Title	Combining sensor selection with routing and scheduling in wireless sensor networks
Author(s)	Muldoon, Conor; Trigoni, Niki; O'Hare, G. M. P. (Greg M. P.)
Publication date	2011-08-29
Conference details	Paper presented at the 8th International Workshop on Data Management for Sensor Networks (DMSN'11), held in conjunction with the International Conference on Very Large Data Bases (VLDB), Seattle, USA, 29 August 2011
Item record/more information	http://hdl.handle.net/10197/3510
Publisher's statement	This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version will be published in the This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in the Proceedings of 8th International Workshop on Data Management for Sensor Networks (DMSN'11).

Combining Sensor Selection with Routing and Scheduling in Wireless Sensor Networks

Conor Muldoon and Niki Trigoni
Department of Computer Science
University of Oxford
{conor.muldoon, niki.trigoni}@cs.ox.ac.uk

Gregory M. P. O' Hare
CLARITY: The Centre for Sensor Web Technologies
School of Computer Science and Informatics
University College Dublin
gregory.ohare@ucd.ie

ABSTRACT

In wireless sensor networks, determining the set of locations to activate sensors, such that the amount of information received is maximised, is an important task. With the model driven approach to sensor networks, a considerable amount of research has been conducted into the sensor selection problem, whereby informative locations to activate sensors are determined through the use of Gaussian Processes. Current approaches to sensor selection, however, do not take bandwidth constraints into consideration. When bandwidth is limited, the amount of time required to realise a given set of communication requests, in terms of routing and scheduling, must be accounted for. Traditionally, in wireless sensor networks, sensor selection, routing, and scheduling are considered separately. These three processes, however, are highly coupled and the overall utility of the data collection process, in terms of the amount of information it produces, can be improved by integrating them. In this paper, we present two approaches to combining routing, scheduling, and sensor selection such that they inform each other so as to improve the performance of the network in the bandwidth limited case.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms

Keywords

Sensor Selection, Gaussian Processes

1. INTRODUCTION

The model driven approach to data acquisition in sensor networks provides a way to balance the trade-off between the quality of the sensed measurements and the costs incurred in retrieving the data. Sensor readings are typically correlated over space and considerable research has been invested into approaches that place sensors, or select sensors to be active, at informative locations using Gaussian Process (GP) models of the spatial phenomena [10, 3, 11]. GPs provide a way of performing regression for (1) locations where there are inactive sensors and (2) other locations of interest, such as locations where no sensors can be placed, hazardous locations, or arbitrary query locations. When performing sensor selection with GPs, submodular objective functions are used to model diminishing returns. This captures the intuitive idea that the more sensors that are active in a particular area, the less utility increases by activating additional sensors in that area.

One of the limitations of current work in sensor selection [10, 3, 11], however, is that it does not take bandwidth constraints into consideration. In prior work there is only a single constraint in the network, such as a constraint on the number of active sensors or a limit on the total expected number of message transmissions [11]. Although a constraint on the number of sensors can be used to determine informative locations for node placement, when performing sensor set selection, or choosing a subset of sensors to be active[1], it will not provide a good approach in the case where bandwidth is limited and the time required to realise the communication links must be taken into consideration. Just using a single constraint on the total expected number of message transmissions will also perform poorly in this case.

In this paper, sensor selection is performed subject to bandwidth constraints. With this approach, an epoch is specified and a TDMA MAC layer is used to facilitate data transmission in an interference free manner and to enable the scheduler to determine which nodes should transmit concurrently. Current approaches to performing scheduling with a TDMA MAC layer [12, 5] assume that data all must be routed to the base station and are designed to minimise the total amount of time required. In this work, the goal is to route data from a subset of sensors subject to a time limit[2].

The key contributions of this paper are two heuristic approaches to addressing the complex optimisation problem of

[1] Sensor selection, in this case, is concerned with sampling; all nodes remain active for routing.

