Poster Abstract: Pushing a Standard Wireless Sensor Network Stack for Ultra-low Data Rates

Usman Raza
University of Trento and
Bruno Kessler Foundation
Trento, Italy
raza@fbk.eu

Amy L. Murphy
Bruno Kessler Foundation
Trento, Italy
murphy@fbk.eu

Gian Pietro Picco
University of Trento
Trento, Italy
gianpietro.picco@unitn.it

ABSTRACT

Time series forecasting aims at improving energy efficiency in wireless sensor networks (WSNs) by reducing the amount of data traffic. One such technique has each node generate a model that predicts the sampled data. When the actual, sensed data deviates from the model, a new model is generated and transmitted to the sink. Reductions in application data traffic as high as two orders of magnitude can be achieved. However, our experience in applying such forecasting in a real world deployment shows that the *actual* lifetime improvement is significantly less due to networking overheads. The study reported here reveals that careful, coordinated network parameter tuning can leverage the reduced traffic of forecasting techniques to increase lifetime without compromising application performance.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Applications; Routing Protocols; E.m [**Data**]: Miscellaneous; C.4 [**Performance of Systems**]: Measurements

General Terms

Design, Experimentation, Measurement, Performance

Keywords

MAC, Routing, Optimization, Energy efficiency, Low power

1. INTRODUCTION

Time series forecasting [3,7] can be used to reduce the data rate in WSN applications that can function with only an approximation of the data sensed by the distributed nodes. In this technique, each node locally computes a model that predicts the data trend. This model is transmitted to the sink, which uses the collected models to approximate the data sensed by each node. As long as the "forecasts" of each model remain within well-defined error bounds of the actual sensed values, no communication is required. Otherwise, a new model is computed and sent to the sink.

SenSys'13, Nov 11-15 2013, Roma, Italy
ACM 978-1-4503-2027-6/13/11.
http://dx.doi.org/10.1145/2517351.2517423 .

Table 1: Data Reduction with DBP

WSN Application	Datasets	Data Reduction
Tunnel Lighting	Light	99.74%
Soil Ecology	Air Temperature	91.83%
	Soil Temperature	98.80%
Indoor Sensing	Humidity	99.50%
	Light	97.58%
	Temperature	99.60%

Our novel, linear modeling technique, Derivative Based Prediction (DBP), has demonstrated [6] up to 99% reductions in the amount of data generated in a sample network. DBP takes a small sequence of sensed data and constructs a line approximating the trend within that sequence. Future sensed data is compared to the data predicted by the linear model. If the sensed values are too far away from the line for too long, a new model is generated and sent to the sink.

To evaluate the effectiveness of the model, we applied DBP to six different data sets and their applications: *i)* a WSN in a road tunnel used to monitor and control the lighting [2], *ii)*, soil data from the Life Under your Feet Project [4] used by biologists to study micro-climates, *iii)* indoor humidity, light and temperature data from a testbed at the Intel Berkeley research lab [1], applied to building climate control. When applying reasonable application-dependent error tolerances, Table 1 shows that DBP achieves reductions in the transmitted data from 91 to 99%.

Nevertheless, when this extremely small amount of data is sent on top of a standard WSN network stack composed of CTP and BoX-MAC, the system lifetime improves *only* 3-times [6]. As shown in Figure 1, this relatively low improvement can be attributed to the large control overheads present in these protocols. For example, BoX-MAC controls the radio duty cycle, periodically waking up the radio to check for a possible intended reception. With shorter wakeup intervals, transmission times at each hop are reduced. However, as there is very little data, the receive checks are often unnecessary, resulting lost energy. Therefore, the size of the wakeup interval must be adapted

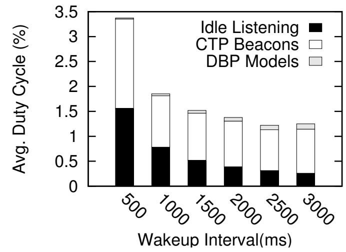
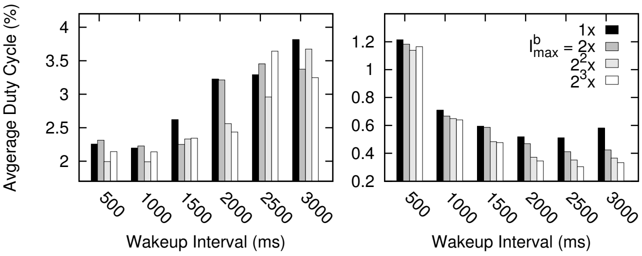


Figure 1: Energy Breakdown.



(a) Without DBP. (b) With DBP.

Figure 2: Effect of parameter tuning on average duty cycle. Note the different scales on the y-axis.

to balance this trade-off between idle listening and active transmission costs while staying within the latency tolerances of the application. At the routing layer, CTP incurs significant overhead due to the large number of broadcast beacons. Because broadcasts are, on average, twice as expensive as unicast transmissions for duty cycled radios, we must adapt the CTP beacon interval to lower the overhead while still maintaining reliable collection trees.

