

Testing Selective Transmission with Low Power Listening

Morten Tranberg Hansen¹, Rocío Arroyo-Valles², and Jesús Cid-Sueiro ^{*}

¹ Aarhus University,
Aabogade, 34, 8200 Aarhus, Denmark,
mth@cs.au.dk,

² Universidad Carlos III de Madrid,
Av. de la Universidad, 30, 28911 Leganés-Madrid, Spain
{marrval, jcid}@tsc.uc3m.es

Abstract. Selective transmission policies allow nodes in a sensor network to autonomously decide between transmitting or discarding packets depending on the importance of the information content and the energetic cost of communications. The potential benefits of these policies depend on the capability of nodes to estimate its current energy consumption patterns. As a case study, this paper tests the performance of a particular selective transmission algorithm over a simple network using a low power listening MAC protocol on real sensor node hardware.

Keywords: selective transmission, implementation, energy-aware

1 Introduction

To overtake the energy limitations in sensor networks, selective transmission strategies allow nodes to discard messages if the energy cost of transmission is not compensated by the importance of message content. This is used as a basis for censoring sensors in decentralized detection [1] and other schemes [5, 2]. A common assumption in these proposals is that the energy consumption of the sensors and their battery level are known. Thus, free parameters of the selective algorithms can be directly related to the energy consumption [2] or depend on them [6]. Moreover, to increase the benefits of discarding messages, selective transmission strategies assume that the cost of transmission is much higher than that of reception. These assumptions are not trivially satisfied. The average power consumption of transmission and reception depends not only on protocol design at all levels of the communication stack but also on the specific network deployment. As a consequence, the configuration of the free parameters in selective transmission algorithms cannot be done in advance and hence adaptive techniques must be applied.

This paper analyzes the problem of estimating the free parameters of a selective transmitter by correctly estimating the energy spent in each of its respective

^{*} This work was partially funded by projects TEC2008-01348 from the Spanish MCI, and SensoByg (<http://sensobyg.dk/english>) at Aarhus University.

states. Our work can be taken as a case study: we have tested the implementation of the selective transmission strategy proposed in [2] using TinyOS with its Low Power Listening (LPL) MAC layer running on a number of Tmote Skys (motest).

2 Selective Transmission

The selective transmission model in [2] defines the state of any given node at time k by two variables: the energy reserve, e_k , and the importance of the message to be transmitted, x_k . This importance value may reflect, e.g., the priority, the relevance or the quality of the information conveyed, and it is assumed to be provided by the application layer to the node carrying the message. Furthermore, we assume that the network routing algorithm (whatever it is) has defined a set of neighbors for each node, in such a way that any sensor node holding a message at time k has to make a decision about sending or not the message to one of its neighbors. The decision rule is a function of the node state.

We consider four energy expenses that deplete batteries: data collection by a sensing device, e_S , reception of a message from other node, e_R , message transmission, e_T and idle state e_I . After receiving (or sensing) a message, the battery level decreases e_R (or e_S) if it is not transmitted, and $e_R + e_T$ (or $e_S + e_T$) if it is transmitted. If no messages are received, the node consumes e_I . In general, e_S , e_R , e_T and e_I may depend on k .

A selective transmitter is allowed to discard low graded messages with the expectation of transmitting more important upcoming messages later. Optimal policies to maximize the expected importance sum of all messages transmitted during the node lifetime have been derived [2]. If the importance sequence, x_k , is stationary, it turns out that, for large values of the available energy, e_k , the optimal decision rule is: to transmit the message at time k iff the importance value is higher than some threshold, μ , which is given by the solution of

$$\mu = \rho \mathbb{E}\{\max\{x_k - \mu, 0\} | x_k > 0\}. \quad (1)$$

Though (1) cannot be solved in closed form, we derive the following stochastic rule which provides a simple estimate that converges to μ for large k ,

$$\mu_k = \left(1 - \frac{1}{k}\right) \mu_{k-1} + \frac{\rho}{k} \cdot \max\{x_k - \mu_{k-1}, 0\}. \quad (2)$$

Parameter ρ depends on the average node statistics as

$$\rho = \frac{(1 - P_I)E_T}{P_I E_I + P_R E_R + P_S E_S}, \quad (3)$$

where $E_S = \mathbb{E}\{e_S\}$, $E_R = \mathbb{E}\{e_R\}$, $E_T = \mathbb{E}\{e_T\}$, $E_I = \mathbb{E}\{e_I\}$, and P_I , P_R and P_S are the probability of a node being in idle, *receive* or sensing mode, respectively³. The estimation of all these energetic statistics is discussed in the following.

