SCAR: Context-aware Adaptive Routing in Delay Tolerant Mobile Sensor Networks

Cecilia Mascolo
Department of Computer Science
University College London
Gower Street, London, WC1E 6BT
United Kingdom
c.mascolo@cs.ucl.ac.uk

Mirco Musolesi
Department of Computer Science
University College London
Gower Street, London, WC1E 6BT
United Kingdom
m.musolesi@cs.ucl.ac.uk

ABSTRACT

Sensor devices are being embedded in all sorts of items including vehicles, furniture but also animal and human bodies through health monitors and tagging techniques. The collection of the information generated by these devices is a challenging task as the data results in enormous amounts and the sensors have scarce resources (especially in terms of energy for the forwarding of the data). Fortunately, the data is often delay tolerant and its delivery to the sinks is, in most cases, not time critical.

This paper tackles the problem of the delivery of mobile sensor data to sinks. We devise a Sensor Context-Aware Routing protocol (SCAR), which exploits movement and resource prediction techniques to smartly forward data towards the right direction at any point in time. In order to cope with the possibly frequent sensor faults, we also adopt a multi-path routing approach which increases the reliability.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*wireless communication, store and forward networks, distributed networks*; C.2.2 [**Computer-Communication Networks**]: Network Protocol—*routing protocols*

General Terms

Design, Algorithms

Keywords

Sensor networks, delay tolerant networks, mobile sinks, adaptive routing, buffer management, time series forecasting

1. INTRODUCTION

The proliferation of wireless sensors[1] and embedded computing is imposing new challenges to the development of data collection research and technologies. Sensor devices are now present in virtually all sorts of items, from vehicles and furniture to humans and animals. This generates network of wireless connected devices with topologies which could be very dynamic. The monitoring abilities of these devices range from pollution and temperature to health-care and mobility. The amounts of data generated by these applications are usually quite large, however, fortunately, the data is, in most cases, also *delay tolerant*, in the sense that it can wait in the network for quite a while before being collected.

The scenario we envisage in this paper is one where the mobile sensor nodes (e.g., animals, vehicles or humans) route data through each others in order to reach sink nodes, which can be either mobile or fixed. The fixed nodes are intended as nodes connected to a backbone network and therefore able to forward the data to the appropriate place when this is reached (Figure 1).

The challenges offered by this scenario are many and include the quantity of data to be shipped to the sinks, the potentially scarce communication power (i.e., energy and bandwidth) of the nodes, the possible communication and sensor hardware faults, the mobility and the scarce buffer size of the nodes.

Different techniques could be employed for mobile sensor data gathering. A basic strategy would be to only allow data delivery when sensors are in direct proximity of the sinks. This technique has very little communication overhead, given that messages are only sent directly from the sensor node generating messages to the sink. However, depending on how frequently sensor nodes meet the sinks, the delivery of the data might be very poor. This is particularly true if the sinks are very few and spread out.

More refined techniques would include epidemically inspired approaches [20], which would spread the data over the sensor network, so that eventually a sink could be reached. This approach has very good delivery ratio if buffers are sufficiently large, however the overhead in terms of communication and, therefore, energy is quite high.

In [21] an approach which is based on a probabilistic delivery approach for data messages is presented. The paper also discusses how the replication of the data over the sensor network can be constrained using a fault tolerance value associated to each data message. However, this approach still has quite a high overhead in terms of message spreading, due to the coarse grained delivery probability technique used for the choice of the nodes on which to replicate and

IWCMC'06, July 3–6, 2006, Vancouver, British Columbia, Canada.

the amount of replication involved by the approach. In sensor networks where energy and, therefore, communication overhead is an issue, the spreading of the message needs to be carefully controlled and traded off for the delivery ratio. This is even more true if the nodes have limited memory so that the buffer size is small and very few messages can be stored.

In this paper we present SCAR (Sensor Context-Aware Routing), a routing approach which uses prediction techniques over context of the sensor node (such as previously encountered neighbors, battery level, ..) to foresee which of the sensor neighbors are the best carriers for the data messages. We further adopt different classes of messages in order to achieve an intelligent buffer management.

