Using buddies to live longer in a boring world[*]

Samir Goel[1], Andrea Passarella[2], and Tomasz Imielinski[1]
[1]Department of Computer Science, Rutgers University, USA
{gsamir,imielins}@cs.rutgers.edu
[2]University of Cambridge, The Computer Laboratory, UK
andrea.passarella@cl.cam.ac.uk

Abstract

In a sensor network monitoring natural environment, the readings of sensor nodes show high temporal and spatial correlation. This stems from the fact that most characteristics of natural environment (chemical concentration, temperature, humidity, pollution, etc) do not change abruptly in space and time. From a sensor's perspective then, the environmental phenomena are quite predictable, and hence, "boring". In this paper, we introduce the novel idea of buddies to exploit temporal correlation. We propose a simple Buddy protocol to implement this idea. We analyze the performance of the protocol and identify factors that influence it. We discuss research challenges that would pave the way for more efficient protocols based on this idea.

1. Introduction

Numerous applications of sensor networks require sensor nodes, operating on AA batteries, to last for months if not years. Examples include habitat monitoring, battlefield reconnaissance, forest-fire detection and notification, chemical concentration sensing, etc. Today's batteries cannot support lifetime of months/years — a Berkeley Mote with all its components (sensors/radio) ON would last for 30 hrs on a single Energizer CR2450 lithium battery [5]. Going by the current technology trend, the battery capacity is expected to improve at a slow rate of 2-3% per year. Thus, the only way to extend lifetime of a sensor node is by using intelligent mechanisms that make judicious use of the energy resources.

It is a well-known fact that radio is the most energy consuming resource in a sensor node [5]. Energy cost of transmitting/receiving one byte is at least an order of magnitude higher than that for sensing and computation[1]. At a higher level, it has been shown [8] that in sensor networks *idle listening*, the case when radio is ON and idle, waiting for communication from neighbors, is the dominant factor in energy consumption. This suggests that a sensor node should not only transmit as few bytes as possible, it should also turn OFF its radio as much as possible.

In this paper, we propose to achieve this objective by exploiting the temporal correlation in the readings of sensor nodes. Our key idea is that two neighboring nodes can help reduce each other's energy consumption by entering into a collaborative *buddy* relationship. These *buddies* take turns in keeping their radio ON. At any point of time, the node that has its radio ON also act as a representative for its buddy, answering queries and participating in monitoring operation on its behalf. Buddies exchange information in order to make sure that the representative node's response on behalf of its buddy meets the accuracy constraint. In the ideal case very little information needs to be exchanged. In such a case, this collaborative arrangement would allow both nodes to cut down their energy consumption by a factor of 2. This basic idea can be extended to a group of N collaborating nodes. These nodes take turns to keep their radio ON. The representative node now responds on behalf of its $N - 1$ buddies. In the ideal case, this cuts down the energy consumption by a factor of N.

While saving energy, Buddy protocol also ensures that *three* key characteristics of the network are maintained: (a) any degradation in the quality of monitoring operation is bounded by network or application-specified constraints, (b) nodes meets the constraints on per-hop communication delay, and (c) turning OFF of node's radio does not adversely impact the reachability of the network.

These three characteristics differentiate this work from existing works on distributed clustering protocols (e.g., GAF [7]). Such protocols divide the nodes in the network into clusters, such that even if any one node from each clus-

---

[1]The cost of transmitting one byte is approximately the same as that for receiving it [1].

ter is ON, the reachability in the network is preserved. They save energy by turning OFF radios of nodes that are redundant with respect to connectivity. However such solutions do not consider the fact that while nodes may be redundant with respect to connectivity, their readings may not be redundant. These solutions save energy at the cost of *unbounded* reduction in quality of monitoring. Buddy protocol saves energy while at the same time meeting the constraint on quality of monitoring.

Temporal correlation may also be exploited by using local compression at the sensor node. Buddy protocol differs from such schemes in an important way. With compression a node necessarily needs to delay transmission of its readings. The larger the delay, the higher the potential for reduction in the size of transmission. Such delay may not be acceptable to the application. For example, in the setup of Great Duck Island Project [6], sensor nodes transmitted their readings once every 5 minutes. In order to compress readings from, say, 10 epochs, a sensor node would need to delay its response by 50 minutes!

In this paper, we describe the architecture and the protocol that implements the idea presented above. We analyze the performance of the protocol and identify factors that influence it.

