A prototype low-cost wakeup radio for the 868 MHz band

Bas van der Doorn

Faculty of Electrical Engineering, Mathematics, and
Computer Science,
Delft University of Technology,
Mekelweg 4,
2628 CD Delft, The Netherlands
E-mail: b@doornvd.com

Winelis Kavelaars

SOWNet Technologies BV,
Delftechpark 26,
2628 XH Delft, The Netherlands
E-mail: w.kavelaars@sownet.nl

Koen Langendoen*

Faculty of Electrical Engineering, Mathematics, and
Computer Science,
Delft University of Technology
Mekelweg 4,
2628 CD Delft, The Netherlands
E-mail: k.g.langendoen@tudelft.nl
*Corresponding author

Abstract: In many wireless sensor network scenarios, battery-powered nodes must operate for years, which necessitate the need for advanced power management of the radio. In this paper, we pursue the idea of using a second, ultra low-power radio that can be used to trigger a remote interrupt, so that a receiver can fire up its primary radio to engage in efficient high-speed communication with the sender. Wakeup radio avoids the complex bookkeeping associated with energy-efficient MAC protocols, but at the price of additional hardware. We present the first design of a low-cost wakeup radio made out of standard components. Our working prototype implementation operates in the European license-free 868 MHz band, and includes a specialised low-power microcontroller filtering out interference (e.g. GSM signals), which makes it practical for use in many real-world settings.

Keywords: WSNs; wireless sensor networks; low power; link layer; experimentation; wakeup radio.

Reference to this paper should be made as follows: van der Doorn, B., Kavelaars, W. and Langendoen, K. (2009) 'A prototype low-cost wakeup radio for the 868 MHz band', *Int. J. Sensor Networks*, Vol. 5, No. 1, pp.22–32.

Biographical notes: Bas van der Doorn currently holds a position as Software Engineer with SOWNet Technologies BV. He studied computer science at Delft University of Technology. He spent his final year with the Parallel and Distributed Systems (PDS) group working on the design and implementation of the wakeup radio described in this paper as part of his research project. He successfully defended his MSc thesis in May 2007.

Winelis Kavelaars received a BSc degree in electrical engineering from Utrecht University in 1999, after which he started his professional career at KPN (the Dutch national telephone company). Three years later, he accepted a position as Project Leader at TNO Defense, Safety and Security. At TNO, he started and led a five-year project to develop wireless sensor network technology for military purposes, which included the development of a wireless compound security system. In 2006, he left TNO and founded SOWNet Technologies BV, which develops wireless sensor network solutions for the commercial sector.

Koen Langendoen is a Full Professor of Computer Science, and holds the chair on Embedded Software at Delft University of Technology. He received his MSc degree in computer science from the Vrije Universiteit, Amsterdam in 1988 and PhD degree in computer science from the Universiteit van Amsterdam in 1993. He has been active as a postdoc in the area of parallel and mobile computing. Currently, he leads a research group in wireless sensor networks, working on localisation algorithms, MAC protocols and various pilot applications.

1 Introduction

An important selling point of Wireless Sensor Networks (WSNs) is that they are easy to deploy; no wires, cables or other infrastructure is necessary to get started, because nodes are self-supportive by including sensing, processing, storage and communication capabilities. Current-generation node hardware usually operates with batteries, since other, ambient sources of energy (like solar cells) do not provide the required power levels, interfere with operational conditions or negatively impact form factor and shape. A serious problem with using batteries is that the amount of energy is severely limited, which calls for advanced power management complicating the operation of the network. For example, running a Mica2 node flat out will drain a pair of penlight batteries in just a few days instead of a few years, as dictated by maintenance cost.