³ Actually, the model in [2] does assume $E_S = E_R$. We have generalized the model to assume different costs for sensing and receiving, which is more realistic.

3 Implementation

We implemented the selective transmission algorithm in TinyOS 2.x using its default CC2420 radio stack with its Carrier Sense Multiple Access (CSMA) channel access mechanism below its X-MAC [3] variant of a Low Power Listening (LPL) MAC layer. We used the component-based structure of TinyOS to add local modifications to the radio stack in order to support energy estimation according to the radio states and mote statistic according to the LPL layer states. Fig. 1 gives an overview of our implementation.

3.1 Energy Consumption

To measure the energy consumption online at a sensor we use software-based on-line estimation [4] based on the radio and sensor state to deplete a virtual battery. We neglect the energy consumed by the micro-controller. The energy measurements are based on the reported current consumptions from the data-sheets (19.6 mA during reception, 17.04 mA during transmission, 1 mA by the SHT11 Temperature sensor) multiplied by the amount of time spent in each state, measured with a 32 kHz timer. Battery is depleted according to the last state at every state change. The physical layer (PHY) of the radio stack was modified to report the off, receive, or transmit radio states to the energy measurement component (Fig. 1).

3.2 Mote Statistics

Using an asynchronous LPL MAC layer, sensor node lifetime is divided into three states: *idle*, *receive*, and *transmit*, from which we estimate the energy costs E_I , E_R , and E_T , and the probabilities P_I and P_R . These estimations are done in the mote statistic components based on the *idle*, *receive*, and *transmit* states together with all packet reception events and timer wake-ups reported from the MAC layer (see Fig. 1, which also shows how the mote statistics component uses the changing battery level from the energy measurement component).

Fig. 2 illustrates the definition of *idle*, *receive*, and *transmit* states reported from the MAC layer to the mote statistic component. Fig. 2(a) shows how the MAC layer handles receptions. After sleeping for a fixed period of time, it wakes up the radio, samples the channel for energy, and only stays awake if energy is detected. Otherwise, it is an *idle* state. Energy might be detected due to noise, which we also refer to as an *idle* state, or due to an actual packet reception, which

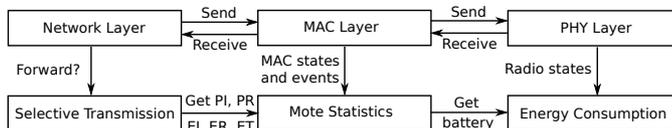


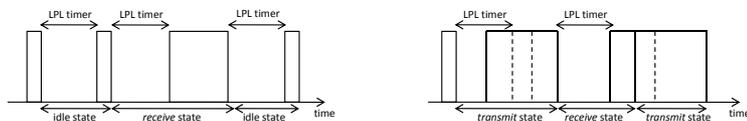
Fig. 1. Embedded Software Architecture

we refer to as a *receive* state. The length of a *receive* state varies with the number of received packets. To guarantee that a receiver is awake during transmission and acknowledges a successful reception, a transmitter repeatedly sends the same packet for a duration longer than the sleep interval. The transmission of a packet (referred to as *transmit* state) can interfere with the *idle* and *receive* states (see Fig. 2(b)). First, the transmission may happen during a sleep period, where the radio is then turned on and the next wake-up is ignored (an *overlap*). Second, the transmission may happen when the radio is already on so that the current *receive* (or *idle*) state is cutoff once the transmission starts (a *take-over*).

On top of normal packet receptions during a *receive* state and wake-ups, the MAC layer also reports free receptions. As packet transmission includes a waiting period for acknowledgments (ACKs) a sensor might receive a packet from another neighbor during transmission. We refer to these receptions as free in the sense that they do not imply any cost (the radio is in *receive* mode already).