[2] We aim to maximise the utility of the data transferred, rather than minimize the energy consumption of sensor nodes.

combining sensor selection, routing, and scheduling. These two approaches are compared to a baseline approach where routing and scheduling are performed independently of sensor selection and a load shedding mechanism is used to reactively deal with high network loads. With the first approach, routing and scheduling are performed first and sensor selection is performed afterwards given the routing and scheduling decisions. The second approach performs these three processes iteratively. The viability of the this work is evaluated in simulation through the use of a real world data set, which consists of a 54 node indoor deployment.

2. BACKGROUND

In this section, we provide an overview of previous work on sensor selection, routing, and scheduling. We point out that whereas there have been several efforts to jointly optimize routing and scheduling decisions, there is currently no work that exploits the interplay between all three processes.

Sensor Selection: Determining informative locations to activate sensors has been the subject of a considerable amount of research within wireless sensor networks. In [1], the problem is addressed in an optimisation framework based on utility functions. A standard approach to this problem is to model sensor readings as random variables for which the mean, variance, and covariance can be determined. Using these statistics, inferences can be made for locations where there are no sensors active from other correlated sensor readings.

In [10, 7], a greedy algorithm is used to optimise a submodular objective function to determine which set of locations to select. Submodular objective functions model diminishing returns. The idea is that the more sensors that are active, the less the utility increases by activating additional sensors. Formally, $f\colon 2^S \to \mathbb{R}$ is submodular if for $A \subset B \subset S$ and $x \in S$, $f_i(A \cup \{x\}) - f_i(A) \geq f_i(B \cup \{x\}) - f_i(B)$. In this work, the mutual information of a Gaussian process (GP) model of the chosen set of sensors and the set of locations where predictions are to be made is used as the selection criteria. As noted in [10, 7], the mutual information is a submodular function and represents a better approach than using the entropy, or the locations where the uncertainty is greatest, in that when using the entropy, locations tend to be chosen around the edge of the area of interest or query region. Given that sensor readings provide information about the local area, information beyond the area of interest is wasted.

Scheduling and Routing: TDMA MAC layers often perform better than contention based approaches in situations where the network is saturated and throughput must be maximised [9]. With contention based approaches, typically routing is performed first and then scheduling is performed implicitly given the routing decisions through the use of collision avoidance or collision detection schemes. One of the drawbacks to using TDMA MAC layers, however, is that they require time synchronisation. There are a number of approaches to addressing this issue, such as through the use of the Glossy flooding architecture [6].

In [12], various approaches to modelling interference with a TDMA MAC layer are considered. In general, the signal to interference plus noise model will perform the best, however, the scheduling algorithm for this approach is intractable. An alternative approach is to use the protocol model of interference presented in [8]. With the protocol model, if a transmitter node, T_1, is sending a message to a receiver node, R_1, and another transmitter node, T_2, is concurrently transmitting to a different receiver node, R_2, the receiver node, R_1, will not successfully receive the message if it is within a constant factor of T_2's transmission range. Similarly, receiver node, R_2, will not successfully receive the message if it is with a constant factor of T_1's transmission range. When using the protocol model, a greedy scheduling algorithm can be used that takes as input a demand vector that represents the number of packets to be scheduled on each link [12].

In sensor networks, scheduling cannot be performed optimally in terms of maximising the number of concurrent edge transmissions without first making decisions in relation to routing to determine the edge workloads and to identify scheduling conflicts due to the hidden terminal problem. Conversely, routing cannot be performed optimally in terms of throughput, for instance through the use of a maximum flow algorithm, without first making decisions in relation to scheduling to determine the edge capacities. In [12], the scheduling algorithm takes an edge demand vector as input, thus, it is assumed that routing has been performed first, then scheduling. Alternative approaches perform scheduling first and then routing given the scheduling decisions [15] or, indeed, jointly perform scheduling and routing [4].

There are several approaches to performing routing with TDMA MAC layers [9]. In the remainder of this paper, we adopt a similar approach to [12]. That is, we assume that routing is performed first and scheduling subsequently.