Given that a node's radio is the dominating source of energy consumption – even when running idle, listening for potentially incoming traffic – many novel Medium Access Control (MAC) protocols have been developed by researchers in the WSN community. These protocols typically trade off performance (latency, throughput) for a reduction in energy consumption by duty cycling the radio. Protocols differ in their approach to switching the radio into sleep mode, ranging from unsynchronised periodic sleeping [e.g. B-MAC (Polastre et al., 2004) and WiseMAC (El-Hoiydi and Decotignie, 2004)] to precisely timed schedule-based access [e.g. LMAC (van Hoesel and Havinga, 2004) and TRAMA (Rajendran et al., 2003)]. Although diverse in nature, these protocols share the property that nodes switch their radio on and off according to a regular schedule. The advantage is that this regularity provides the knowledge of when neighbouring nodes are awake and can be sent to. The disadvantage is that the enforced regularity rarely matches an application's needs compromising energy efficiency or performance. This holds especially for event-based applications in which an external event triggers network activity as, for example, in the case of an alarm system being activated by a burglar. Since low latency is important in this example, nodes must wakeup frequently to check for potential alarm messages that need to be forwarded, wasting energy in doing so.

The fundamental uncertainty of when an incoming message is to be expected has a negative impact on all software-based approaches to power management. An alternative, hardware-based approach is to use a second, ultra low-power radio that can be used to trigger a remote interrupt at a 'sleeping' neighbour, who can then fire up its primary radio to engage in efficient high-speed communication with the sender. The secondary radio, also known as the wakeup radio, can be extremely simple since it only has to be capable of sending/receiving a single type of signal. This minimises hardware costs and energy consumption, but makes it more sensitive to noise limiting the range in comparison to more complex designs. Nevertheless, the on-demand capabilities of the wakeup radio make it an attractive option for sensor networking. In many cases the sensing range, which is usually orders of magnitude smaller than the communication range (a few centimetres vs. tens of metres), calls for a dense deployment in which the range limitation of the wakeup radio is of little importance. Wakeup radio excels in combining low power and fast response, giving it an edge in various scenarios including the alarm system already mentioned and sensor nodes being queried by a mobile sink, for example, a car driving along a dike collecting soil moisture data from sensor nodes buried into the ground every kilometre.

Wakeup radio is not a new concept. It was already proposed in the early days of sensor networking, for example, the PicoRadio project (da Silva et al., 2001) includes a feasibility study with a wakeup radio consuming just 1 μW back in 2001. To date, however, we are not aware of any working prototype wakeup radio presented in the literature. The closest-related work, coming from the mobile computing domain, is the Wake-on-Wireless system (Shih et al., 2002) that includes an RFM TR1000 radio (consuming 7 mW) to wake up a Cisco AIR-PCM350 802.11b wireless networking card (consuming 1080 mW) in a PDA configuration. Noticing the lack of a working wakeup radio in the WSN community, we designed a low-cost version made out of standard components for use in the 868 MHz band. This design was motivated by practical concerns, and reuses as much of the primary (CC1000) radio as possible. In particular, the standard antenna is shared for receiving/sending wakeup signals, which are generated in software by toggling the send power of the CC1000 radio between minimum and maximum. In essence, our wakeup circuitry consists of a dedicated receive path, including a specialised low-power microcontroller to filter out interference (i.e. GSM signals), making it a practical and low-cost design.

To study the feasibility of the design we performed a detailed analysis in comparison with (models of) three advanced MAC protocols duty cycling the main radio. Next, we created a prototype implementation of the wakeup-radio design. It did not meet some goals (range, power consumption), but was evaluated to be fully functional. As such the prototype is a firm, first step towards a practical, low-cost wakeup radio; the paper details a number of improvements that should give the second-generation implementation the desired four-fold gain in lifetime over traditional MAC protocols for WSNs.

The remainder of this paper is structured as follows. Section 2 briefly discusses related work. Then, Section 3 details the design of the wakeup radio for a Mica2-like platform. Using key figures from this design, we explore the feasibility of the wakeup radio in comparison to state-of-the-art MAC protocols in Section 4. Next, in Section 5, we discuss practical issues that surfaced when implementing a prototype version, and then present experimental results in Section 6. Finally, Section 7 concludes the paper.

