$$\int_0^{\frac{AAU-\hat{r}}{\overline{s}}} P_{use} dt \tag{5}$$

We analyze the data from the field experiment in section 3.5 to first determine the probability density function of the distances between each pair of cows in the 30 cow data set. Figure 12 summarises the results. The inter-node distance is concentrated in the range 0-20m and suggests that setting the contact radius at or below 20m can be highly beneficial for cooperative localization, as $P(\hat{r} < 20)$ is nearly 73%. Using the virtual fencing dataset, we consider contact radius values at 5m, 10m, 20m and 30m with respective $P(\hat{r})$ of 0.26, 0.52, 0.73 and 0.81 and a speed of 0.4m/sec. Figure 14 compares the area under the P_{use} curve using equation 5 as N grows from 1 to 30 nodes. This shows a high dependence of optimal $\hat{r}$ on the number of nodes in the deployment, particularly that the optimal $\hat{r}$ shrinks with increasing number of nodes. The expected $\hat{r}$ that maximizes Puse for our 30 cow deployment is 5m.

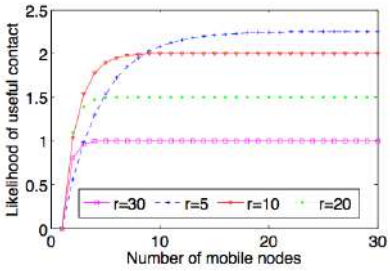


Figure 14. Effect of number of nodes and contact radius on P_{use}

4.1.2 Contact Radius Selection

We now explore the effect of the contact radius setting on performance. The main mechanism for setting the contact radius is to change radio receiver sensitivity and/or change radio transmit power. Given that these setting will not provide a fixed or uniform signal transmission, the value of R conservatively reflects the maximum contact radius.

To accurately model the tradeoffs of contact logging, we extended the Python simulator with a Message class. Each instance of the Message class represents a contact beacon which enables communication between instances of the Collar class. A transmitted contact beacon is registered as received if the distance between the sender and receiver is less than the contact radius. The time slot for sending contact beacons and the order in which nodes send and receive contact beacons are randomized to reflect reality. Each beacon contains the following information: node ID, node position, and node uncertainty.

Figure 15 summarizes the impact of contact radius on node energy consumption and error rate for the entire virtual fencing dataset. The contact beacon period is set to 1 second. The *overlap* parameter here for the DC_{MCU} is equal to DC_{gps}, as the frequent radio beacon transmissions and channel checking imply that the radio nearly always on when the GPS module is on. The optimal contact radius is 5m, which is in line with the analysis in Figure 14. Although contact logging reduced GPS duty cycle by up to 25%, the overall node power consumption is only reduced by about 15%, as contact logging increases the radio activity and associated MCU activity. In addition, contact logging incurs an increase in error rates over plain GPS duty cycling as nodes rely more on neighbours' estimates and exceed AAU more often. The radio power consumption notably increases for larger contact radii, which increases the range of node transmissions and causes more overhearing. The conclusion from these results is that simple contact logging with a fixed radius and a

1 second beacon period does not yield worthwhile benefits over simple GPS duty cycling.

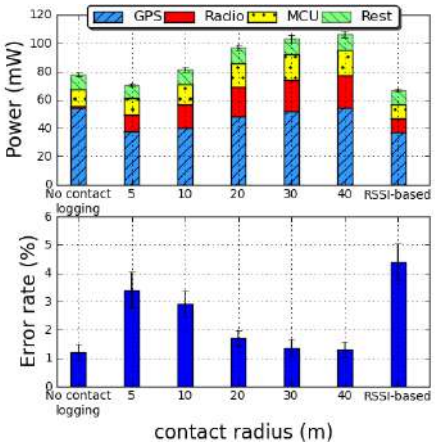


Figure 15. Effect of contact radius (static AAU)

4.1.3 RSSI-based Ranging

As mentioned above, estimating the distance $\hat{r}$ between two nodes as the radio range R is conservative. The use of received signal strength indicator (RSSI) can relax this estimate. While RSSI is a poor estimator of absolute range in mobile or non-line-of-sight environments due to multi-path signal propagation and interference, it still highly correlates with range between two nodes [7]. Here, we use RSSI to bound the range between two nodes as follows. Line-of-sight RSSI measurements with varying distance are collected prior to deployment in an open field. Let the line-of-sight RSSI value of two nodes that are $\hat{r}$ meters apart be X. If during a deployment, a node measures an RSSI of X from a neighbour, then the neighbour is at most $\hat{r}$ meters away.