In the next section, we describe our assumptions about the system, and briefly review PREMON paradigm [2] upon which the Buddy protocol is based. In Section 3, we provide details of the Buddy protocol. In the subsequent section, we analyze the performance of Buddy protocol and identify factors that influence it. We enumerate future research direction and conclude in Section 5.

2. Background

2.1. System Model

We consider a multi-hop sensor network that supports monitoring operation. A monitoring operation requires sensor nodes to report their readings every T_a units of time (e.g., "Report pollution level in the tunnel, every 5 minutes"). We assume that with each query or monitoring operation, the application specifies an error threshold, γ. This defines the maximum acceptable difference between the readings received by the application and the true readings of the sensor. We also assume that all nodes must guarantee a maximum per-hop communication delay of T_d.

2.2. PREdiction-based MONitoring paradigm

PREMON [2] exploits correlation in sensor readings in order to save energy. Consider a sensor node being queried (i.e., a node involved in monitoring activities). Instead of reporting readings periodically, a sensor node in PREMON generates a prediction of its readings, encodes it concisely as a *prediction-model*, and sends it to the monitoring entity (e.g., the sink). The sensor node also specifies the duration for which the prediction model is valid. During this lifetime, the sensor node transmits only those readings that differ from the predicted readings by more than a certain pre-specified *error threshold*, γ. Such readings are termed as *"violations"*. In the absence of any transmissions from the sensor node, the monitoring entity uses the sensor's prediction model to determine the readings of the sensor node. At the end of the lifetime of the prediction-model, the sensor generates a new prediction model and sends it to the monitoring entity.

Computing a good prediction model is an orthogonal problem. For example, a simple regression model could be used. Specifically, the network designer may pre-specify a set of basis functions, $F = \{f_1(t), f_2(t), ..., f_n(t)\}$. The prediction model, $M(t)$, may then be concisely expressed as a vector of coefficients, $W = \{w_1, w_2, ..., w_n\}$, such that $M(t) = w_1 f_1(t) + w_2 f_2(t) + ... + w_n f_n(t)$, where $t = t_1, t_1 + T_a, t_1 + 2T_a, ..., t_2$, and T_a is the reporting frequency requested by the application. $M(t)$ is defined only at these discrete instants of time; at all other points, its value is undefined. Computing prediction-model may involve techniques ranging from simple linear regression to more sophisticated ones [5].

Note that in PREMON a queried sensor node should keep its radio ON even during intervals when its readings are predictable. This allows it to respond to requests from other monitoring entities in the network. PREMON paradigm trades increased computation (for deriving prediction models) for savings in number of transmissions. Buddy protocol extends the PREMON approach by i) allowing models to be installed within the network instead of in the central monitoring entity, and ii) allowing nodes to turn their radio OFF after installing the prediction model, without affecting responsiveness and network recheability.

2.3. How predictable is data in real world?

To understand if a prediction-based approach could be really effective, we analyzed the dataset collected during the Great Duck Island Project [6]. We present the results for humidity and temperature readings from one representative sensor node. The samples were collected every 5 minutes for approximately 3 days.

We use linear regression for generating predictions. Specifically, we use K readings of the sensor to estimate the coefficients of the linear model, $M(t) = w_1 + w_2 \cdot t$, that result in best fit (minimum mean square error). We use the resulting model as a predictor for the next Δ readings. After Δ readings, the value of coefficients w_1 and w_2 of the
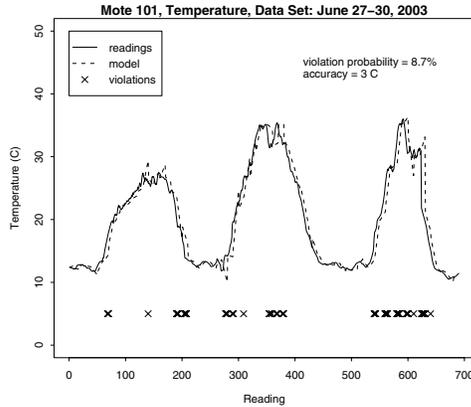
Figure 1. Actual versus predicted readings of temperature sensor

model are estimated once again, using the last K readings. Figure 1 shows the result of applying this technique to the readings of temperature sensor. The parameter values chosen were: $K = 5$, $\Delta = 10$, and error threshold $\gamma = 3$ units (for both sensors). The results show that this simple prediction model "correctly" predicted 91% of the readings of temperature sensor. The same mode was able to correctly predict 57% of the readings for humidity sensor. This examples illustrates that the Buddy-protocol idea can be really effective. Clearly, the number of correct predictions depend on the threshold value – the higher the threshold, the higher the number of correct predictions. The graph illustrating this relationship has been omitted due to space constraints. Interested readers are referred to [3].