2 Related work

In 2001, the PicoRadio project was the first to advocate the advantages of wakeup radio and boldly estimated that a specialised radio interface could consume as little energy as 1 μW (da Silva et al., 2001). In follow-up work, the term *reactive radio* was introduced and the design target moved to a more realistic power consumption level of around 50 μW with some components like a highly optimised frequency oscillator being detailed (Pletcher, 2004). To date, no working reactive radio has been reported on.

Another design study was undertaken by Gu and Stankovic (2005), who proposed a fully passive design for the ultimate low-power wakeup receiver consuming no energy at all. Data from SPICE circuit simulations showed that this design could work, but that the operating range would be limited to at most 3 m. Gu and Stankovic propose several modifications to extend the range to about 10 m, but these involve active components like amplifiers and comparators, increasing the power consumption to around 100 μW. Again, no working hardware was demonstrated.

A more realistic setup for a wakeup radio was proposed as part of the STEM protocol (Schurgers et al., 2002). In this scheme, each sensor node would be equipped with two 'standard' radios operating on different frequency channels. One being used for medium reservation, by means of sending out a busy tone, the other for the actual data transfer after a wakeup tone was observed. To keep the power consumption of the wakeup (tone) radio as low as possible, STEM proposes to duty cycle this radio with 'sleeping' nodes regularly polling the tone channel for activity. Simulation results report reasonable latency with a 1% duty cycle, yielding an effective power consumption in the order of 500 μW. Today this level can also be achieved by advanced MAC protocols using a single radio.

An advantage of using an ordinary (duty-cycled) low-power radio, as STEM proposed, is that it allows for waking up specific nodes. In its simplest form, a wakeup (tone) signal contains no information and, consequently, all nodes receiving the wake up must turn on their primary radios to check if the message is destined to them. In the case of unicast traffic, all but one can return to sleep immediately. By encoding the address of the intended receiver in the wakeup signal, unnecessary wake ups of neighbouring nodes can be completely eliminated.

The costs of a single wakeup are fixed, but generally much higher than a single carrier sense as employed by advanced energy-efficient MAC protocols like WiseMAC and SCP-MAC. Please refer to a recent survey of WSN-specific MAC protocols by Langendoen (2008) for details. The wakeup costs may be amortised over periods of inactivity, in which MAC protocols regularly perform carrier sense operations. For low traffic wakeup radio performs best, but for higher data rates the polling-based mechanism use less energy (see Section 4). The work by Miller and Vaidya (2005) explores several schemes for predicting the next message arrival after an initial wakeup, which allows for scheduling an efficient carrier sense, effectively combining the best of both worlds.

3 Design

Given the lack of a working wakeup radio for sensor networks, we set out to design one that could interface with the T-node hardware platform in regular use by our research group. The T-node platform is quite similar to the familiar Mica2 mote from Crossbow, and is built around the 8 MHz ATmega128L processor and Chipcon CC1000 radio operating in the European license-free 868 MHz band through a 1/4 wavelength (8.6 cm) whip antenna. In the following sections, we will describe the basic requirements and overall design of the wakeup radio, and provide additional detail regarding the analog and digital signal processing aspects of the design.

3.1 Requirements

A practical and low-cost wakeup radio must fulfil a number of important requirements:

- *Standard components* should be used to speed up development and reduce cost in comparison to designing a single chip solution.
- *Low cost* is critical, since the additional hardware must be in the 5–10% price range of a complete node to make wakeup radio viable. In the case of the T-node platform, the costs of the wakeup radio should therefore not exceed €5. One way to reduce costs is to leverage as much of the T-node as possible. In particular, we can reuse the antenna and even the CC1000 radio for generating the wakeup signal. The consequence is that the wakeup radio must operate in the 868 MHz band.
- *Low interference* should be achieved to reduce the number of false alarms waking up nodes through the use of other wireless equipment like GSM cell phones and WLAN networks. In this respect, 'friendly fire' of random T-nodes communicating with each other may prove to be the biggest source of interference, causing unnecessary wake ups of neighbouring nodes. The inverse of missing a wakeup alarm should also be avoided, because retransmissions at the wakeup level are costly in terms of latency and energy.