Figure 16 shows 2400 line-of-sight RSSI measurements we collected for the Atmel RF212 radio that is used in the collar nodes. The measurements were collected by fixing a mobile node in a field near our office and moving another node within a 70m radius around the fixed node. Moving the transmitter in different directions around the receiver was to reduce any biases arising from the antenna radiation pattern not being fully omni-directional. Both nodes remained at an elevation of 1m from the ground. Each point in the figure corresponds to a single RSSI measurement. The solid line represents the maximum distance at which each discrete RSSI value was recorded by the RF212 radio in our measurements. For the purpose of radio ranging, we can use this line as a less conservative estimate of $\hat{r}$ for any RSSI X.

The last bar plot in Figure 15 shows the power consumption and error rate for RSSI-based contact logging. It clearly achieves the lowest power consumption relative to all contact radii and GPS duty cycling. In terms of error rate, RSSI-based contact logging causes an increase of 3% in error rate over GPS duty cycle yet remains well within the 5% target set for our application.

4.2 Beaconing Period

The baseline contact logging protocol is periodic, where each node has an internal timer for triggering transmission

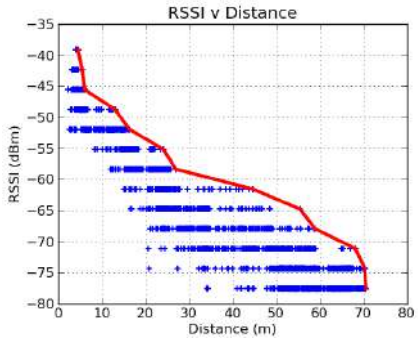


Figure 16. Line-of-Sight RSSI measurements for RF212 Radio. The bounding line represents the maximum possible line-of-sight distance for each observed RSSI value.

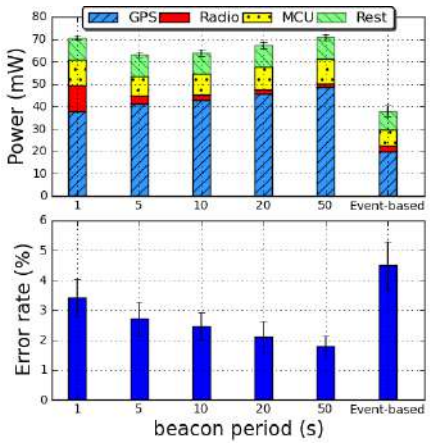


Figure 17. Periodic and event-based beacons

of contact beacons. Whenever neighbours hear contact beacons, they determine whether to rely on their neighbour's position estimate if that reduces their uncertainty. The contact period itself is dependent on many independent factors, including the clustering patterns of nodes, their relative speeds, terrain, and AAU. This suggests the need for adapting the contact beacon period to these factors, which is a non-trivial optimisation problem with highly dynamic and unpredictable inputs.

Before attempting to solve this problem, we first explore the impact of the contact beacon period on the accuracy/energy tradeoff in the first part of this section. Next, we propose an event-driven contact beacon transmission, based on local position state changes.

4.2.1 Periodic Beaconing

To investigate the effect of beacon period on the index, Figure 17 shows the power consumption and error rate for contact beacon periods of 1, 5, 10, 20, and 50 seconds. Increasing the contact beacon period from 1 to 5 seconds reduces both power consumption and error rate. While a longer beacon period causes GPS power consumption to increase, it also shrinks the radio power consumption. For beacon periods of 1 second, we observed that the GPS power consumption remains relatively high. The reason is that, on many occasions, a node powers its GPS ON to get lock, only to

turn it OFF, before getting lock, due to reception of useful message from neighbour. This continuous switching of the GPS module prevents it from lower its consumption further with short beacon periods.

4.2.2 Event-based Beaconing

An additional effect with periodic beaconing is the implicit synchronization of several neighbours of a node that obtains lock. When the node obtains lock, its beacon inform its neighbours of this event, and all neighbours set their uncertainties accordingly. If the neighbours happen to be within the same $\hat{r}$ from the node, and with similar assumed speeds $\bar{s}$, then they are likely to turn on their GPS modules at nearly the same time.