To our knowledge, there is currently no existing work that takes into account heuristics for routing and scheduling to perform sensor selection subject to bandwidth constraints. Whilst there are topology control and clustering algorithms [14] that choose subsets of nodes to be active, such approaches do not incorporate interference models or consider the amount of time required to realise a given set of communication requests in optimising over transmission scheduling decisions.

3. COMBINING ROUTING, SCHEDULING, AND SELECTION

With current approaches to performing scheduling with a TDMA MAC layer, the goal is to minimise the total amount of time required to send all data to the base station [5, 12]. In this paper, the goal is to choose an informative subset of sensors such that the amount of time required to collect data from the selected set is not greater than a specified epoch value or the TDMA frame length. In order for this to be achieved, sensor selection is performed subject to bandwidth constraints related to routing and scheduling. We present two approaches to combining the processes of routing, scheduling, and sensor selection and compare them to a baseline approach that represents a straightforward combination of current algorithms for these three processes. With the baseline approach, which we refer to as Independent Routing, Scheduling, and Selection (IND_RS_SEL), an ordering is placed on the sensors, in terms of their incremental utility, independently of the routing and scheduling decisions. This ordering is used by nodes when load shedding to determine which packets get sent to the base station. With the second approach, which is referred to as Integrated Routing, Scheduling, and Selection (INTG_RS_SEL), routing is

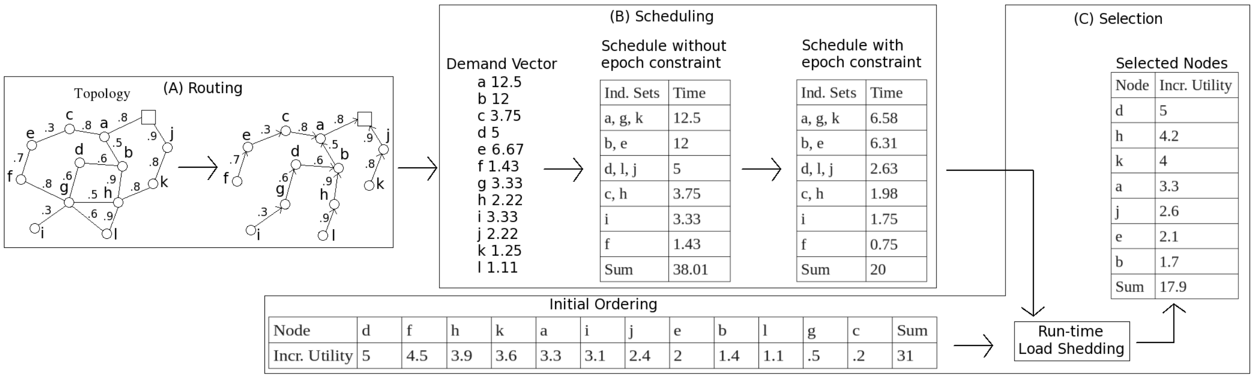

Figure 1: Independent Routing, Scheduling, and Selection

performed first, then scheduling, and subsequently sensor selection given the routing and scheduling decisions. The third approach, referred to as Iterative Routing, Scheduling, and Selection (ITR_RS_SEL), iteratively performs routing, scheduling, and sensor selection in an incremental manner. With the exception of the load shedding mechanism in IND_RS_SEL, all routing, scheduling, and sensor selection decisions are made in a centralised manner at the base station[3].

3.1 Independent Routing, Scheduling, and Selection (IND_RS_SEL)

Routing and Scheduling: With IND_RS_SEL (see Figure 1 (A)), routing is initially performed using a standard algorithm for constructing a minimum hop tree given the connectivity graph. Initially, the base station is added to the routing tree. Subsequently, the nodes that are fewest hops from the root, but have not yet been connected are added. When adding a node, if the node is in communication range with a number of other nodes, the edge that connects to the node that is fewest hops to the base station is used. If, at this stage, there are still a number of edges to choose from, the edge with the highest link quality is chosen. For example, in Figure 1 (A), when adding Node g, the link to Node d is chosen rather than Node h in that it has a higher link quality (the values for the link qualities are denoted on the edges of the connectivity graph). The link quality values represent the ratio of successful packet transmissions to total packet transmissions. In order to determine the link quality values, there is an initial profiling phase, whereby nodes transmit data to their neighbours and record the packet success rate[4].