Multiple carriers are chosen among the neighbor of the data source sensor, based on their history in terms of encounters, mobility and resources, however the number of replicated data around in the network is still considerably smaller than in any epidemic based approaches, in particular than in [21], where the effects of replication may lead to an epidemic-like spreading of the message.

Our prediction based techniques for choosing a carrier is based on Kalman Filters and has been exploited in [16], where we describe our Context-aware Adaptive Routing protocol for mobile ad hoc networks. SCAR has maintained the prediction based approach used in CAR but all the aspects related to the communication and the replication had to be redesigned. In particular SCAR has to suit the high data traffic of sensor networks. This is achieved by limiting the horizon in which deterministic information is kept to the neighbors of a sensor, and, given the fault rate of a sensor network, we have introduced an intelligent buffer management algorithm and multiple carriers for the message.

The structure of this paper is as follows: in Section 2 we present our approach, whereas in Section 3 we discuss its novelty, comparing it with the state of the art. Section 4 concludes the paper, outlining our current research directions.

2. OUR APPROACH

In this section we discuss the details of SCAR. Our approach can be summarized as follows: the mobile sensor nodes try to send their data to sink nodes, scattered over the field; each sensor node will try to deliver its data in bundles to a number of neighboring sensor nodes which seem to be the *best carriers* to reach a sink.

The decision process by which nodes select the best carriers is based on prediction of the future evolution of the system. Our solution relies on the analysis of the history of the movement pattern of the nodes and their colocation with the sinks and on the evaluation of the current available resources of the sensors. In particular, each node evaluates its change rate of connectivity, colocation with sinks, and battery level. The forecasted values of the context attributes describing the context are then combined to define a delivery probability $P(s_i)$ for each sensor s_i to deliver bundles to sinks.

While moving, the sensors will transfer their data to other sensors only if these have a higher probability to deliver the data to sinks (i.e., they are better carriers). The calculation of the delivery probability is *local* and it does not involve any distributed computation. Nodes only periodically exchange information about their current delivery probability and their available buffer space with the neighbors. We assume that each device of the system is actively involved in the storing-and-forwarding process: this is a reasonable assumption, since usually sensor networks are owned and deployed by a single organization.

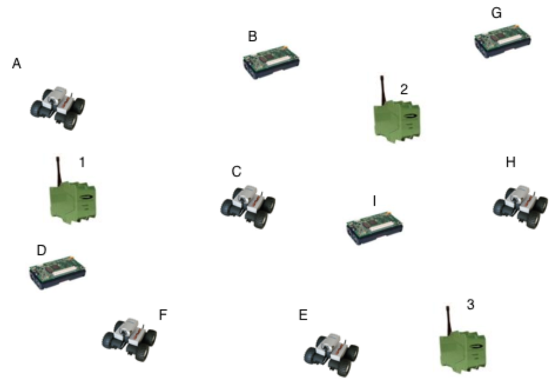

Figure 1: Sensor network composed of sensors (indicated with letters) and sinks (indicated with numbers). Sensors and sinks can be mobile or fixed.

2.1 Multi-carrier Selection

Each sensor that is the source of data tries to place bundles on a number of neighboring nodes which have the best chance to deliver them to a sink node.

Each node maintains an ordered list of the neighboring nodes (including itself) decreasingly ordered according to their delivery probabilities. Each node then replicates the bundle to the first R nodes ($R-1$ nodes if the node itself is in the first R positions of the list). The value of R is specified by the user and it can be considered as a priority level associated to the data retrieved by the sensor.

The replica sent to the node with the highest delivery probability is labeled as *master copy*. The other replicas are labeled as *backup copies*. These can be overwritten if buffers are full, whereas master copies are deleted only when sensors exchange the data with the sinks. In general, this distinction is used for an intelligent management of the buffer, that we will describe in Section 2.3. A unique identifier is also associated to each bundle. Replicas of the same bundle have the same identifier.

