Efficient Node Discovery in Mobile Wireless Sensor Networks

Vladimir Dyo, Cecilia Mascolo

[1] Department of Computer Science, University College London
Gower Street, London WC1E 6BT, UK
v.dyo@cs.ucl.ac.uk
[2] Computer Laboratory, University of Cambridge
15 JJ Thomson Avenue, Cambridge CB3 0FD, UK
cecilia.mascolo@cl.cam.ac.uk

Abstract. Energy is one of the most crucial aspects in real deployments of mobile sensor networks. As a result of scarce resources, the duration of most real deployments can be limited to just several days, or demands considerable maintenance efforts (e.g., in terms of battery substitution). A large portion of the energy of sensor applications is spent in node discovery as nodes need to periodically advertise their presence and be awake to discover other nodes for data exchange. The optimization of energy consumption, which is generally a hard task in fixed sensor networks, is even harder in mobile sensor networks, where the neighbouring nodes change over time.

In this paper we propose an algorithm for energy efficient node discovery in sparsely connected mobile wireless sensor networks. The work takes advantage of the fact that nodes have temporal patterns of encounters and exploits these patterns to drive the duty cycling. Duty cycling is seen as a sampling process and is formulated as an optimization problem. We have used reinforcement learning techniques to detect and dynamically change the times at which a node should be awake as it is likely to encounter other nodes. We have evaluated our work using real human mobility traces, and the paper presents the performance of the protocol in this context.

1 Introduction

Energy efficiency is a crucial aspect in wireless sensor networks. The amount of energy of a sensor network may be limited by the constrained size of devices or, for instance, by the efficiency of the source of energy, e.g., the limited size of a solar panel. In such situations, the only sensible approach to energy saving is duty cycling, i.e., the control of the awake times of sensor nodes.

Duty cycling however, limits the ability of nodes to discover each others as when nodes are sleeping they cannot detect contacts. The problem of neighbour detection is even more serious if the sensor network is mobile, as the topology in these networks changes rapidly.

Node detection is not a problem if nodes are equipped with specialized sensors, such as motion detectors or accelerometers. However, these devices increase the cost and the size of the equipment and are not always available or deployable (e.g., in some zoological applications which need cheap or very small sensors). In this paper we will assume that the detection of neighbours only happens through normal short-range radio.

The existing work on duty cycling [1] has mostly tackled static networks with fixed topologies and is not applicable to mobile scenarios, given the variability of the topology. A major challenge for some mobile networks is the uncertainty of the node arrival time. If the node arrival time is not known, the only chance a node has to discover all the nodes passing by is to be always awake, which is very energy inefficient.

MAC layer optimizations for listening times such as the ones developed in [2] offer some form of optimization of the power consumption, however not at the level of granularity which could be achieved with patterns recognition. A considerably better optimization can be achieved by using some knowledge of the encountering patterns in the network in order to decide when to switch on (and off) the radio. This, of course, can only be applied when encounter patterns exist, which however is often the case in wildlife and human applications.

In this paper we propose an energy efficient node discovery approach for mobile wireless sensor networks. The main idea of our method is the online detection of periodic patterns in node arrivals and the scheduling of wake-up activity only when contacts are expected. The approach is based on reinforcement learning techniques [3]. In this approach each node monitors the number of encounters depending on time of day and concentrates more energy budget (i.e., more awake time) into predicted busier timeslots. The approach also allows for some extra energy to monitor other timeslots in order to cope with variation in the patterns and to refine what it has learned, dynamically.

The approach can be applied to scenarios such as wildlife monitoring, as indicated above, or human-centric networks. In order to evaluate the performance of the approach we have verified it with real human mobility traces, used to drive mobile sensor movement in a synthetic way, in a simulator.

The rest of the paper is organized as follows: Section 2 contains a general overview of our approach. Section 3 contains the adaptive technique for learning arrival patterns. Section 4 describes the protocol. Section 5 present an implementation and evaluation of our approach, respectively. Section 6 discusses related work with conclusions and possible future work.

2 Overview of the Approach

The main goal of our approach is to allow nodes to detect each other's presence but, at the same time, to save energy by switching off their radio interface as much as possible. As we outlined in the introduction, the detection of neighbours allows many activities such as the relaying of the data to sinks and the logging of encounters.

Discovering nodes is expensive and requires either periodic scanning (as in Bluetooth) or periodic continuous transmission of a radio tone (if the nodes are using a Low Power Listening based protocol [2]). A high scanning rate will guarantee quick discovery but will waste energy, especially in situations, where no encounters are likely to occur. On the other hand, a low scanning rate can miss many important contacts. Specifically, the goal of the approach is to devise a simple adaptive algorithm to control the scanning rate, considering past encounter history.