Once the routing process has completed, scheduling is performed, assuming data from all sensors is to be sent to the base station, using a centralised algorithm similar to the greedy approach discussed in [12]. With this approach, the overall time period for message transmissions is divided into a number of transmission intervals of different length. In each interval, a feasible set of links transmit concurrently in an interference free manner. That is, all links that are transmitting in the same interval represent an independent set within the conflict graph constructed using the interference model. The scheduler orders links in terms of their workload[5]. Subsequently, links are greedily added to the first available interval in which they are not in conflict with the currently added links. In this way, links that have a high workload and are not in conflict with each other, in terms of timing, will often be scheduled in the same interval. One difference from this work and that of [12] is that edges are ordered in terms of the amount of time they require to transmit their workload rather than by the number of messages they have to transmit. The reason for this is that we do not make the assumption that link qualities are uniform; transmitting several messages over a high quality link could require less time than transmitting fewer messages over a poorer quality link. For example, in Figure 1 (A), the edge from Node e has 2 messages to transmit and a link quality of 0.3 and, thus, requires 6.67 time units, whereas the edge from Node c has 3 messages to transmit and a link quality of 0.8 requiring 3.75 time units[6]. Figure 1 (B), illustrates the demand vector constructed for the routing tree in Figure 1 (A), assuming all data must be sent to the base station. Using this demand vector, the edges are greedily assigned to different independent sets. For instance, when the edge from Node a is initially added, the next highest edge that is not in direct interference range, or in conflict with it due to the hidden terminal problem, is the edge from Node g. Subsequently, only the edge from Node k can be added to the same independent set. The amount of time assigned to this independent set is then set to 12.5, which is the maximum transmission time of the edges from nodes a, g, and k.

One of the novelties of this work is that, in contrast to other TDMA-based approaches [5, 12], scheduling is performed subject to a time limit. In order to reduce the scheduling time, the ratio between the total amount of time to collect data from the network with the current workload

[3]In this work, the cost of broadcasting the scheduling and selection decisions is considered negligible in the context of long running queries.

[4]The profiling phase is also used to obtain data to train the GP covariance function [13].

[5]It should be noted that, in the examples discussed in this paper, the traffic has no dependencies. For example, the traffic demand from Node g in Figure 1 already exists without the need to wait for the traffic from Node i. That is, at least as many epochs or TDMA frames as the longest path from the base station to a leaf node must go by before the traffic demands settle to the values given Figure 1 B.

[6]It should be noted these values are approximate; data rate chosen is "quantised" to fit the link quality.