Each node keeps monitoring if there are neighbors with better probability of delivery than its own. If this is the case, the data bundles are shifted from one buffer to the other. This, however, implies that the data bundles are only replicated on a number of nodes in the first hop, while they are forwarded (i.e., deleted from one node and copied on another), later if the carriers, while roaming, find either a sink or a better carrier.

As we are in a sensor network, the high level of faults in the nodes implies that we need to allow for some more replication on the data (which is not something supported in the basic CAR protocol [16]). However, if the amount of data generated by the sensors is considerable, the approach of replication adopted by both epidemic-like protocols and in [21] incurs in heavy overheads. We replicate less but we try to control the replication in an intelligent way by predicting the future evolution of the system. In other words, data are replicated R times, with R that may be order(s) of magnitude less than the number of sensors composing the system.

As it will be explained in the following section, the delivery probability of the nodes also keeps into account the energy level of the nodes, so to avoid that some best carriers become strong attractors and run into low battery problems more quickly than others. In other words, we will show that as the battery level decreases, the probability of being selected decreases.

2.2 Choice of Best Carriers

In order to select the best carrier(s) for the data bundles, we use a mechanisms based on the estimation of the future behavior of each sensor node based on the history of its colocation with sinks, its changing rate of connectivity (i.e., its mobility), and its power level.

2.2.1 Forecasting techniques for probabilistic routing

Each node predicts, using time series forecasting techniques, the evolution of its *context* described by a set of attributes. In particular, we consider three indicators describing its colocation with the sinks, its change degree of connectivity and its battery level.

More specifically, a utility function is associated to each context indicator. Our aim is to maximize each attribute, in other words, to choose the node that presents the best tradeoff between the attributes representing the relevant aspects of the system for the optimization of the bundles delivery process. Analytically, considering k attributes with associated utility functions $U_1(s_i), ..., U_k(s_i)$, the problem can be reformulated as a multiple criteria decision problem [13] with k goals:

$$Maximize\{U(s_i)\} = f(U_1(s_i), ..., U_k(s_i)) \tag{1}$$

The combined goal function using the the so-called *Weights method* can be defined as

$$Maximize\{\sum_{j=1}^{n} w_j U_j(s_i)\} \tag{2}$$

where $w_1, w_2, ...w_k$ are *significance weights* reflecting the relative importance of each goal.

In our case, the solution is very simple, since it consists in the evaluation of the function $f(U_1, ..., U_k)$ using the values predicted for each node and in the selection of the node(s) i with the maximum such value.

The overall utility function $U(s_i)$ gives a measure of the probability that a node s_i is ability of delivering bundles to the sinks (i.e. of being co-located with them in the future). The delivery probability of each sensor will be equal to its composed utility function. More formally, the delivery probability of a sensor s_i is defined as

$$P(s_i) = U(s_i) \tag{3}$$

Two devices are co-located if they are in the same transmission range (i.e, one hop distance).Therefore, this utility function is computed considering its relative mobility (calculated by evaluating its change degree of connectivity history), its colocation with sinks, and its survivability (calculated by considering its battery level history)[1]. We associate a utility function to each of these indicators, respectively $U_{cdc}(s_i)$, $U_{coloc}(s_i)$ and $U_{battery}(s_i)$), and we compose these utility functions using a weighted sum as follows:

$$U(s_i) = w_{cdc}\widehat{U}_{cdc}(s_i) + w_{coloc}\widehat{U}_{coloc}(s_i) + w_{bat}\widehat{U}_{bat}(s_i) \tag{4}$$

where

- $\widehat{U}_{cdc}(s_i)$ measures the change degree of connectivity of the node i that we define as the number of connections and disconnections that a node has experienced over the last period $[t-1, t]$ seconds normalized by considering the nodes that have been in reach in this period. This parameter measures relative mobility and, consequently, the probability that a node will meet different nodes in a given period of time, that is the aspect that we are interested in. In fact, being in reach of a large number of different nodes increases the probability of meeting sensors with higher delivery probability or sinks. On the other hand, it may be possible to have a node that moves around but always together with the same nodes; in this case, the node is always co-located with the same devices. Even if its physical mobility is high, its topological mobility (i.e., considering its abstract connectivity graph) is equal to 0.