Parameters E_I , E_R and E_T can be adaptively estimated as the sample averages of e_I , e_R and e_T , respectively, during the node lifetime. The idle state costs is the simplest case. If e_I is the energy consumption of the k -th idle state, the average idle energy is computed iteratively as $E_I(k) = (1 - \frac{1}{k}) E_I(k-1) + \frac{1}{k} e_I$. If m packets are received during a single *receive* state, then $E_R(k+m-1) = (1 - \frac{m}{k+m-1}) E_R(k-1) + \frac{1}{k+m-1} e_R$. The computation of the transmit state cost E_T is analogous to E_I , $E_T(k) = (1 - \frac{1}{k}) E_T(k-1) + \frac{1}{k} e_T$. However, the computation of e_T after each *transmit* state is quite involved. To understand it, note that e_T represents the cost overhead of deciding to transmit. If *transmit*, *receive* and *idle* states did not overlap, this would be equivalent to the energy of the *transmit* state. In general, however, this is not the case, and e_T must be computed as the difference between the cost of the *transmit* state (denoted as e_{Tr}) and the energy the node would have expended if the packet had been discarded. Therefore, a *transmit* state may take place in a non-interfering way (Fig. 3(a)), interfere with the *idle* and *receive* states by overlapping (Fig. 3(b)) or taking them over (3(c)). We analyze these three cases to compute e_T .

Case 1: Non-interfering transmissions (Fig. 3(a)). If no free receptions happen during the transmission, e_T is simply the cost of the *transmit* state e_{Tr} . However, if n free receptions happen, the overhead depends on the next state. If it is *idle*,



(a) *Idle* and *receive* states without the influence of overlapping *transmit* states. (b) Overlapping *transmit* states which takes over the *idle* or *receive* states.

Fig. 2. States of the LPL MAC layer. The dashed lines indicate what would have happened in case of no overlapping transmissions.

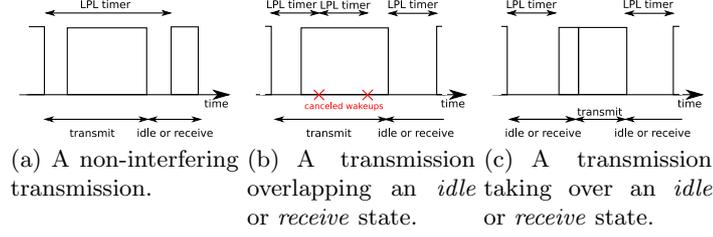


Fig. 3. The three cases of a transmission.

it would have been a *receive* state in case we had not transmitted, so that $e_T = e_{Tr} + E_I - nE_R$. But if it is a *receive* state receiving m packets, it would have been a *receive* state receiving $m+n$ packets in case we had not transmitted, and $e_T = e_{Tr} + mE_R - (m+n)E_R = e_{Tr} - nE_R$. We will later refer to the stored e_{Tr} and n as the pending transmission cost and pending free receptions.

Case 2: Overlapping transmissions (Fig. 3(b)). An overlapping transmission can be detected by the number of canceled wake-ups, w , according to the LPL MAC layer. If no free receptions happen, the cost needs to be compensated with w *idle* states. However, if n free receptions uniformly distributed during transmission happen, some of the *idle* states would have been turned into *receive* states. Then, the overhead is $e_T = e_{Tr} - nE_R - \max\{w - n, 0\} + E_I$.

Case 3: Taking-over transmissions (Fig. 3(c)). The transmission will shorten the *idle* or *receive* and then, it will be handled as a non-interfering transmission. Defining the measured energy cost of the taken over *idle* or *receive* state as d_I and d_R , respectively, the expected remaining energy cost of the taken over *idle* or *receive* state, if the transmission did not take place, is subtracted from the measured transmission cost e_{Tr} . Thus, the overhead of transmission is $e_T = e_{Tr} - (E_I - d_I)$ if the previous state was *idle* and $e_T = e_{Tr} - (E_R - d_R)$ if it was a *receive* state. In this case, we do not update E_I and E_R .

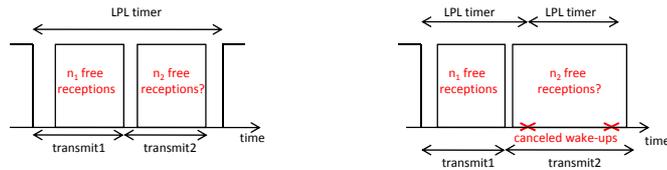


Fig. 4. Two different scenarios of a successive transmission following a non-interfering transmission with free receptions.