3. Buddy Protocol

3.1. Basic Idea

In order to exploit correlation in its readings, each node attempts to establish buddy relationship with its neighbors. This results in formation of a number of buddy-groups in the network. Each buddy-group has one representative node responsible for answering queries and participating in monitoring operation on behalf of all the other nodes in the group. This allows all other nodes to turn their radio OFF, resulting in significant energy savings. The members of the buddy-group collaborate to rotate the responsibility of being the representative, so as to spread the consumption of energy uniformly, over the group members. Two key characteristics must be preserved in order to make this energy-saving mechanism transparent to the application, i.e., i) quality of monitoring, and ii) reachability in the network. As explained above, the PREMON approach is able to pre-

serve the quality of monitoring by the joint use of the prediction model and of the violations. On the other hand, by including a periodic polling scheme in the cluster operations (as described in the next section) the Buddy protocol is able to guarantee that each node in the network is reachable within the specified maximum delay constraints.

To make things simple, we used standard clustering mechanisms (e.g. GAF [7]), to build the buddy groups. Each cluster is a buddy group, and the cluster-head is the representative node for the group. Ways to tune clustering algorithms to the specific needs of Buddy protocol is a subject of ongoing research work.

3.2. Operational details

Within a cluster, we assume that nodes follow a style of operation similar to PSM (power-saving mode) mode of 802.11. The cluster-head performs the role of access-point, buffering data for all the nodes in the cluster. The cluster-head indicates the presence of buffered data by using MAC-level beacons that are sent out periodically. In order for nodes to meet the per-hop communication delay requirements, the nodes wake up every T_d units of time to receive beacon from the cluster-head. We assume that T_d is an integer multiple of the beacon-period.

Within each cluster, every node (except cluster-head) estimates the cost of operating in PREMON mode and DEFAULT mode. In the DEFAULT mode, a queried node periodically sends readings to the clusterhead, while a non-queried nodes just wakes up to get periodic cluster-head beacons. Every T_Δ units of time ($T_\Delta = \Delta \cdot T_a$), the node decides the mode of operation that is going to be more energy efficient, and switches to that mode. It stays in the chosen mode for T_Δ units of time. If a node decides to use PREMON mode, it sends prediction-model of its readings to the cluster-head. The node may then turn OFF its radio, turning it ON only for sending violations and get beacons. If a node decides to operate in DEFAULT mode, it needs not send any message to the cluster head. The cluster-head knows the mode of operation of all the nodes in the cluster. It generates readings (and replies to new queries) on behalf of the nodes operating in PREMON mode. It passes queries/monitoring operation request to any node operating in DEFAULT mode. These nodes report their readings directly to the querying/monitoring entity. Note that we assume that all messages are sent reliably (i.e., we assume an ARQ scheme with infinite re-transmissions).

In the next section, we describe how a node may estimate the cost of operating in DEFAULT and PREMON mode, and provide a preliminary analysis about the Buddy protocol effectiveness.

4. Performance Analysis

In this section we derive expression for the cost of operation in the two modes. We identify factors that influence these costs and derive "feasible regions" in the parameter space where using PREMON mode is more cost effective than DEFAULT mode. We express the energy cost in terms of the duration for which the radio was ON. This is a good approximation for two reasons: (a) Energy consumed by radio is an order of magnitude higher than that of computation, and (b) The cost of transmitting and receiving data are roughly the same [1].

4.1. Estimating costs in DEFAULT and PREMON modes

When a node operating in DEFAULT mode is not being queried, it only needs to listen to beacons from the cluster-head every T_d units of time. The costs of doing this over the time interval T_Δ is given by $T_D^{NM} = \frac{T_\Delta}{T_d} \cdot T_{beac}$, where, T_{beac} is the duration of a beacon packet. A node that is being queried needs to also send its readings every T_a units of time. The cost of doing this over the time interval T_Δ is given by $T_D^M = \frac{T_\Delta}{T_d} \cdot T_{beac} + \frac{T_\Delta}{T_a} \cdot T(d)$, where $T(d)$ represents the average time required to send a packet, of size d bits, reliably to the cluster-head. It is a function of channel bit error rate, BER, and packet size. Note that all parameters required to compute the cost of operating in DEFAULT mode, for T_Δ units of time, are available locally at the sensor node.