  More precisely, let $N_{i_{t-1}}$ the set of the neighbors of the node h at time t, the input value to the predictor at time t for $\widehat{U}_{cdc}(s_i)$ is equal to:

  $$U_{cdc}(s_i) = \frac{|N_{i_{t-1}} \cup N_{i_t}| - |N_{i_{t-1}} \cap N_{i_t}|}{|N_{i_{t-1}} \cup N_{i_t}|} \tag{5}$$

  where N_{i_t} is the number of nodes in reach of the sensor s_i at time t.

  Intuitively, this corresponds to the number of nodes that has been become in reach or out of reach in the time interval $[t-1, t]$ normalized by dividing it for the total number of nodes met in the same time interval.

- $\widehat{U}_{coloc}(s_i)$ summarizes the history of colocation of the sensor s_i with a sink. Therefore, the value of $\widehat{U}_{coloc}(s_i)$ is high if a node has been recently co-located with a sink.

- $\widehat{U}_{bat}(s_i)$ gives an estimation of the future battery level of the node. The value 1 corresponds to a full battery, whereas 0 corresponds to an empty one.

The relative importance of these utility functions is defined by using the weights w_{cdc}, w_{coloc} and w_{bat}. Weights are used to assign different importance to the different dimensions of the sensor context. For example, if the battery level is a critical dimension (that is often the case in wireless sensor networks, except for devices embedded in cars, planes or trains), a high value should be assigned to w_{bat}.

It is important to note that these utility functions represent an estimation of the future trend of these indicators calculated by exploiting time series analysis and forecasting techniques and not the current values of these utility functions. We use the symbol $\widehat{\ }$ to indicate the fact that these are predicted values and not current ones

[1]Even if we take into consideration only these three context indicators, our framework allows for the integration of other utility functions describing other aspects of the system that may be important to improve the performance of the storing-and-forwarding strategy.

The forecasted values are calculated by exploiting Kalman filter prediction techniques [12] that were originally developed in automatic control systems theory. These are essentially a method of discrete signal processing that provides optimal estimates of the current state of a dynamic system described by a *state vector*. The state is updated using periodic observations of the system, if available, using a set of *prediction recursive equations*.

In fact, it is possible to express this prediction problem in the form of a state space model. We have a time series of observed values that represent each context. From this it is possible to derive a prediction model based on an inner state that is represented by a set of vectors, and to add to this both trend and seasonal components [2].

It is worth noting that one of the main advantages of the Kalman filter is that it does not require the storage of the entire past history of the system, making it suitable for a sensor network setting in which computational and memory resources are very limited. Moreover, this technique is also very lightweight from a computational point of view, since the forecasting model only requires the update of the values representing the state using a system composed of linear equations (without any integration or differentiation required). In view of the fact that we use existing results, we do not present the mathematical aspects of the application of state space models theory and Kalman filter time series analysis in this paper; however, the interested reader can find these in [16].

2.2.2 Synchronization Issues

There are potential issues related to the fact that sensors can also be in idle mode. We need to consider two cases:

- *Nodes are data sources* Each sensor after selecting the nodes for the initial replication, if it cannot transfer the bundles immediately, it waits for a short period of time, so to increase the chances that to be active at the same time as the other node(s) to transfer its bundles. If these attempts are not successful, after the expiration of a pre-defined timeout, the sensor selects the subsequent node from their ordered delivery probability list.

- *Nodes are intermediate carriers* Each sensor transfers the data immediately after receiving the delivery probability from the other nodes, if this is higher. In this case, the probability that both nodes are active at the same time is very high.

However, in both cases, acknowledgment messages are used in order to enable re-transmissions. After a certain number of retransmissions, the other sensor is considered unreachable and, then, also in this case, the subsequent node from the ordered list is selected.

2.3 Buffer Management

2.3.1 Bundle Priorities

As discussed, a replica of a bundle can be a master o backup copy. When two nodes exchange their delivery probability, they also send the number of available slots in their buffer. We assume that the size of the buffer slots is fixed[2].