Furthermore, several successive (but still independent) transmissions may happen after each other. This has a consequence for the non-interfering transmission with free receptions as the overhead cannot be estimated until the next *idle* or *receive* state. In Fig. 4(a), the second transmission does not overlap with any wake-ups. If it does not contain any free receptions, it can be handled as a non-interfering transmission. Instead, if the successive transmission contains free receptions, we have two pending transmissions whose overhead can not be estimated until the next *idle* or *receive* state. As E_T measures the average transmission overhead, we let the pending transmission costs and free receptions be an average of the two. In Fig. 4(b), the successive transmission does overlap with some wake-ups. The free receptions from the first transmission should be transferred to these wake-ups. The first wake-up is then handled as a non-interfering transmission with no free receptions and the second is handled as an overlapping transmission with the free receptions from both transmissions.

Since the local data captured by the sensing device runs in parallel with the states of the MAC layer, the specific sensing cost cannot be separated from the communication costs. Consequently, parameters E_I , E_R , and E_T become overestimated. However, it can be shown that by underestimating the sensing cost as $E_S = 0$, we have compensated for this overestimation.

Conventional frequency-based estimates are used to compute P_I and P_R (we have no need for P_S when $E_S = 0$), and to save memory all energy and probability estimates for $k < 100$ are based on a limited Exponential Weighted Moving Average (EWMA). For $k \geq 100$, E_I , E_R and E_T are replaced by $E_x(k) = 0.99E_x(k-1) + 0.01e_x$, which allows the estimate to adapt to changes in the environment.

3.3 Selective Transmission

The selective transmission component is called by the network layer (see Fig. 1) whenever the sensor node needs to make a decision about a packet transmission. It makes use of E_T , E_I , E_R , P_I , and P_R from the mote statistics to estimate ρ in (3), which is then used in (2).

4 Experiments and Evaluation

We ran the selective transmission implementation on top of the TinyOS CC2420 LPL layer on a number of Tmote Sky platforms. All experiments average 5 similar runs with an LPL interval of $500ms$, a data rate of one packet per $2s$, and a delay to turn off the radio after packet reception of $30ms$ (which is used as a mechanism in TinyOS to allow a transmitter to send a number of packets without a receiver turning off its radio). To avoid data periods from multiple transmitters to be synchronized, the data timer is set to a random time during the second half of the interval. Importance values of messages, which should be provided by the application layer to source nodes, are assigned according to random samples of a long ramp distribution (1,2,4,8,16,32) with decreasing probabilities. Moreover,

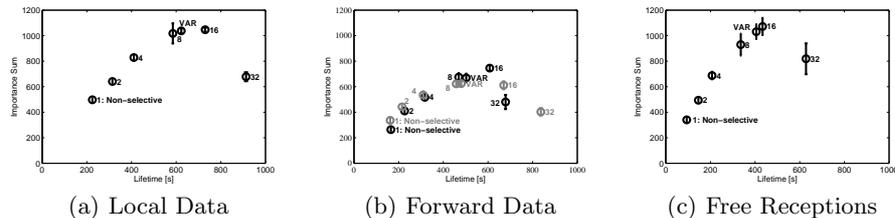


Fig. 5. Importance sum and lifetime for variable and all fixed thresholds.

messages are sent immediately after generated or received. The initial battery of sensor nodes is set to $200mAh$, and a sensor node is considered dead once its battery depletes. In the following, we compare the implementations based on the total importance sum of messages sent throughout the sensor node lifetime.

To test the benefits of selective transmission based on locally sensed data we deployed two nodes, a transmitter and a receiver, within radio range. The transmitter periodically senses data and makes a transmit or discard decision. Fig. 5(a) shows the average importance sum and its standard deviation, and the (average) transmitter lifetime, for the adaptive threshold (based on (2) and the energy estimates, labelled as VAR) and for all fixed thresholds (1, 2, 4, 8, 16 and 32, where the number indicates the minimum importance value that is transmitted). Selective transmission policies outperform the non-selective transmitter (threshold 1). Moreover, the adaptive threshold performance is close to the best threshold, 16. The slight differences can be explained by some suboptimal decisions during the initial steps, when the energy estimates are not accurate.