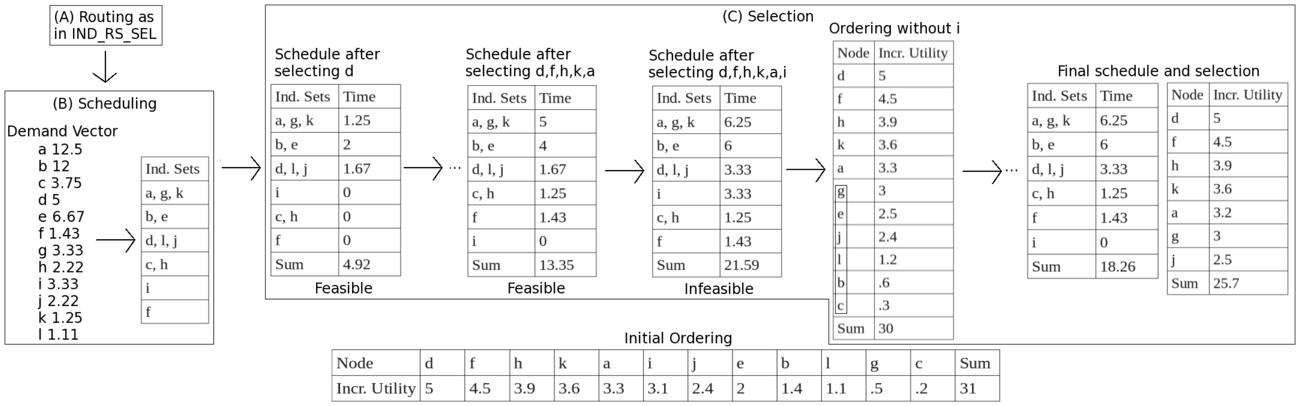

Figure 2: The schedule after selecting the first 5 sensors is feasible. At this point it is not possible to select Node i, which is the node with the next highest utility. Subsequently, the ordering of nodes changes given Node i cannot be selected.

and the epoch is determined. Subsequently, the amount of time assigned to each independent set is divided by this ratio. For example, in Figure 1 (B), the total amount of time required to transmit all messages is 38.01, whereas the epoch is 20, so the ratio is (38.01/20) 1.9005. The time intervals of each of the independent sets are divided by this value to create a schedule within the epoch. For instance, the amount of time assigned to the first independent set is (12.5/1.9005) 6.58. Once this process has completed, the schedule is broadcast to all nodes in the network along with a utility ordering of sensors. In cases where edges, or their ancestors in the routing tree, do not have enough time to transmit all of their messages, the utility ordering is used within a load shedding mechanism at run-time to determine which packets are transmitted. The remainder of this section discusses how this utility ordering is determined.

Sensor Selection: Independently of routing and scheduling, the centralised greedy algorithm presented in [10, 7] is used to place an ordering on the sensors in terms of their incremental utility using the mutual information of a Gaussian Process (GP) model. As noted in Section 1, GPs can be used to make predictions at locations where no nodes are present. GPs, when used for regression, represent an approach to non-parametric supervised learning that, in contrast to other regression techniques, can be used to determine the uncertainty of predictions made at arbitrary points [13]. This uncertainty takes into consideration the density of the observation nodes along with the observation noise. When performing sensor selection with GPs, all locations of interest are considered, not just locations where there are inactive sensors. That is, the selection process is optimised with regard to all locations where inferences are to be made. If V is the set of locations of interest and $A \subset V$ is the set of locations in V where there are nodes present, we wish to choose a set of active sensor locations $B \subseteq A$ that maximises the mutual information between the locations B and $V \backslash B$. Given a set of locations $V \backslash B$ where predictions are to be made, and a set of locations B of active sensors, the mutual information is defined as $I(X_B; X_{V\backslash B}) = H(X_B) + H(X_{V\backslash B}) - H(X_V)$.

The random variables X_B, $X_{V\backslash B}$, and X_V are constructed using a GP covariance function [13] for the respective locations. The function H represents the entropy of a n-variate Gaussian, which is defined as $\ln\sqrt{(2\pi e)^n|\Sigma|}$, where Σ is the covariance matrix of the Gaussian.

When using the greedy algorithm to place an initial ordering on sensors, the incremental utility for adding a node x to the previously chosen set of sensors C is determined by $I(X_{C\cup x}; X_{V\backslash C\cup x}) - I(X_C,\ X_{V\backslash C})$. For example, in Figure 1 C, the initial incremental utility of selecting Node h given that d and f have been previously selected is 3.9. As can be seen from Figure 1 C, the utility of some of the selected nodes differs from the initial utility. The reason for this is that when a node cannot be selected, the utility of the subsequent nodes changes. So, for instance, the utility of Node h increases from 3.9 to 4.2 given that Node f cannot be selected. Not considering these changes in the utility values is one of the disadvantages of the load shedding approach and this is addressed in INTG_RS_SEL and ITR_RS_SEL.

