

# Optimizing sleeping intervals in preamble sampling MAC for WSNs

Holger Karl

Universität Paderborn, Computer Networks group

**Abstract.** Preamble sampling is a popular mechanism in WSN MAC protocols. This paper optimizes the energy consumption by adjusting the preamble length to a known or estimated, possibly varying event rate.

## 1 Introduction

Among the many proposed energy-efficient MAC mechanisms for WSNs, *preamble sampling* [1] is used in various protocols (WiseMAC [3], B-MAC). The idea is to have the receiver wake up with a period  $\Delta$  and check whether there is any transmission currently ongoing. If so, the receiver stays awake until the start of the actual packet; else, it immediately goes back to sleep. A transmitter must ensure that a receiver will actually stay awake before it transmits the packet by sending a *preamble* for (at least) a length  $\Delta$ .

El-Hoiydi [1,2] has analyzed preamble sampling's delay and throughput, extending the classical Aloha-type analysis. What is missing is an optimization of the preamble length for Poissonian event arrivals and an analysis how to adapt the length to an unknown arrival rate.

This paper is a first step to such an optimization. To keep the analysis tractable, only the case of a single transmitter/receiver is considered; a multi-transmitter MAC case is left for future study. For a single transmitter/receiver pair, the paper analyzes (Section 2) and optimizes (Section 3) the preamble length  $\Delta$  for energy consumption and provides a simple-to-use approximation formula suitable for in-field adaptation; it characterizes the overhead necessary to adapt to an unknown (Poissonian) arrival rate (Section 4), and it looks at the overhead caused by Markov-modulated Poisson traffic (Section 5).

## 2 Analysing energy consumption at given preamble length $\Delta$

Assume a transmitter with Poisson events of rate  $\lambda$ . The packet duration is  $T_{\text{Pkt}}$ , the transmission power is fixed at  $P_{\text{TX}}$ , the power consumed during reception (of preamble or packet) is  $P_{\text{RX}}$ . For waking up and checking for a preamble, the receiver consumes an energy  $E_{\text{wkup}}$ . There are two main simplifications here: First, the CSMA aspect is ignored since there is only a single transmitter; second, the time needed for detecting preamble/idle channel is assumed to be small to

$\Delta$  and is thus ignored. Also, the receiver restarts its sleeping/sampling cycle *after* a packet has been received (because of Poisson events, assuming that the statistics of the time to the next event is unchanged is acceptable).

The first step is to find the expected energy consumption as a function of  $T_{\text{Pkt}}$ ,  $P_{\text{TX}}$ ,  $P_{\text{RX}}$ ,  $E_{\text{wkup}}$ ,  $\lambda$ , and  $\Delta$  (expectation taken over the random arrival process). To do so, the distribution of the number of wakeups before detecting a preamble and of the remaining preamble listening time are derived.

## 2.1 Number of wakeups

Let  $X$  be a random variable describing the inter-event times (and, thus, roughly the times between transmissions of a preamble);  $X$  is exponentially distributed with parameter  $\lambda$  and successive  $X$  are independent of each other. Let  $Z$  be a random variable describing the number of wakeups a receiver executes before receiving the preamble. The density of  $Z$  is:<sup>1</sup>

$$\begin{aligned} P(Z = k) &= P(k\Delta \leq X \leq (k+1)\Delta), \quad k \geq 0 \\ &= P(X \leq (k+1)\Delta | k\Delta \leq X) P(k\Delta \leq X) \\ &= (1 - P(X \geq (k+1)\Delta | X \geq k\Delta)) P(X \geq k\Delta) \\ &= (1 - P(X \geq \Delta)) P(X \geq k\Delta) = (1 - e^{-\lambda\Delta}) e^{-k\lambda\Delta} \end{aligned}$$

Note that from line 3 to 4, the memoryless property of  $X$  is used. Thence:

$$E[Z] = \frac{e^{-\lambda\Delta}}{1 - e^{-\lambda\Delta}} \quad (1)$$

## 2.2 Time to listen to preamble

Suppose now that the receiver has detected a preamble. How long does it have to listen to the preamble before the actual packet starts? Let the random variable  $Y$  describe this time. It is easiest analyzed looking at its complementary cumulative distribution function, again using the memorylessness of  $X$  in the second step of Equation (2);  $F_X$  is the cumulative distribution function of  $X$ ;  $0 \leq y \leq \Delta$ .