To test the selective transmission when the node receives data, we deployed three nodes on a line: a non-selective transmitter, a selective forwarder, and a receiver. The transmitter periodically generates and sends packets through the forwarder (which does not sense data) to the receiver. Fig. 5(b) (black plot) shows that the importance sum of the adaptive threshold is closer to 8 than to the best constant threshold, 16. A further analysis of this case showed that the suboptimal behavior can be explained by the randomness of the importance sequence. Further tests with a periodical importance sequence (still keeping the same frequencies of each importance values), which reduce randomness, demonstrate that the variable threshold behaves optimally (Fig. 5(b), gray plot).

To test selective transmission with free receptions we placed three nodes on a line again, but this time with both the transmitter and the forwarder periodically generating data and sending them to the receiver. Free receptions may happen at the forwarder when it starts to transmit local generated packets before it receives a packet from the transmitter. Fig. 5(c) shows that the adaptive threshold accurately predicts the best constant threshold in this case as well.

Fig. 6 (left) shows the final probability estimates P_I and P_R for all thresholds in the free reception scenario. The lower the threshold is, the higher the probability of reception is, because the node is less selective and therefore, P_I decreases. Fig. 6 (right) shows the final energy estimates for all thresholds in

the free reception scenario. Although the current consumption during reception is slightly higher than that of transmission (see Sec. 3.1), E_T is much higher than E_R because of the longer time spent by nodes during transmission states. This explains the considerably superior performance of the selective transmission policies with respect to a non-selective transmission in the chosen scenario. The correctness of the energy and probability estimates is implied by the fact that the adaptive threshold correctly predicts the best constant threshold. As sanity check, we can compare the energy estimates to the expected maximum transmission cost (assuming current consumption of $20mA$) of $10mC$ and see that on average it is a bit above 50% which is expected for a LPL MAC protocol.

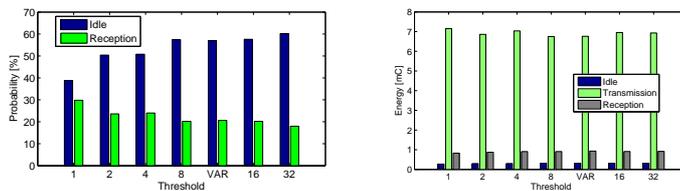


Fig. 6. Probability (left) and energy (right) estimates for all thresholds in a free reception scenario.

5 Conclusion

The implementation of a selective transmission policy on top of a real LPL MAC protocol using a specific procedure to estimate energy consumption of the node states has shown how this kind of strategies can be used to extend the network lifetime and maximize the total importance sum of the transmitted messages. Future work includes the performance analysis in a larger testbed and more complex scenarios (e.g. interferences, link quality, etc).

References

1. Appadwedula, S., Veeravalli, V., Jones, D.: Decentralized detection with censoring sensors. *IEEE Transactions on Signal Processing* 56(4), 1362–1373 (2008)
2. Arroyo-Valles, R., Marques, A.G., Cid-Sueiro, J.: Optimal Selective Transmission under Energy Constraints in Sensor Networks. *IEEE TMC* 11(8), 1524–1538 (2009)
3. Buettner, M., Yee, G.V., Anderson, E., Han, R.: X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In: *SenSys '06: 4th Int. Conf. on Embedded Networked Sensor Systems*. pp. 307–320. New York, NY, USA (2006)
4. Dunkels, A., Osterlind, F., Tsiftes, N., He, Z.: Software-based on-line energy estimation for sensor nodes. In: *EmNets '07: Procs. of the 4th workshop on Embedded networked sensors*. pp. 28–32. ACM, New York, NY, USA (2007)
5. Lei, J., Yates, R., Greenstein, L.: A generic model for optimizing single-hop transmission policy of replenishable sensors. *IEEE TWC* 8(2), 547–551 (2009)
6. Quek, T., Dardari, D., Win, M.: Energy efficiency of dense wireless sensor networks: To cooperate or not to cooperate. *IEEE JSAC* 25(2), 459–470 (2007)