To counter this effect and to reduce radio power consumption further, we propose an alternative to periodic contact beaconing in the form of event-based beacons. In this approach, a node send its contact beacon whenever it locally detects one of the following position state changes: (1) its position is updated upon obtaining GPS lock; (2) its position is updated upon receiving a useful beacon from one of its neighbours; or (3) it has powered on its GPS module and is in the process of obtaining GPS lock. A beacon sent in state (3) serves as a GPS lock backoff beacon, and its purpose is to inform neighbouring nodes that it will imminently obtain lock, thereby avoiding the implicit synchronization issue.

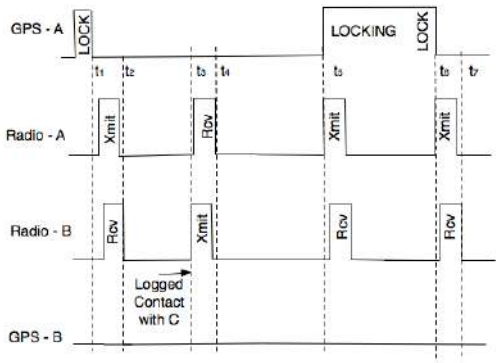


Figure 18. Event-triggered beaconing example: nodes only send beacons when they are getting lock, they get lock, or they log a useful contact with a neighbor

To support event-based beaconing, nodes use a modified beacon format that includes the expected time to obtain lock (based on history of lock times and off times).This beacon is known as the GPS lock backoff beacon. Figure 18 illustrates how event-based beaconing works. At time t_1, node A obtains a GPS lock. As a result, it sets it uncertainty to $U_{gps}(t_1)$ and broadcasts a beacon declaring its id, uncertainty, and position. Node B receives this beacon at time t_2, and, after determining that contact with node A reduces its own uncertainty, sets its uncertainty according to Equation 3. Nodes A and B then proceed to grow their uncertainty region according to their local speed models. At time t_3, node B hears a beacon from node C with a low uncertainty, and updates its local uncertainty region. Subsequently, it broadcasts a beacon announcing its updated uncertainty, enabling node A to reduce its uncertainty as well at time t_4. By time t_5, the uncertainty at node A grows close to AAU, so A decides to power on its GPS module to avoid exceeding AAU. It also broadcasts a GPS lock backoff beacon informing all its neighbours of this action to avoid multiple neighbours trying to simultaneously get GPS lock. Node B hears this beacon, and refrains from powering on its GPS module to get lock. At time t_6, node A succeeds in getting lock, reducing its uncertainty to $U_{gps}(t_6)$, and broadcasts a beacon informing neighbours of this change. Finally, B hears node A's beacon at time t_7 and updates its uncertainty as well. During all these exchanges, four radio packets are transmitted and received, node A's GPS obtained 2 locks, while node B's GPS module remained in off mode.

The last bar plot of of Figure 17 shows the results for event-based beaconing. With a slight increase in error rate, it nearly halves the power consumption relative to all of the beacon periods we consider. The energy saving from event-based beaconing stem from 3 factors: (1) the reduced radio beaconing enables the radio to sleep more often; and (2) the reduced overlaps in GPS lock attempts significantly reduce the GPS power consumption; (3) this implicitly enables the MCU to sleep more often as well, as most radio beacons are sent when the GPS is on.

Finally, we consider the effect of combining RSSI-based contact logging and event-based beaconing on error rate and power consumption. Figure 19 summarises the results of the various mechanisms in the paper, including GPS duty cycling, GPS duty cycling with contact logging (contact radius of 5m and beacon period of 5 seconds), RSSI-based contact logging with 5 second beacon period, event-based beaconing with a 5 m contact radius, and event- and RSSI- based contact logging. The latter yields the lowest power consumption at nearly 30mW; however, it causes the error rate to cross the error tolerance due to RSSI spreads. This means that event-based beaconing with a 5 m contact radius remains the most energy-efficient approach that meets the 5% error tolerance for the virtual fencing application.