We consider duty cycling as a sampling process. Intuitively, to detect more encounters, a node needs to sample more frequently when more encounters are expected. Moreover, the node should avoid to sample when no encounters are expected. Thus, the goal is to maximize:

$$R(a) = \sum_i E_i * d_i, \quad s.t. \sum_i d_i < D_{budget} \tag{1}$$

Where E_i and d_i is an expected number of encounters, a a duty cycle at timeslot i, and D_{budget} is a daily energy budget. As we see from the equation, there is a balance between number of contacts and energy consumption. Thus it seems natural to formulate the problem as a maximization of the number of successful encounters per unit of energy consumption.

As already indicated, we consider a specific class of applications, when periodic encounter patterns exists. These represent a large class, which include human and animal life.

3 Learning Arrival Patterns

In this section we describe the core ideas behind the pattern arrival mechanism we adopted. The basic behaviour of the algorithm drives each node to estimate the hourly activity of its neighbours and to progressively concentrate the discovery process only when encounters are expected. Indeed, the intuitive idea behind this behaviour is that continuously scanning for neighbours when no one will be around implies a waste of energy.

3.1 Model

We now introduce the formal model behind the approach. An agent (in the reification of our system, a node)[3] interacts with the environment through perception and action. At each step, an agent observes the state $s_t \in S$ of an environment and responds with an action $a \in A(S_t)$. The action results in a certain reward $R : S \times A \to R$. The goal of an agent is to maximize a long-term reward based on the interactions with an environment. Specifically, the goal is to learn a policy mapping from states to actions that maximizes the long-term agent reward.

[3] In this section we will refer to node and agents referring to the same entity: agent is the name used in the machine learning theory we adopt.

A day is modelled as N timeslots. A node has the following set of actions: i) sleep ii) wake-up ii) set duty cycle (1-100%). A node controls the duty cycle by changing the discovery beacon rate. A high duty cycling might or might not increase the chances of detecting more contacts. For example, it might be sufficient for a node to work from 11am to 12am, but with a 10% duty cycle (as opposed to 100%). A reward r is the number of successful encounters. The goal of an agent is to detect the maximum number of successful encounters within a given energy budget.

After taking each action, a node observes the outcome and updates the payoff for a given timeslot. The payoff estimation is done using an exponentially weighted moving average (EWMA) filter. The filter estimates the current payoff value by taking into account the past measurements, $r_n = r_{measured} * \alpha + r_{n-1} * (1 - \alpha)$ Where r_n and r_{n-1} are respectively the estimated and previous payoff values. $r_{measured}$ is the measured payoff over the last time slot. The weight assigned to past measurements $(1 - \alpha)$ depends on how responsive the node has to be to changing environment.

We now describe the balanced strategy which could be used to adapt the node's duty cycle and a random strategy which we will use as baseline for the evaluation.

Balanced. In a balanced strategy we propose to dynamically adjust the node's duty cycle proportionally to an expected reward. Therefore the node *does not commit* to any timeslot, but spreads its energy proportionally to the expected reward. The node sets its duty cycle according the following rule:

$$D(a) = \frac{r(a)}{\sum_{a\prime \in A} r(a)} \tag{2}$$

$D(a)$ is a duty cycle in the current timeslot, $r(a)$ is an expected reward from taking an action a. It is computed as indicated in Formula (3.1) For example, if there are several peak hours during a day, the budget will be spread evenly among all peaks. During quiet times the node continues to sample the environment but with lower intensity.

Random. In a random strategy (which we use for comparison) the node spreads its energy budget evenly throughout a day, i.e., it sends beacons with a certain fixed interval. The strategy is equivalent to normal asynchronous wake-up scheduling with fixed duty cycle, so would not require additional implementation. The obvious problem with the random strategy is that a node will waste resources when there are no nodes around. This will become evident in our evaluation.

4 Algorithm

In this section we present an algorithm for adaptive node discovery. The algorithm should allow the detection of 'quiet' periods and exclude them from the discovery process, allowing the node to sleep in that time for as much as possible. The daily budget assignment could be performed by the application, depending on the known energy availability: for example in the scenario we envisage, it is very clear how big the batteries can be and how long the zoologists want them to last for, therefore the daily budget can be inferred.