$$\begin{aligned} P(Y \geq y) &= P(X > y + k\Delta | X > k\Delta \wedge X < (k+1)\Delta) \text{ for some } k \\ &= P(X > y | X > 0 \wedge X < \Delta) \\ &= P(X > y | X < \Delta) = \frac{P(y < X \wedge X < \Delta)}{P(X < \Delta)} = \frac{F_X(\Delta) - F_X(y)}{F_X(\Delta)}. \end{aligned} \quad (2)$$

Thus,  $P(Y \leq y) = 1 - (1 - F(y)/F(\Delta)) = (1 - e^{-\lambda y})/(1 - e^{-\lambda\Delta})$ . Some arithmetic then yields the expected preamble receive time:

$$E[Y] = \frac{1}{\lambda} - \frac{\Delta e^{-\lambda\Delta}}{1 - e^{-\lambda\Delta}}. \quad (3)$$

---

<sup>1</sup> In essence, this derives the known result that an exponential r.v. corresponds to a geometrically distributed number of fixed time slots.

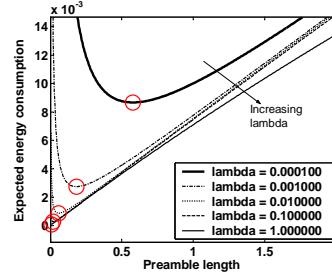
### 2.3 Expected energy consumption per MAC interaction

Putting Equations (1) and (3) together gives the expected energy consumption:

$$\begin{aligned} E[\text{energy}] &= (\Delta + T_{\text{Pkt}})P_{\text{TX}} + E[Z]E_{\text{wkup}} + (E[Y] + T_{\text{Pkt}})P_{\text{RX}} \\ &= (\Delta + T_{\text{Pkt}})P_{\text{TX}} + \frac{e^{-\lambda\Delta}}{1 - e^{-\lambda\Delta}}E_{\text{wkup}} + \left(\frac{1}{\lambda} - \frac{\Delta e^{-\lambda\Delta}}{1 - e^{-\lambda\Delta}} + T_{\text{Pkt}}\right)P_{\text{RX}}. \end{aligned} \quad (4)$$

The term  $(\Delta + T_{\text{Pkt}})P_{\text{TX}}$  is the transmitter's energy consumption to transmit a packet, the term  $E[Z]E_{\text{wkup}}$  the energy for the wakeup attempts, and the last term is the energy to receive the remaining preamble and the actual packet. Any additional overhead for ACKs or retransmissions is not accounted for.

Figure 1 illustrates Equation (4), using parameters similar to those in reference [1]; the circles highlight the optimal  $\Delta$  for a given  $\lambda$ .



**Fig. 1.** Expected energy consumption (in J) for various  $\lambda$  (in 1/seconds) as a function of  $\Delta$  (in seconds);  $P_{\text{TX}} = P_{\text{RX}} = 5 \text{ mW}$ ,  $T_{\text{Pkt}} = 1 \text{ ms}$ ,  $E_{\text{wkup}} = 0.25 \mu\text{J}$

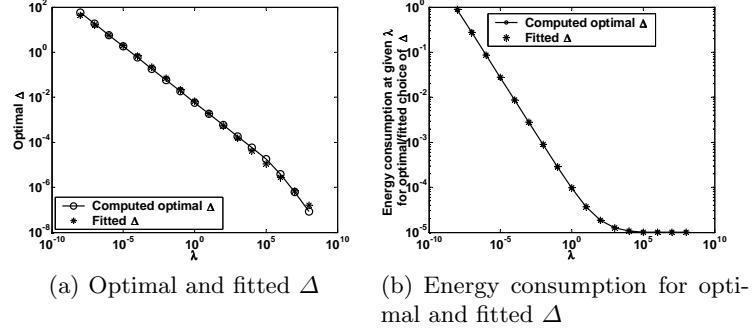
### 3 Optimize energy consumption in $\Delta$

Choosing an energy-optimal  $\Delta_{\text{opt}}$  is, in principle, easy given Equation (4). An analytic derivation, however, becomes unwieldy and would hardly be practical to use on a wireless node at runtime. It is also not necessary. Rather, for given values of  $T_{\text{Pkt}}$ ,  $P_{\text{TX}}$ ,  $P_{\text{RX}}$ , and  $E_{\text{wkup}}$ , a simple approximation of  $\Delta_{\text{opt}}$  as a function of  $\lambda$  can be derived by regression fitting Equation (4).