4.3 Managing Performance

We now outline an adaptive duty cycling [8] online algorithm that we have implemented on the Fleck nodes and that delivers both required position performance and lifetime. As already discussed, each node evolves its uncertainty estimate while its GPS is powered off and uses opportunistic contact logging to reduce that uncertainty.

Whenever a node receives a GPS position lock, it determines if the fix falls more than the distance AAU from the previous fix. The node maintains a circular list of the outcome of this test for the last M samples. When the error rate in this list exceeds σ^*, the node is not meeting the position performance criteria so the assumed speed is incremented and the GPS duty cycle will increase. On the other hand, if the error rate is below a different threshold the node is over-delivering in terms of positioning performance and using a higher duty cycle than it needs, in which case the assumed speed is decremented. The logic also takes into account the current energy state $E(t)$ and the desired energy state E^* required to meet the lifetime objective. The algorithm is shown in figure 20. To resolve any conflicts between the energy and accuracy constraints, the user preference input sets the prior-

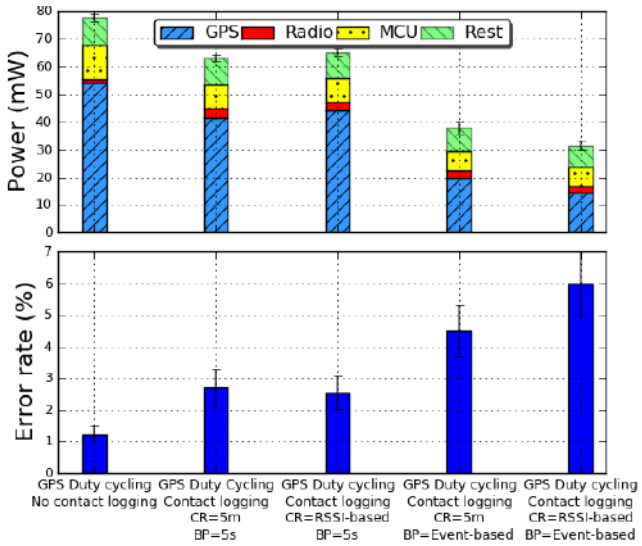


Figure 19. Comparison of configurations. Combination of RSSI-based and event-based beaconing has the lowest power consumption, but exceeds error tolerance.

ity of increasing speed to lower the error rate or decreasing speed to increase the lifetime.

loop {every GPS fix}
  d(t)= $\sqrt{[x(t)-x(t_0)]^2+[y(t)-y(t_0)]^2)}$
  if d(t) > AAU **then**
    list.insert(1)
  else
    list.insert(0)
  end if
  if $\frac{list.sum}{M} > \sigma^*$ **then**
    if user pref == accuracy OR $E(t) > E^*$ **then**
      $\overline{s} = \overline{s} + \delta^+$
    end if
  end if
  if $\frac{list.sum}{M} < \sigma^*$ **then**
    $\overline{s} = \overline{s} - \delta^-$
  end if
end loop

Figure 20. Adaptive Duty Cycling Strategy

We implemented this strategy in our simulator and set a power consumption of 44 mW for achieving 3 months lifetime with the 4 D-cell batteries. The initial battery energy was scaled down to match the 2-day duration of the data set. The error tolerance is 5%. We set δ^+ to 3-$\overline{s}$ and δ^- to 0.1 to implement fast response to errors and slow energy recovery. For $M = 1$, Figure 21 shows the impact of the user preference metric on the residual energy and the error rate. The algorithm delivers the desired lifetime when the energy preference is set, and approaches the error tolerance when the accuracy preference is set. Note that the measured error rate indicates the node's online measurements, while the real error rate is the ground truth value.