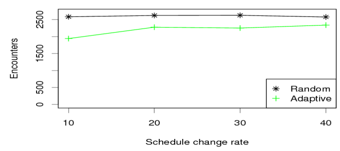
1. The node starts by following a random strategy, i.e., spreads its duty cycle equally in each timeslot. As it discovers new nodes it dynamically reajusts its budget according to the following steps.
2. Once discovered, the nodes remain synchronized for a duration of an encounter. Short term synchronization is possible with built-in timers without the need of globally synchronized clocks. As long as there is at least one node in range, the node sends periodic keep-alive messages every $T_{keepalive}$ seconds.
3. If a node does not hear from a neighbour for T_{expire} seconds, it assumes an encounter is terminated and increments the timeslot counter. $C_t = C_t + 1$.
4. At the end of each day a node updates its timeslot counters M_t using a EWMA smoothing filter: $M_t = C_t * \alpha + M_t * (1 - \alpha), t = 0..N_{slots}$. Where M_t is an estimated encounter frequency at timeslot t and C_t is the actual number of encounters in timeslot t registered during current day. The node then resets the daily counters C_t.
5. At the beginning of the current timeslot *(t)*, a node sets a beacon rate to be: $F_{beacon} = \frac{M_t}{\sum_i M_i} \frac{B}{E_{beacon}}$. Where B is a daily energy budget, E_{beacon} is an amount of energy required to scan the neighbourhood. The node converts the beacon frequency into interval time between beacons $T_{beacon} = 1/F_{beacon}$. If the duration of this period is longer than the timeslot duration $T_{timeslot}$, the node beacons with a probability $p = \frac{T_{beacon}}{T_{timeslot}}$. The node then schedules the next wake up by the beginning of the next timeslot. The node has preconfigured minimum and maximum beacon rates F_{min} and F_{max}. The minimum beacon rate is needed to guarantee a certain level of exploration, even when no discovery is expected. The maximum beacon rate limits the amount of energy a node spends in one timeslot.

5 Evaluation

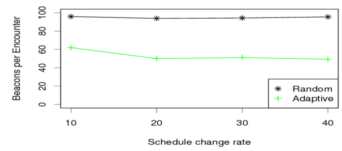
The goal of experiments is to compare the random and adaptive algorithms presented in Section 4 on the performance of two basic applications: encounter tracking and message dissemination.

5.1 Evaluation Settings

Dataset. We evaluated our approach through simulations in TOSSIM with real human connectivity traces to emulate node mobility and dynamic encountering. We used human mobility traces from MIT reality mining bluetooth traces [4] to drive the sensor movement like their were tagged individuals. The traces were collected using 96 people carrying Bluetooth mobile phones over a duration of 292 days. The evaluation was done on 60 more active nodes over 3 months



(a) Encounters



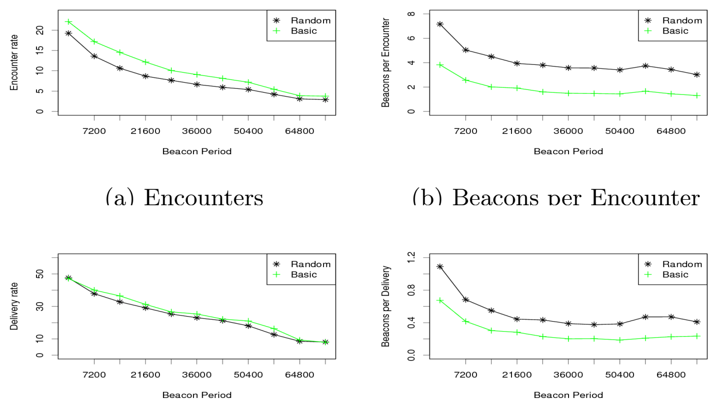
(b) Beacons per Encounter

of traces. Due to power limitations, the original traces are result of sampling every 300s, which might have missed some encounters and introduced a certain granularity of encounter duration. In this paper, however, we assume that the traces represent ground truth data about physical movement of entities and that our optimal result would be to detect all contacts. All the nodes were booted at random times between 0 and 3600 seconds. The evaluation was done over 5 runs for each algorithm x budget combination.

Impact on discovery rate. In the first experiment we measure encounters between the nodes for various wake-up algorithms and compare with a baseline random algorithm over synthetic traces in the following settings. We generated 7200 random encounters for 36 nodes for a duration of 90 days. The duration of each encounter was uniformly distributed between 300 and 900s. To model dynamic environment the network operated according to one of two schedules. In schedule A, all the links were established between times 8am and 3pm; in schedule B, all the encounters were established between times 22pm and 5am. We then generated a trace, where both schedules alternated every 10, 20, 30 and 40 days. All nodes were running encounter tracking application and were required to detect and log encounters between the nodes. The beaconing is the expensive process which requires nodes to stay up for a long period of time. In our experiments we measure the energy a node spends on beaconing (node discovery) on the performance of basic applications of encounter logging.

Figure 1a shows that the nodes running adaptive strategy managed to detect almost the same number of encounters as nodes running random strategy. At the same time, the adaptive strategy required up to 50% fewer beacons than a random one (Figure 1b). The performance degraded with more frequent schedule changes, but remained higher than random.