3.2 Integrated Routing, Scheduling, and Sensor Selection (INTG_RS_SEL)

Routing and Scheduling: With the second approach (see Figure 2), routing is initially performed using the same algorithm as IND_RS_SEL. Subsequently, scheduling is performed and edges are assigned to independent sets. In this case, however, the time associated with the independent sets is discarded (see Figure 2 (B)). With this approach, time is dynamically allocated to independent sets as sensors become selected, as discussed below.

Sensor Selection: When performing sensor selection, sensors are added in descending order of their incremental utility; again using the mutual information selection criteria [10, 7]. When a sensor is being considered for selection, the time allocated to each edge along its path to the base station is increased. For example, in Figure 2 (C), when the first sensor, Node d, is selected, the time allocated to the independent sets containing d, b, and a is increased by the amount of time required to transmit a message over these edges. If, when considering a sensor for selection, this leads to an infeasible schedule given the workload of the currently selected set, the sensor is discarded. So, for instance, in Figure 2, Node i cannot be selected, given the current workload of the network, in that it would increase the scheduling time

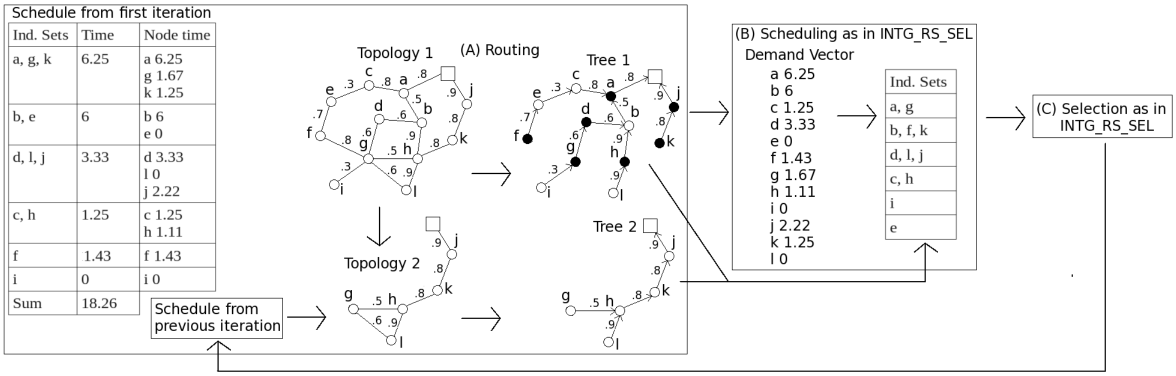

Figure 3: Sensors selected in the first iteration are routed through Tree 1 and the second iteration through Tree 2. The current workload and associated paths are the input to the scheduling step.

beyond the epoch. When a sensor cannot be selected, the next viable node with the highest utility with respect to the previously selected sensors will be chosen. That is, the utility values are computed in an incremental manner as sensors are selected/discarded. In this way, the ordering of sensors will be different than that of an initial ordering performed independently of routing and scheduling. For example, in Figure 2, given that Node i cannot be selected, Node g will have a higher utility than nodes j, e, and b.

3.3 Iterative Routing, Scheduling, and Sensor Selection (ITR_RS_SEL)

In the third approach (see Figure 3), the algorithm begins operating in the same way as INTG_RS_SEL. In this case, however, when the selection process has completed, routing and then scheduling are performed again given the currently selected set of sensors and associated workload.