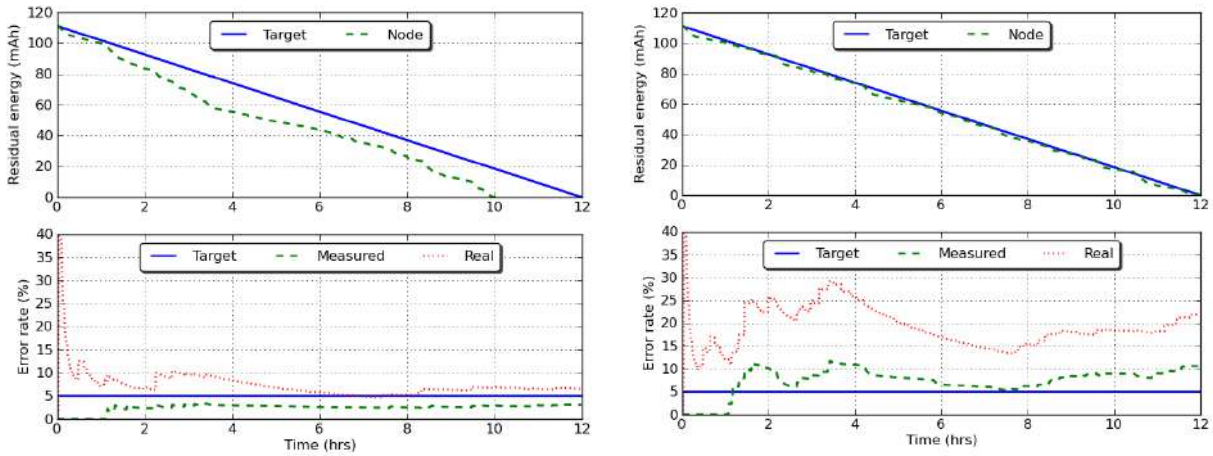
5 Related Work

The first instance of animal tracking by networked embedded systems is the ZebraNet project [9], which tracks individual position records for animals every of a few minutes. However, ZebraNet collars include a solar panel, making the energy management problem quite a tractable one. Positioning is done by GPS only, and the nodes flood their information by flooding in order to facilitate data acquisition by the mobile sink. The Electronic Shepherd project [10] also uses GPS localization for tracking herds of sheep in the mountains. "Real-time" is obtained thanks to a GPRS modems in the collar, even though position is recorded and sent back to base only a few times per day in order to prolong battery life. Unlike our work, nodes are asymmetric: only a few animals in the herd wear collars including GPS and GPRS modules. All the others wear a small ear-tag including only a microcontroller and a radio-chip.

The work in [11] considers energy efficient localization for base nodes on the path of the mobile nodes, and the energy implications of each approach. While the speed models in [11] are similar to this paper, their work is purely simulation-based and does not consider the benefits of contact logging or online adaptation according to energy and position changes. The work in [12] addresses the tradeoff between localization accuracy and energy efficiency. It considers static, dynamic, and dead reckoning mobility models and studies their effect on accuracy and energy through ns2 simulations. Our work addresses the same tradeoff but uses empirical GPS data to conduct analysis on three different speed models and a realistic energy model of the GPS-enabled nodes that includes the previously unreported stochastic lock time model. We also consider the effect of dynamically changing the required localization performance based on the mobile node's distance from an exclusion zone. Further we explore contact logging as a complement to GPS for achieving a better balance between accuracy and efficiency. In a similar fashion, Pattern et al. [13] provide a tuning knob to obtain various energy-accuracy tradeoffs by the careful choice of duty cycle and activation radius.

The works in [14] and [15] address this tradeoff as well by considering improvements to the RIP method [16] as the baseline localization method. A key difference between You's work and ours is that we support a variable AAU. Another distinction is their use of mote-based RSSI localization schemes in indoor environments, which require a deployed network infrastructure, consume less power, and require a calibration phase. Finally, they do not consider contact logging for reducing localization uncertainty.

Recent work [17] [18] has also explored collaborative localization that enables nodes to detect neighboring signals for position refinement. Chan et al. [17] use a cluster-based approach, where nearby nodes can establish clusters through IEEE 802.15.4 radios as a neighbour detection sensor that enhances WiFi localization. Our approach is similar in its reliance on contact signals, but it differs in that the contact signal is sent and received by the same radio, eliminating the need for multi-radio nodes. Liu et al. [18] also use a form of contact logging that establishes anchor nodes that know their location, pseudo-anchor nodes that estimate their location



(a) User preference set to accuracy

(b) User preference set to energy

Figure 21. Effect of user preference

based on contact with anchor nodes, and free nodes that rely on anchor or pseudo-anchor nodes for their location. This resembles the multi-hop contact logging in our virtual fencing application. However, their work focuses on algorithms to bound the uncertainty propagation in iterative localization techniques through the use of static anchor nodes, while we focus on bounding the localization uncertainty of mobile anchors and neighbouring nodes that rely on them within an energy constraint. In addition, nodes dynamically become anchors in our model based on their current GPS state.