Additionally, broadcasts and the wakeup interval are interrelated. A recent study [5] reveals that reducing the number of broadcasts increases the optimal wakeup interval. Conversely, a long wakeup interval decreases idle listening to a much bigger extent than the accompanying increase in broadcast transmission costs (See Figure 1). Therefore, a large overall lifetime improvement is possible by jointly optimizing multiple layers of the network stack.

2. PRELIMINARY RESULTS

To evaluate the potential lifetime gains possible with DBP on a standard network stack composed of CTP and BoX-MAC, we ran a series of experiments with a 40-node indoor testbed. We use the road tunnel lighting application as the data reduction seen by DBP is representative of the applications we have studied. To realistically evaluate the application in a controlled setting, each testbed node replays the light data sensed in a real road tunnel in Trento, Italy during a 4-hour period. For network lifetime, we use the average radio duty cycle as a proxy for energy consumption as the radio is the most power hungry component of the system.

The two parameters we vary in our experiments are the BoX-MAC wake up interval and CTP's inter beacon interval. The latter is controlled by CTP's Trickle algorithm in which each node sends one beacon after a random time between $I^b/2$ and I^b. The Trickle algorithm initializes I^b to I^b_{min} (0.125s by default) and doubles it every I^b up to a maximum of I^b_{max} (500s by default). One option to lower the control overhead and save energy is to adapt the values of I^b_{min} and I^b_{max}. We choose to increase the value of I^b_{max} as long as CTP can build reliable routes with a high delivery ratio. Figure 2 plots the average network duty cycle versus the wakeup interval for 4 different I^b_{max} values 500, 1000, 2000, 4000 s (labeled 1x, 2x, 2^2x and 2^3x).

In Figure 2(a), nodes do not use DBP and send one data sample every 30s. The best wakeup interval, corresponding to the minimum duty cycle, is 1000ms irrespective of the value of I^b_{max}. This wakeup interval balances the trade-off between idle listening and active transmission costs. Above 1000ms, the data transmission costs increase considerably, increasing the duty cycle. In contrast, for the ultra-low traffic observed with DBP, the data transmission time is a small fraction of costs contributing to the radio duty cycle. Therefore, a long wakeup interval greatly reduces idle listening and thus the duty cycle in Figure 2(b).

Turning our attention to the modification of the broadcast beacon interval with I^b_{max}, we see that because beacons are a small percentage of the total traffic when DBP is not used, the average network duty cycle is inconsistently affected by increases in I^b_{max} due to collisions and other multi-path effects. Instead, with the lower data traffic, beacons dominate, and tuning the beacon interval offers significant, consistent benefits. Specifically, with DBP, the optimal wakeup interval is 2500 ms and the decrease in the number of beacons nearly cuts in half the average duty cycle (to 0.6). If we then compare this to the optimal value without DBP shown in of Figure 2 (wakeup interval = 1000 ms, I^b_{max}=1x), we see that this combination of data forecasting and parameter tuning can improve lifetime 8-fold. In our tunnel application, where the standard configuration offers a 2-year battery lifetime, this simple tuning increases lifetime beyond 15 years.

3. CONCLUSION

Our preliminary results indicate that joint parameter optimization has the potential to significantly increase the lifetime improvement achievable with time series forecasting. As the low data rates of our target scenario are characteristic of applications employing other data reduction techniques such as data compression, in-network aggregation, adaptive sampling and stochastic data modeling, these results are applicable in wide range of scenarios.

4. REFERENCES

[1] Intel Lab Data. `http://db.csail.mit.edu/labdata/labdata.html`.

[2] M. C. M. Corrà, L. D'Orazio, R. Doriguzzi, D. Facchin, S. Guna, G. P. Jesi, R. L. Cigno, L. Mottola, A. L. Murphy, M. Pescalli, G. P. Picco, D. Pregnolato, and C. Torghele. Is there light at the ends of the tunnel? Wireless sensor networks for adaptive lighting in road tunnels. In *Proc. of the Int. Conf. on Information Processing in Sensor Networks (IPSN)*, 2011.

[3] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *Proc. of the Int. Conf. on Very large data bases-Volume 30*, pages 588–599. VLDB Endowment, 2004.

[4] R. Musaloiu-E, A. Terzis, K. Szlavecz, A. Szalay, J. Cogan, and J. Gray. Life under your feet: A wireless soil ecology sensor network. In *Proc. of the Wkshp. on Embedded Networked Sensors (EmNets).*, 2006.

[5] D. Puccinelli, S. Giordano, M. Zuniga, and P. J. Marrón. Broadcast-free collection protocol. In *Proc. of the Conf. on Embedded Network Sensor Systems*, pages 29–42. ACM, 2012.

[6] U. Raza, A. Camerra, A. L. Murphy, T. Palpanas, and G. P. Picco. What does model-driven data acquisition really achieve in wireless sensor networks? In *Proc. of the Int. Conf. on Pervasive Computing and Communications (PerCom)*, pages 85–94. IEEE, 2012.

[7] D. Tulone and S. Madden. PAQ: Time series forecasting for approximate query answering in sensor networks. In *European Conf. on Wireless Sensor Networks*, pages 21–37. Springer, 2006.