As it turns out, the *logarithms* of  $\lambda$  and  $\Delta$  can be easily fitted using a quadratic polynomial, shown in Equation (5);  $\Delta_f$  indicates the fitted value for the optimal  $\Delta$ . (Linear fits are not quite acceptable over a wide range of  $\lambda$ .)

$$\log \Delta_f = a \log^2 \lambda + b \log \lambda + c \quad (5)$$

For the example parameters used in Section 2.3,  $a = -0.0026$ ,  $b = -0.5269$ ,  $c = -5.0171$ . The fit for  $\Delta$  is shown in Figure 2(a), the resulting energy consumption in Figure 2(b). In this example, the largest increase in energy consumption when using the fitted  $\Delta$  instead of the optimal computed  $\Delta$  is about 3.5%.



**Fig. 2.** Fitted  $\Delta$  and resulting energy consumption for various  $\lambda$ ;  $P_{\text{TX}} = P_{\text{RX}} = 5 \text{ mW}$ ,  $T_{\text{Pkt}} = 1 \text{ ms}$ ,  $E_{\text{wkup}} = 0.25 \mu\text{J}$  (note double-logarithmic scales)

#### 4 Adapting $\Delta$ to an unknown, fixed arrival rate

Using such a regression-based fit, even a sensor node can choose a near-optimal  $\Delta$  for a given  $\lambda$ . However,  $\lambda$  is usually not known. A simple idea is to use observed interarrival times of events and to estimate the actual  $\lambda$ . A common option would be to store a few observation and then use a maximum likelihood estimator, but this needs memory. Alternatively, a simple autoregressive estimation of the mean arrival time can be attempted: Maintain an estimate  $\lambda_i$  of the arrival rate after  $i$  events have been observed, update this estimate using the interarrival time (IAT) of the  $i + 1$ st event and a constant smoothing factory  $\alpha \in (0, 1)$ .

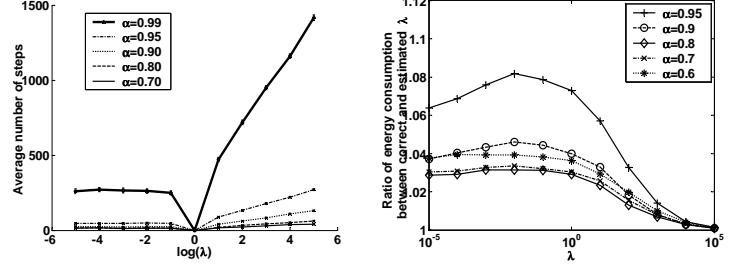
$$1/\lambda_{i+1} = \alpha/\lambda_i + (1 - \alpha)\text{IAT}_{i+1} \quad (6)$$

Assuming an arbitrary, initial arrival rate of, say,  $\lambda_0 = 1$ , and using the estimated  $\lambda_i$  to derive  $\Delta$  from the fitted model, what is then the energy overhead? First, Figure 3(a) shows the average number of steps to adapt  $\lambda$ , from an arbitrary start value of 1, using different adaption parameters  $\alpha$ .

The energy consumption itself can be obtained by simulation. Assuming a preamble length  $\Delta$  and the time to the next event is TTE, then the actually (not expected) consumed energy is given by Equation (7).

$$E_{\text{actual}} = (\Delta + T_{\text{Pkt}})P_{\text{TX}} + \left\lceil \frac{\text{TTE}}{\Delta} \right\rceil E_{\text{wkup}} + (\text{TTE} - \Delta \left\lceil \frac{\text{TTE}}{\Delta} \right\rceil + T_{\text{Pkt}})P_{\text{RX}} \quad (7)$$

Thence, the energy consumed when using (clairvoyantly) the correct  $\lambda$  to compute  $\Delta$  or an estimate of  $\lambda$  as computed according to Equation (6) can be compared. Figure 3(b) shows this ratio: For a wide range of  $\lambda$ , starting from an initial estimate of  $\lambda = 1$  only has a small energy overhead when adapting  $\lambda$  and, thus,  $\Delta$ . Results were obtained using the parameter settings from above, 200 events, and averaging over 200 independent iterations (clearly, the more steps are used, the lower the overhead for the initial adaptation becomes – this is addressed in the following section). Apparently,  $\alpha = 0.8$  seems like a good choice.