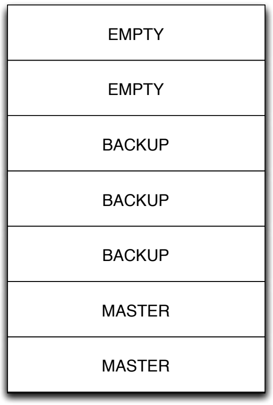

Figure 2: Buffer Management: the figure shows a buffer with a size equal to 7, with 2 master copies in it. In this case, the node will advertise 5 available slots.

A slot is considered available, if it does not contain a bundle or if its content can be overwritten (i.e., the slot contains a backup copy). For example, in Figure 2, a buffer composed of 7 slots is represented. The buffer contains three backup copies and two slots are empty. The sensor will then advertise 5 available slots.

Bundles are copied in the buffer of the other sensors firstly using the available empty slots and then overwriting the slots containing backup copies. Finally, we would like to discuss an interesting limit case. It may happen that a buffer is full and contains only master copies. In this case, the sensor will not accept any bundle from the other nodes [3]. However, if the node has been selected to carry so many master copies, this is due to the fact that its probability of being in reach of a sink is very high, so this situation will probably last for a very limited period of time (i.e., the sensor will get in reach of a sink soon and will transfer all the bundles and then free all the slots in its buffer).

2.3.2 Bundle Deletion Mechanisms

When a sensor meets a sink, the latter sends a hash table containing the identifiers of the bundles to the former. The sensor deletes all the bundles that have already been delivered from its buffer and then sends all the bundles that have not been delivered yet to the sink.

When two mobile sinks get in reach, they exchanges these hash tables. Then, each sinks updates its hash table adding the identifiers of the bundles delivered to the other sink and not already present in it. A timestamp is associated to each entry of the tables and the older ones are periodically removed.

2.4 Exchange of Context Information

Neighbors exchange these values of their delivery probability. Each node maintains an ordered *delivery probability*

[2]For simplicity, we also assume here that all the bundles have the same size. However, this mechanism can be easily extended in order to considered bundles of variable size, such as bundles that require two buffer slots and so on.

[3]In this case (i.e., when the number of slots is equal to 0), sensors will not advertise their deliver probability, in order to avoid a waste of energy, since this action will be completely useless.

list. Each entry of this table has the structure *(sensorId, deliveryProb, availableSlots)*; where *sensorId* is the sensor identifier, *deliveryProb* is its currently delivery probability and the last field is the number of available slots defined as discussed in Section 2.3.

Periodically, each sensor sends its delivery probability to their neighbors together with the number of the available slots. Each sensor sends its delivery probability after completing the neighbor sensing process that is performed after the transition from the idle to the active state.

2.5 Replication Process

As said before, each sensor keeps monitoring if neighbors with better probability of delivery than its own exist. This is done by examining the context information received by the other nodes. If there is a node in proximity with a higher delivery probability, the bundles are transferred to that node.

It is important to note that a bundle is copied from a sensor s_A to a sensor s_B if and only if the probability of s_A is a lot larger than the probability of s_B:

$$P(s_B) \gg P(s_A) \quad (6)$$

This is evaluated by setting an *exchange threshold* ζ. Therefore, the replication process between s_A and s_B happens if and only if

$$P(s_B) - P(s_A) > \zeta \quad (7)$$

This prevents replication actions that are not characterized by a good trade-off between delivery probability and energy consumption. Moreover, it avoids possible bundle thrashing, that may cause considerable waste of energy.

Finally, if the buffers of the other nodes do not have space for all the bundles to be transferred, priority is given to the master copies. If there is not enough space for all the master copies, these are selected for replication randomly[4]. The same happens for the backup copies.

2.6 Emergency Replication

An additional mechanism is introduced in order to cope with situations where nodes carrying master copies exhaust their battery. When the battery level is low (i.e., under a certain threshold), the master copies of the bundles are copied to the nodes in reach that have a sufficient number of free slots without considering the current values of their delivery probabilities.