In PREMON mode, the node listens to the beacon from the the cluster-head every T_d units of time as in DEFAULT mode. Furthermore, possible violations have to be sent, as well. Let p_v represent the probability of occurrence of this event. Then, during time interval T_Δ, on an average, the sensor node needs to transmit $p_v \cdot \Delta$ number of violations to the cluster-head. The total cost of operating in PREMON mode is given by $T_B = \{T(m) + T_{beac} + p_v \cdot \Delta \cdot (T_{beac} + T(d))\} + \{(\frac{T_\Delta}{T_d} - \Delta) \cdot T_{beac}\}$.

Note that the first part of the above equation represents the cost of sending a prediction-model (of size m bits) and sending violations. The second part represents the cost of listening to the beacons from the cluster-head. Note that every T_a units of time, the node can skip listening to the beacon because for this epoch the cluster-head knows the reading of the sensor node and it can reply to possible new queries on its behalf. Also note that, unlike in DEFAULT mode, the energy cost in PREMON mode is the same irrespective of whether the sensor node is being queried or not.

For computing the energy cost of PREMON mode all parameters except p_v are known locally at the sensor node. Sensor node keeps a running estimate of p_v, by locally gen-
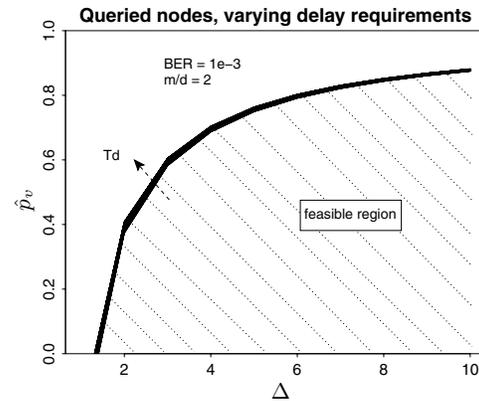

Figure 2. Results fot node being queried

erating a prediction-model and comparing its actual readings against the predicted ones. It does this irrespective of its mode of operation.

4.2. Feasible Region

We analyze the energy costs in the two modes in order to identify "feasible regions" in the parameter space where PREMON mode is more energy efficient than the DEFAULT mode. Let I_{ps} be the ratio of energy cost in PREMON mode to that in DEFAULT mode. Let us define $\hat{p}_v$ as the value of p_v for which $I_{ps} = 1$. Thus $\hat{p}_v$ defines a range of $p_v = [0, \hat{p}_v)$ for which PREMON mode is more energy efficient (i.e., the boundary of the feasible region).

Figure 2 shows result for the case when node is being queried. It shows that the feasible region in the parameter space for p_v and Δ is quite large. Intuitively, $\hat{p}_v$ increases with increase in Δ — a longer model validity allows to amortise the cost of sending a model over a longer time interval. The flattening out of graph for larger values of Δ has important consequences — a prediction model need *not* predict correctly too far into the future to be effective. It argues for simple prediction-models with good prediction accuracy in the short-term. Note that the feasible region is largely insensitive to the value of T_d. This is because of the small cost of listening to the beacons against that of transmitting readings.

For the case when a node is not being monitored, the cost in PREMON mode remains the same, whereas the cost in DEFAULT mode is substantially smaller — the only operation involved is listening to beacons from the cluster-head every T_d units of time (please refer to [4] for more details). Thus, in general, when a node is not being queried, it is more energy efficient to operate in DEFAULT mode.
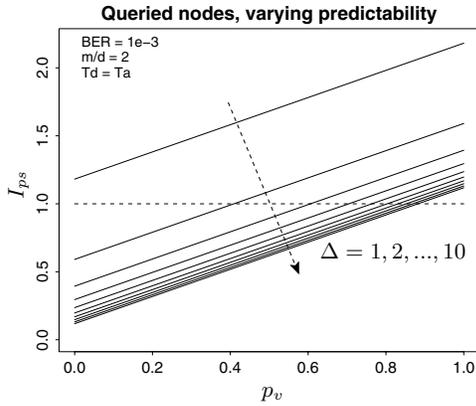
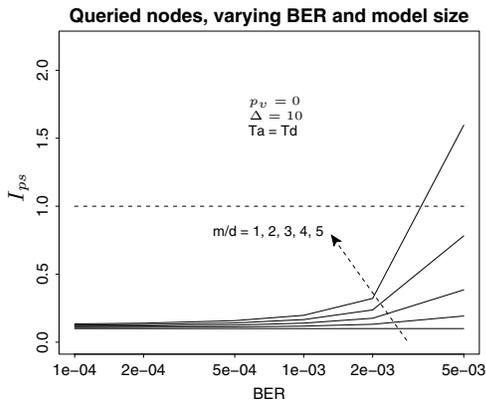**Figure 3.** I_{ps} **as function of the model accuracy**