Iterative Routing: In situations where an edge has a low workload with respect to other edges within the same independent set, there will be additional time available for data to be sent from its source node on a different link. There will also be remaining scheduling time available if the total amount of time required to schedule the current workload is less than the epoch. In order to take advantage of this additional time, a new graph is constructed that represents the nodes and edges that have additional capacity remaining. When creating the graph, nodes that do not have enough time to transmit another message, or do not have a path to the base station with additional capacity remaining, using the additional time from their independent set, some of the remaining scheduling time, or a combination of both are not included[7]. Once the reduced graph has been constructed, the minimum hop routing algorithm is used to construct a routing tree to the base station. In this respect, the algorithm operates in a somewhat similar manner to a maximum flow algorithm. That is, a node could potentially increase its throughput by sending data on an alternative route and avoiding congested links. For example, the second topology graph in Figure 3 (A) is constructed given the schedule from the first iteration. In this case, nodes g, l, j, and l have additional time remaining to transmit another message given the amount of time allocated to their independent sets[8]. Additionally, if the remaining scheduling time is assigned to Node h, it will have a feasible path to the base station, to send another message, via nodes k and j. Given this topology, Tree 2 is constructed using the minimum hop routing algorithm.

Scheduling: Once the new routing tree has been constructed, scheduling is performed again and edges are reassigned to independent sets. The input to the scheduling algorithm is a combination of the demand vector for the selected sensors, the currently selected paths, along with the previously selected paths[9]. For example, in Figure 3 (B), the demand vector represents the workload for the nodes selected in the first iteration (the black nodes in Tree 1).

Sensor selection: When the scheduling process has completed, sensor selection is performed again[10]. The selection algorithm uses the new tree when a sensor is being considered for selection. That is, when increasing the time assigned to each of the edges on the sensors path to the base station in selecting the sensor or determining if it is infeasible. For example, given Tree 2 in Figure 3 (A) and the current workload of the network, it will be possible to select Node l in the second iteration of the selection algorithm.

The algorithm continues to iteratively perform routing, scheduling, and sensor selection until no more additional sensors can be selected.

4. RESULTS

In this section, the temperature data from the Intel Berkeley Research Lab Data Set [2], which consists of a 54 node indoor deployment, is used to verify the viability of this work. The squared exponential Gaussian process covariance function [13] was used in performing regression for the locations of interest. The optimisation goal is to maximise the utility of the active nodes within the network and is evaluated using the Root Mean Squared Error (RMSE), which is a measure of the difference between the predicted values and

[7]A depth first search is used to determine the viable paths.

[8]It should be noted that Node e also has enough time to transmit another message, but it does not have a feasible path to the base station.

[9]The paths are used in constructing the conflict graph and all transmissions from the same node are scheduled within the same interval.

[10]It should be noted that the iterative algorithms for sensor selection and scheduling are the same as INTG_RS_SEL.

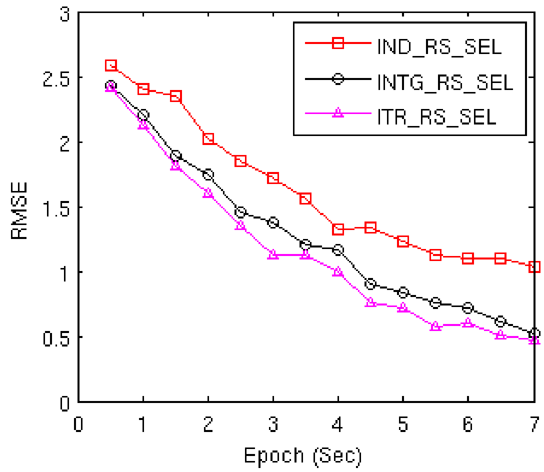

Figure 4: RMSE given an epoch

the ground truth. For an estimator $\hat{\theta}$ with respect to an estimated parameter θ, the RMSE is defined as $\sqrt{\mathbb{E}[(\hat{\theta}-\theta)^2]}$.

In Figure 4, the two approaches to performing sensor selection, routing, and scheduling, discussed in Section 3, are compared to the baseline. The x-axis represents different values for the epoch in seconds and the y-axis represents the RMSE[11]. As can been seen from Figure 4, ITR_RS_SEL, in general, performs best, but when the epoch is small the benefits are less pronounced. When the epoch increases, however, more additional capacity is made available when scheduling and routing are re-performed[12], enabling a greater number of additional sensors to be selected. So, for instance, when the epoch is 5, ITR_RS_SEL performs 42% better than IND_RS_SEL and 14% better than INTG_RS_SEL.