In general, the fact that the battery level is taken into consideration in the calculation of the delivery probability should be sufficient to avoid these situations. However, it may happen that nodes with low battery level store master copies because of a particular combination of weights that gives a low relative importance to battery level and/or high values of the colocation and change degree of connectivity attributes. For this reason, this mechanism is introduced to increase the fault tolerance of the system and it will be used only in "emergency" situations.

[4]Alternatively, a priority may be associated to each bundle and used for this selection process. The number of initial replicas can also be used as priority. In this paper, we assume that all the data sources have the same importance (i.e., priority).

2.7 Predictability of the Sensor Network Scenario

Our system relies on predictions about the future values of context attributes. However, in some conditions predictions are not reliable, e.g., because the time series describing a particular context attribute is random or exhibit a behavior that cannot be forecasted with accuracy (i.e., within a given prediction error) using the model used. Therefore, it is important to assess the confidence level of context predictions, and modify forwarding decisions accordingly.

To assess the quality of context predictions it is possible to use the technique presented in [18], based on the analysis of the forecasting error [3]. A *predictability component* receives in input both the observed value (at time t) of a context attribute and the predicted value (computed at $t-1$). The analysis over time of the difference between these two values (called the *residual* value) enables to determine whether the prediction model (the Kalman filter in our case) has enough information to predict the next value of the time series with the required accuracy. In essence, this is true when the residuals are randomly distributed and their value is close to zero. The analysis of the predictability of the time series can be performed periodically in order to save resources. However, it is worth noting that this technique is rather lightweight in terms of use of resources.

When the predictability component determines that predictions are unreliable, we will use alternative protocols to carry the data, for example epidemic-style approaches.

3. DISCUSSION AND RELATED WORK

There have been a number of attempts of dealing with delay tolerant networks [8] overcoming the limitation of synchronous forwarding. In the area of mobile ad hoc networking, for instance, epidemic routing protocols [20] form the basis for much of the work in this field. Chen and Murphy refined the epidemic model, presenting the so-called Disconnected Transitive Communication paradigm [4]. Their approach argues for the use of utility functions, but it provides a general framework rather than a detailed instantiation, and so aspects related to the composition of calculated delivery probabilities are almost entirely missing.

In [19], Small and Haas describe a very interesting application of epidemic routing protocols to a problem of cost-effective data collection, using whales as message carriers. In [14], Lindgren et al. propose a probabilistic routing approach to enable asynchronous communication among intermittently connected clouds of nodes. Their approach is based on the fact that the exploited communication model is typically transitive and, for this reason, the probability of message delivery must be calculated accordingly. Zhao et al. in [22] discuss the so-called Message Ferrying approach for message delivery in mobile ad hoc networks. The authors propose a pro-active solution based on the exploitation of highly mobile nodes called ferries. These nodes move according to pre-defined routes, carrying messages between disconnected portions of the network.

In terms of sensor networks a lot of effort has been devolved into data forwarding in static sensor networks [15, 9, 6]. Some attempts have also been done in the direction of more dynamic sensor networks where mobile sinks are available such as [5, 10]. In ZebraNet [11] mobile sensors are deployed for tracking zebras in a hostile and wide environment. This is one of the closest work to ours together with [21].

However, with respect to these works, our data transmission overhead is lower (we do not have epidemic-like dissemination) and, thanks to the prediction techniques used to calculate the probabilities, the delivery of data is still reasonably high. In other words, we believe that our solution provide a better trade-off between the delivery ratio and the energy consumption (i.e., improved sensor survivability). We are still in the process of testing the algorithms, but we believe that this claim is supported by our previous simulation experiments and testing of the Kalman filter forecasting techniques with CAR [16].

4. CONCLUSIONS AND FUTURE WORK

In this paper we have described SCAR, a protocol for data forwarding on mobile sensor networks towards a number of fixed or mobile sinks. We plan to evaluate our approach first through simulation using our realistic mobility model founded on social theory [17]. We are also in the process of porting the algorithm on top of Telos Motes running Contiki [7] in order to evaluate SCAR on a real test bed.