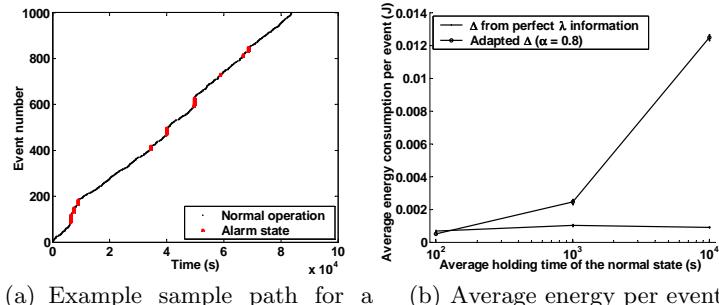


(a) Average number of steps to adapt estimation of  $\lambda$  to within 5 % when using estimated  $\lambda$  values to of actual value, using different  $\alpha$  to compute  $\Delta$  compared to using correct  $\Delta$

**Fig. 3.** Adapting  $\Delta$  to a fixed  $\lambda$  (confidence intervals for confidence level 0.95)

## 5 Adapting $\Delta$ to a Markov-modulated Poisson process

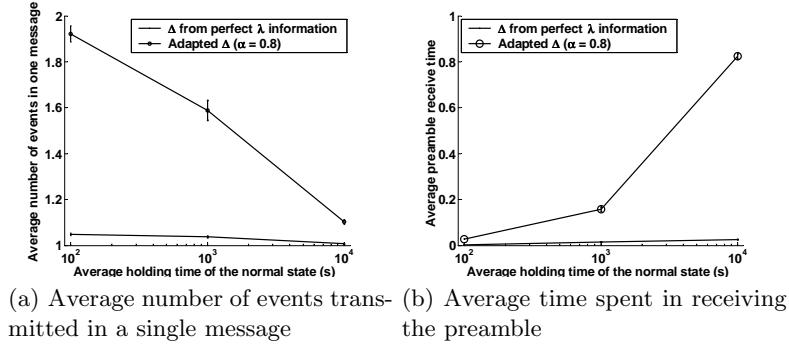
For a WSN MAC, an important function is to be able to change between “modes”, e.g., “normal” and “alarm”, with severe changes in event arrival rates. One adequate model is a Markov-modulated Poisson process (MMPP), where a two-state Markov process represents “normal” and “alarm”; the two associated Poisson processes have low or high rate. As example, state holding times of 10000 and 1 seconds and corresponding rates of 1/100 and 10 1/s are used; Figure 4(a) shows a sample path.



**Fig. 4.** Adapting  $\Delta$  to a Markov-modulated Poisson traffic

The average consumed energy is larger when adapting  $\Delta$  to the observed IAT values instead of to the correct  $\lambda$  (Figure 4(b)). The apparently comparable behavior for low holding times is deceptive, caused by fewer messages and more events per message because of an incorrect  $\Delta$  (Figure 5(a)). The high energy

consumption of an incorrect  $\Delta$  is mainly caused by long times spent in sending/receiving the preamble (Figure 5(b)).



**Fig. 5.** Details on protocol behavior

## 6 Conclusion

To conclude: Optimizing the preamble length to the traffic rate is important for preamble sampling. For a fixed traffic rate, it is even possible to automatically adapt without too high an energy penalty. For varying traffic patterns, however, a self-adapting MAC protocol runs the danger of (a) missing or delaying messages (when switching to alarm mode) and (b) running at high overhead when switching back to normal mode. The crucial point is that even though the transmitter might be aware of a change to a high traffic rate, there is no means of informing the receiver of such a change before the next  $\Delta$ ! A change to a low rate can be more easily announced to the receiver, assuming the sender has this knowledge – but this knowledge would have to be present at the *last* message sent during the alarm mode, at the high data rate, to give the receiver an indication to switch to a  $\Delta$  for lower traffic rates. This might not always be feasible, either. Hence, additional MAC or cross-layer mechanisms are necessary.

## References

1. A. El-Hoiydi. Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks. In *Proc. IEEE Intl. Conf. on Communications (ICC)*, April 2002.
2. A. El-Hoiydi. Spatial TDMA and CSMA with preamble sampling for low power ad hoc wireless sensor networks. In *Proc. Seventh Intl. Symp. on Computers and Communications (ISCC'02)*, page 685, Washington, DC, USA, 2002.
3. A. El-Hoiydi, J.-D. Decotignie, C. Enz, and E. Le Roux. Wisemac, an ultra low power mac protocol for the wisenet wireless sensor network. In *Proc. 1st Intl. Conf. on Embedded Networked Sensor Systems (SenSys)*, pages 302–303, November 2003.