More recently, research has explored energy-aware localization for mobile phones equipped with GPS. For example Constandache *et. al* [19] propose an average error metric for GPS duty cycling. Using proximity mechanisms like Wi-Fi or GSM yields a lower average error for the same energy budget as plain GPS duty cycling. One of their mobility models deduces the current location of the node from recent history when the user is driving on a straight road. Similarly, the work in [20] stores historical information of locations where GPS does not work, and of average speed as a function of time and place. Our algorithm stores no neighbour history, where each node tracks only its own uncertainty through contact logging with neighbours. This minimises local storage requirements for individual nodes and training requirements for establishing historical patterns.

The use of accelerometers has also been proposed as a low power indicator of movement to supplement GPS duty cycling [20][21]. The inclusion of inertial sensor readings, such as magnetometers or accelerometers, is an attractive extension to our approach, which we discuss in the next section. Two recent works [20][22] consider Bluetooth as a radio ranging technology, since it is pervasive in mobile phones. Bluetooth is relatively energy-hungry for neighbour discovery and frequent beaconing, as it relies on a synchronous connection-oriented master slave topology. Our work uses lower power IEEE 802.15.4 for radio ranging that are use asynchronous broadcast channels and a flat topology. This enables a single beacon to be detected by all neighbours within radio range, providing a more scalable approach.

6 Conclusion and Perspectives

This paper has proposed and validated short-range radio contact logging as an effective complement to GPS duty cycling for balancing the positioning accuracy and energy-efficiency tradeoff. Given a user policy that comprises a target lifetime, acceptable uncertainty, and error tolerance, we have established a strategy for tailoring the contact radius and beacon period for any mobile sensor network application. For RSSI-independent contact logging, the strategy considers the distribution of relative distances between the mobile nodes to analytically determine the optimal contact radius. We have confirmed that this contact radius performs best through simulations based on our empirical data set. We then proceeded to identify the most suitable beacon period for our dataset, based on which we proposed event-driven beaconing that reactively sends beacons only when nodes detect state changes.

While the event-driven beaconing strategy yields the lowest power consumption in our scenario, the optimal strategy is highly dependent on the mobility pattern of the nodes, their clustering characteristics, and user policy. For instance, an application with highly mobile nodes may benefit from a more proactive beaconing strategy that transmits beacons at short periodic intervals. Similarly, while RSSI-based contact logging outperformed all statically set contact radii, it may not suit radios with no RSSI or high multi-path deployments.

Assisted GPS is a mechanism that uses cellular phone base stations to provide ephemeris data (satellite date) to mobile nodes in order to reduce their GPS lock time. In WSN deployments with both static and mobile nodes, the static nodes can supply the ephemeris data to the mobile nodes, which in turn seed their GPS receivers on startup with this data to achieve a lock time of 4 seconds (time-to-first-fix). A challenge in using assisted GPS is that the receiver's ephemeris data must be erased every time. If a node ever needs to switch back to using standard satellite based GPS, the GPS receiver will take longer than normal to obtain a fix.

The static nodes that serve as a routing backbone for the

mobile nodes could also serve as local positioning anchors outside the scope of assisted GPS. The area of relative positioning through powerful local beacons is indeed a well-investigated area. While our current virtual fencing deployments do include static nodes that can be used for this purpose, we are moving towards more sparse static node deployments for high scalability in monitoring mobile cattle. The GPS duty cycling and radio contact logging strategies here represent a step towards more sparse deployments.

Another possible enabler of more sparse deployments is the use of inertial sensors, such as accelerometers and magnetometers, to refine position estimates while the GPS is powered off. While this is a proven strategy in other GPS tracking scenarios, its utility for low power devices is an open question. The accelerometer in particular involves an inherent tradeoff between its accuracy for predicting speed and its sampling rate. It can only provide precise estimates of speed for high sample rates, which incurs high energy cost. The use of accelerometers as boolean indicators of motion is certainly an option for input to the algorithm. Magnetometers can certainly provide accurate heading data for low energy cost, thereby limiting the uncertainty growth within a angular cone rather than a circle. This comes at a cost of increasingly computational complexity for growing the irregular uncertainty region as the mobile nodes change directions.