5. CONCLUSION

With current approaches to performing scheduling with a TDMA MAC layer, it is assumed that all data must be routed to the base station and the goal is to minimise the total amount of time required. This paper addresses the problem of choosing an informative subset of sensors such that the total amount of time to collect data from the network is not greater than a specified epoch value. A considerable amount of research has been conducted into determining informative locations to activate sensors using Gaussian processes. A limitation to current approaches, however, is that they do not take into consideration bandwidth constraints related to scheduling and routing or the amount of time required to realise the selected set and associated network topology. In this paper, two approaches were presented that address this issue. Generally speaking, iteratively performing routing, scheduling, and sensor selection performed best in that the interplay between these three processes was exploited to increase the amount of information produced by the network. The viability of this work was evaluated in simulation. Future work will incorporate the cost of profiling in the experiments for situations whereby the topology information and the utility of nodes change over time necessitating iterative profiling phases.

[11] In these experiments, there are two base stations in operation and it is assumed that 30 readings per second can be transmitted over a high quality link in the absence of interference.

[12] ITR_RS_SEL typically completes execution after 2 to 3 iterations.

6. ACKNOWLEDGMENTS

This work was supported by the Irish Research Council for Science, Engineering and Technology, the European Commission Marie Curie Actions under FP7, and the Science Foundation Ireland under Grant No. 07/CE/I1147. Thanks go the anonymous reviewers for the feedback on the paper.

7. REFERENCES

[1] F. Bian, D. Kempe, and R. Govindan. Utility based sensor selection. In *IPSN*, 2006.

[2] Peter Bodik, Carlos Guestrin, Wei Hong, Sam Madden, Mark Paskin, and Romain Thibaux. `http://db.csail.mit.edu/labdata/labdata.html`.

[3] A. Das and D. Kempe. Sensor selection for minimizing worst-case prediction error. In *IPSN*, 2008.

[4] P. Dutta, V. Mhatre, D. Panigrahi, and R. Rastogi. Joint routing and scheduling in multi-hop wireless networks with directional antennas. In *INFOCOM*, 2010.

[5] S.C. Ergen and P. Varaiya. TDMA scheduling algorithms for wireless sensor networks. *Wireless Networks*, 16(4):985–997, 2010.

[6] Federico Ferrari, Marco Zimmerling, Lothar Thiele, and Olga Saukh. Efficient network flooding and time synchronization with Glossy. In *IPSN*, 2011.

[7] C. Guestrin, A. Krause, and A.P. Singh. Near-optimal sensor placements in gaussian processes. In *Machine learning*, 2005.

[8] P. Gupta and PR Kumar. The capacity of wireless networks. *IEEE Transactions on information theory*, 46(2):388–404, 2000.

[9] O.D. Incel, A. Ghosh, and B. Krishnamachari. Scheduling Algorithms for Tree-Based Data Collection in Wireless Sensor Networks. *Theoretical Aspects of Distributed Computing in Sensor Network*, 2010.

[10] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research*, 9:235–284, 2008.

[11] Andreas Krause, Carlos Guestrin, Anupam Gupta, and Jon Kleinberg. Near-optimal sensor placements: maximizing information while minimizing communication cost. In *IPSN*, 2006.

[12] R. Maheshwari, S. Jain, and S.R. Das. A measurement study of interference modeling and scheduling in low-power wireless networks. In *SenSys*, 2008.

[13] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[14] S. Soro and W.B. Heinzelman. Cluster head election techniques for coverage preservation in wireless sensor networks. *Ad Hoc Networks*, 7(5):955–972, 2009.

[15] N. Trigoni, Y. Yao, A. Demers, J. Gehrke, and R. Rajaraman. Wave scheduling and routing in sensor networks. *ACM TOSN*, 2007.