In the second experiment we measure encounters between the nodes over the real traces from MIT reality mining experiments. In the course of experiments, we observed that the number of detected encounters of the adaptive algorithm depends on the maximum and the minimum number of beacons in one timeslot. In the experiment we set it to maximum of 200% and 10% of average (budgeted) scanning rate. All the graphs show the percentage of encounters detected by the 2 algorithms over the total number of encounters in the traces. We then tested the algorithm sensitivity for different timeslot durations and found that longer timeslots perform better for lower scanning rates. In the following experiments



(a) Encounters (b) Beacons per Encounter

(c) Deliveries (d) Beacons per Delivery

Fig. 1: Impact on node discovery and message delivery rates.

the nodes used 1 hour timeslots for scanning rates up to 3600 and 3 hour timeslots for lower scanning rates.

Figures 1a shows the number of detected encounters for scanning intervals from 7200 to 79200 seconds. The graph shows that the adaptive strategy detects more encounters than simple random strategy. It shows that while adaptive detected more encounters, they consumed much fewer beacons (Figure 1b).

Impact on message delivery rate. In this experiment we measure the impact of wake-up strategy on message delivery rates for a simple data collection application. The nodes are using a direct delivery algorithm, in which a sender delivers a message directly upon an encounter with a destination node (e.g., a sink in our scenario). The nodes were configured to generate one message per hour and send it towards one of six sinks in MIT reality mining traces. The message was considered delivered when it reached at least one of the sinks. The sinks were chosen randomly at each simulation run. All the graphs show the delivery rate in percentage from the total number of generated messages.

Figure 1c shows the number of detected message deliveries for scanning intervals from 7200 to 79200 seconds. The adaptive strategy provides better results than simple random strategy. Figure 1d shows the average number of beacon per delivery for the same experiment. It shows that while adaptive provides higher delivery rates, it consumed much fewer beacons.

It is interesting to note that at a beacon rate of one beacon per day, adaptive strategy still maintains a delivery rate of about 40%, more than twice that of random strategy while consuming twice as few discovery beacons.

6 Related work and Conclusion

Energy efficient service discovery can be done using power efficient wake-up scheduling protocols, such as [1, 2]. These protocols allow for very energy efficient communication in static wireless sensor networks. They are not, however, able to exploit the fact that contact patterns might be regular and distributed in such a way that there are periods in which nodes do not encounters other nodes. Our approach works on top of existing wake-up scheduling protocols, allowing them to make better decisions as to when to and how frequently to perform service discovery. In [5] an adaptive node discovery approach is proposed for static sink nodes to track mobile objects: some learning techniques have been used there to drive the discovery, however the network of sensors for that paper was static not allowing for the variability inherent in a mobile sensor network. A *variable inquiry rate* has been used to collect Bluetooth traces in [6]. To save power the nodes were configured to sample the neighbourhood more frequently when no nodes are detected and then reduce the sampling rate if there are nodes around. Although an approach was used to actually collect traces, there was no evaluation quantifying an impact of this technique on the number of detected encounters.

We have presented an approach for flexible duty cycling for mobile wireless sensor networks. We have evaluated the approach with realistic human mobility traces and have shown the performance of our proposed approach with respect to a random wake up scheduling. We are in the process of generalization of the approach to non-periodic patterns. In this case, the node needs to forecast the encounter pattern for the next N steps and then allocate energy budget accordingly.

References

1. Ye, W., Silva, F., Heidemann, J.: Ultra-low duty cycle MAC with scheduled channel polling. In: SenSys'06: Proceedings of the 4th international conference on Embedded networked sensor systems, New York, NY, USA, ACM Press (2006) 321–334
2. Polastre, J., Hill, J., Culler, D.: Versatile low power media access for wireless sensor networks. In: SenSys'04: Proceedings of the 2nd international conference on Embedded networked sensor systems, New York, NY, USA, ACM Press (2004) 95–107
3. Kaelbling, L.P., Littman, M.L., Moore, A.P.: Reinforcement Learning: A Survey. Jurnal of Artificial Intelligence Research **4** (1996) 237–285
4. Eagle, N., Pentland, A.S.: Reality mining: sensing complex social systems. Personal Ubiquitous Comput. **10**(4) (2006) 255–268
5. Dyo, V., Mascolo, C.: A Node Discovery Service for Partially Mobile Sensor Networks. In: MIDSENS'07: Proceedings of IEEE International Workshop on Sensor Network Middleware, Newport Beach, California, USA, IEEE Computer Society (November 2007)
6. Su, J., Chan, K.K.W., Miklas, A.G., Po, K., Akhavan, A., Saroiu, S., de Lara, E., Goel, A.: A preliminary investigation of worm infections in a Bluetooth environment. In: WORM'06: Proceedings of the 4th ACM workshop on Recurring malcode, New York, NY, USA, ACM (2006) 9–16