Our research agenda is driven by the ambitious goal of integrating different devices to form a delay tolerant system that relies on different technologies and transmission media that is able to exploit both deterministic and probabilistic routing algorithms, like SCAR.

Acknowledgments: we would like to acknowledge the support of EPSRC Research Council through project CREAM.

5. REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

[2] P. J. Brockwell and R. A. Davis. *Introduction to Time Series and Forecasting*. Springer, 1996.

[3] C. Chatfield. *The Analysis of Time Series An Introduction*. Chapman and Hall, 2004.

[4] X. Chen and A. L. Murphy. Enabling disconnected transitive communication in mobile ad hoc networks. In *Proceedings of the Workshop on Principles of Mobile Computing (POMC'01)*, pages 21–23, August 2001.

[5] C. Curino, M. Giani, M. Giorgetta, A. Giusti, A. L. Murphy, and G. P. Picco. TinyLime: Bridging Mobile and Sensor Networks through Middleware. In *Proceedings of the 3^{rd} IEEE International Conference on Pervasive Computing and Communications (PerCom 2005)*, pages 61–72, Kauai Island (Hawaii, USA), Mar. 2005. IEEE Computer Society.

[6] A. Demers, J. Gehrke, R. Rajaraman, N. Trigoni, and Y. Yao. Energy-Efficient Data Management for Sensor Networks. In *IEEE Upstate NY Workshop on Sensor Networks*, 2003.

[7] A. Dunkels, B. Gronvall, and V. Thiemo. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the First IEEE Workshop on Embedded Networked Sensors*, Tampa, FL, Nov. 2004.

[8] K. Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of SIGCOMM'03*, August 2004.

[9] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of ACM/IEEE MOBICOM'00*, pages 56–67, 2000.

[10] D. Jea, A. A. Somasundara, and M. B. Srivastava. Multiple Controlled Mobile Elements (Data Mules) for Data Collection in Sensor Networks. In *DCOSS'05*, pages 244–257, 2005.

[11] P. Juang, H. Oki, Y. Wang, M. Martonosoi, L.-S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet. In *APSLOS-X*, San Jose, October 2002.

[12] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME Journal of Basic Engineering*, March 1960.

[13] R. L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preference and Value Tradeoffs*. Wiley, 1976.

[14] A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. *Mobile Computing and Communications Review*, 7(3), July 2003.

[15] S. Madden, M. Franklin, J. Hellerstein, and W. Hong. TAG: a Tiny Aggregation Tree for ad-hoc sensor networks. In *OSDI*, 2002.

[16] M. Musolesi, S. Hailes, and C. Mascolo. Adaptive routing for intermittently connected mobile ad hoc networks. In *Proceedings of the 6th International Symposium on a World of Wireless, Mobile, and Multimedia Networks (WoWMoM 2005). Taormina, Italy.* IEEE press, June 2005.

[17] M. Musolesi and C. Mascolo. A community based mobility model for ad hoc network research. In *Proceedings of the 2nd ACM International Workshop on Multi-hop Ad Hoc Networks: from theory to reality (REALMAN 2006)*. ACM Press, May 2006.

[18] M. Musolesi and C. Mascolo. Evaluating context information predictability for autonomic communication. In *Proceedings of 2nd IEEE Workshop on Autonomic Communications and Computing (ACC'06). Co-located with 7th IEEE Int. Symp. WoWMoM'06*, Niagara Falls, NY, June 2006. IEEE Computer Society Press.

[19] T. Small and Z. J. Haas. The shared wireless infostation model- a new ad hoc networking paradigm (or where is a whale, there is a way). In *Proceedings of MobiHoc'03*, June 2003.

[20] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-2000-06, Department of Computer Science, Duke University, 2000.

[21] Y. Wang and H. Wu. DFT-MSN: The Delay/Fault-Tolerant Mobile Sensor Network for Pervasive Information Gathering. In *Proceedings of INFOCOM'06*, Barcelona, Spain, Apr. 06.

[22] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *Proceedings of MobiHoc'04*, May 2004.