Figure 4. I_{ps} **as a function of the** BER **and** $\frac{m}{d}$

4.3. Energy savings in PREMON mode

We now focus on nodes that are being queried, and analyze the impact of various parameters on the cost of PREMON mode. We eliminate the effect of polling on the cost by assuming that $T_d = T_a$.

Figure 3 shows I_{ps} as a function of the accuracy of prediction model (p_v). Linear increase in I_{ps} indicates that the energy consumption in PREMON mode increases linearly with p_v. As expected, increase in Δ reduces the energy consumption in PREMON mode. Figure 3 shows that, in the best case, for the parameters chosen, energy cost in PREMON mode is 10% of that in DEFAULT mode — a reduction by a factor of 10!

Figure 4 shows the effect of the BER and of the size of the encoded model (in bytes) on the Buddy-protocol effectiveness. We capture the effect of relative size by the ratio $\frac{m}{d}$ — the ratio of the size of packet containing prediction-model to that containing the reading. The results in the figure were generated for $p_v = 0$ — perfectly predictable sensor readings. As the ratio $\frac{m}{d}$ increases, the energy required to deliver the prediction-model increases. As BER increases the difference in energy cost for delivering the two types of packets becomes more pronounced. Interestingly, the results show that even for perfectly predictable sensor readings, the PREMON mode may cost more energy than DEFAULT mode for certain range of values of BER and $\frac{m}{d}$. As p_v increases, although the shape of curves remain the same, the set of curves move up (we have omitted this graph for lack of space). Thus, it is important to strike a good balance between size of prediction-model and its accuracy, and to estimate BER before deciding on the mode of operation.

5. Conclusions and Future Works

In this paper, we presented a novel idea for exploiting temporal correlation. We showed that neighboring sensor nodes can help reduce each other's energy consumption by entering into collaborative buddy relationships. We presented a simple Buddy protocol that implements this idea. We analyzed its performance and identified factors that influence it.

This protocol shows one possible way in which the above idea can be implemented. Designing such a protocol presents many research challenges. For example, the ideal cluster head in Buddy protocol is a node to whom other nodes in the cluster have very good link quality. Devising a BER-based distributed voting mechanism for this purpose is a challenging research problem. Further energy gains can be achieved by extending the Buddy protocol to also exploit spatial correlation. Finally, Buddy protocol can be extended by defining super-groups – grouping cluster-heads and installing the models in a small subset of them, responsible for replying to queries. Interested readers are referred to [4] for more details.

References

[1] G. Anastasi, M. Conti, A. Falchi, E. Gregori, and A. Passarella. Performance measurements of mote sensor networks. In *Proceedings of the ACM/IEEE Symposium on Modeling, Analysis and Simulation of Wireless and Mobile System (MSWIM 2004)*, October 2004.

[2] S. Goel and T. Imieliński. Prediction-based monitoring in sensor networks: Taking lessons from mpeg. *ACM Computer Communication Review*, 31(5), October 2001.

[3] S. Goel, A. Passarella, and T. Imielinski. Using buddies to live longer in a boring world. In *Technical Report DCS-TR-558, Department of Computer Science, Rutgers University*, Sep 2004.

[4] S. Goel, A. Passarella, and T. Imielinski. Using buddies to live longer in a boring world. Technical Report DCS-TR-558, Rutgers University, September 2004.

[5] J. Hill, R. Szewczyk, A. Woo, S. Hollar, and K. P. D. Culler. System architecture directions for networked sensors. In *Proceedings of the 9th Int'l Conf. on ASPLOS*, November 2000.

[6] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler. Lessons from a sensor network expedition. In *Proceedings of the First European Workshop on Wireless Sensor Networks (EWSN '04)*, January 2004.

[7] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad-hoc routing. In *Proceedings of 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '01), Los Angeles*, July 2001.

[8] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proceedings of IEEE INFOCOM'02*, June 2002.