Some concepts from this paper, such as dynamically changing the AAU according to distance from the fence, are specific to the cattle monitoring application. However, our overall strategy is applicable to other mobile tracking applications, such as smart phone social networking applications. Consider groups of friends with smart phones that regularly meet and use location-based services that rely on GPS. If all smart phones keep their GPS modules on to run the services, they will most likely deplete their batteries quickly. Alternatively, the use of contact logging to share the GPS load among the smart phones can prolong the lifetime of all devices. Applying our strategy to this application would require mobility models, clustering patterns, and application-specific user policies to determine the best configuration.

Acknowledgements

The authors would like to thank Chris Crossman, Greg Bishop-Hurley, Philip Valencia, Brano Kusy, and Alban Cotillon for their valuable inputs in realising this work.

7 References

[1] M. Ilyas and R. C. Dorf, editors. *The handbook of ad hoc wireless networks*. CRC Press, Inc., Boca Raton, FL, USA, 2003.

[2] T. Wark et al. The design and evaluation of a mobile sensor/actuator network for autonomous animal control. In *IPSN '07*, 2007.

[3] R. Jurdak, A. G. Ruzzelli, and G.M. P. O'Hare. Adaptive radio modes in sensor networks: How deep to sleep? In *IEEE (SECON)*, June 2008.

[4] Z. Butler et al. From robots to animals: Virtual fences for controlling cattle. *Int. J. Rob. Res.*, 25(5-6), 2006.

[5] Y. Guo et al. Using accelerometer, high sample rate gps and magnetometer data to develop a cattle movement and behaviour model. *Ecological Modelling*, 220(17), 2009.

[6] G. Lin, G. Noubir, and R. Rajaraman. Mobility models for ad hoc network simulation. In *INFOCOM*, 2004.

[7] K. Srinivasan and P. Levis. Rssi is under appreciated. In *EmNets*, 2006.

[8] R. Jurdak, P. Baldi, and C. Videira Lopes. Adaptive low power listening for wireless sensor networks. *IEEE Tran. on Mobile Computing*, 6(8):988–1004, 2007.

[9] P. Zhang and C. Sadler et al. *Habitat Monitoring with ZebraNet: Design and Experiences*, chapter Wireless Sensor Networks: A Systems Perspective. Artech House, 2005.

[10] Thorstensen B. and T. et al. Syversen. Electronic Shepherd – a low-cost, low-bandwidth, wireless network system. In *MobiSys*, 2004.

[11] J. Winter, Y. Xu, and W.C. Lee. Prediction-based strategies for energy saving in object tracking sensor networks. In *IEEE ICMDM*, January 2004.

[12] S. Tilak et al. Dynamic localization control for mobile sensor networks. In *IPCCC*, 2005.

[13] S. Pattern, S. Poduri, and B. Krishnamachari. Energy-quality tradeoffs in sensor track- ing: Selective activation with noisy measurements. In *IPSN*, 2003.

[14] C.-W. You. *Enabling Energy-Efficient Localization Services on Sensor Network Positioning Systems*. PhD thesis, January 2008.

[15] You et al. Impact of sensor-enhanced mobility prediction on the design of energy-efficient localization. *Ad Hoc Netw.*, 6(8), 2008.

[16] B. Kusy et al. Radio interferometric tracking of mobile wireless nodes. In *MobiSys*, 2007.

[17] L. Chan et al. Collaborative localization: Enhancing wifi-based position estimation with neighborhood links in clusters. In *Pervasive*, 2006.

[18] J. Liu, Y. Zhang, and F. Zhao. Robust distributed node localization with error management. In *MobiHoc*, 2006.

[19] I. Constandache, S. Gaonkar, and et al. Enloc: Energy-efficient localization for mobile phones. In *INFOCOM*, pages 2716–2720, 2009.

[20] J. Paek, J. Kim, and R. Govindan. Energy-efficient rate-adaptive gps-based positioning for smartphones. In *MobiSys*, 2010.

[21] M. B. Kjaergaard et al. Entracked: energy-efficient robust position tracking for mobile devices. In *MobiSys*, 2009.

[22] K. Lin, A. Kansal, and et al. Energy-accuracy tradeoff for continuous mobile device location. In *MobiSys*, 2010.