- *Low power* of the wakeup circuitry is mandatory to outperform software-based alternatives. Given that the power consumption of a T-node in sleep mode is around 600 μW and near zero when switched off completely, the wakeup radio should not consume more than about 150 μW to make for a compelling extension of battery lifetime (i.e. a factor of 4).
- *Ten metre range* of the wakeup signal would serve many application scenarios. A shorter range renders the wakeup radio rather impractical, because of the very high node densities that such a range would require. For reference, the effective range for normal communication over the CC1000 radio ranges from 20 m (indoor, obstructions) to 70 m (outdoor, LOS) in our experience. A side effect of the shorter wake-up range is that messages travelling to a sink will have to make more hops; with a wakeup range of 10 m path length would roughly increase with a factor of 5.
- *Low latency* is needed to overcome the multi-hopping overhead, and to arrive at a design point that energy-efficient MAC protocols running with long sleep intervals cannot achieve (see Section 4).

An additional feature would be to include support for waking up specific neighbours instead of all nodes within reach. However, anticipating a rather short range (10 m) the number of neighbours will be limited anyway (around 5 seems reasonable), so unicast support by means of encoding an address in the wakeup signal is considered to be optional.

3.2 Wakeup circuitry

Figure 1 shows the high-level design of the wakeup circuitry and its interface to the T-node components. The incoming signals from the antenna are first fed through a frequency filter to suppress external interference (mainly GSM signals at the 900 MHz band) as much as possible. Then the 868 MHz signal is converted down to a low-frequency baseband signal using a simple diode, followed by a large amplification to make it detectable by an ultra low-power microcontroller. This microcontroller applies some digital processing to filter the residual (self-)interference still present in the input signal. Once a true wakeup signal is detected, the ATmega128L processor of the T-node is triggered into action by means of an interrupt, which will in turn switch on the CC1000 radio to receive the data message that is following shortly.

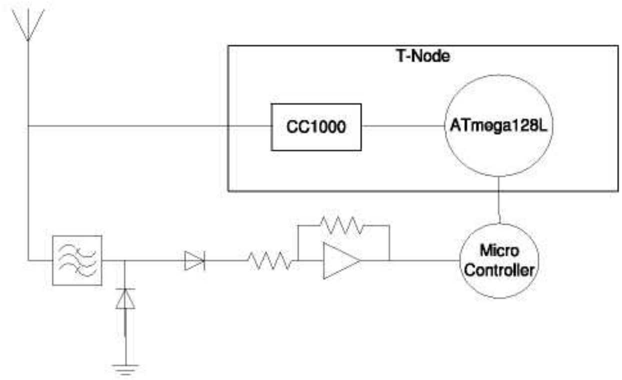
Although using a separate microcontroller may seem overkill for detecting a wakeup signal – a simple comparator would do in principle – we deliberately included it in the design to provide the flexibility that any experimental prototype needs. For example, it would allow us to rectify potential flaws in the analog part of the receiver, and to experiment with unicast support.

3.3 The analog domain

The Chipcon CC1000 radio is used for generating a wakeup signal; the analog circuitry in Figure 1 is used for decoding the signal. To keep the analog part as simple as possible, we use On-Off Keying (OOK), as supported by the CC1000, with a symbol rate of 862 Hz (see Subsection 3.4). An added benefit of using OOK modulation for the wakeup signal is that it radically differs from normal communication using Frequency Shift Keying (FSK). This makes it easy to distinguish between the two with FSK generating a signal with a fixed amplitude, and OOK generating a changing sequence. Thus self-interference can easily be filtered out in the digital domain (see below).

The external interference, from GSM signals and other wireless equipment, must be filtered out in the analog domain, hence, the frequency filter taking in the signals from the antenna. It is the only complex component that has to support 868 MHz signals and has to be very narrow band so that signals in the 880–950 MHz range do not pass through unattenuated. This is quite a challenge as GSM transmitters may use up to 2 W of send power, while the maximum output power of a T-node is about 3 mW. Another constraint on the frequency filter is that it must use very little power. We selected the rather expensive, but completely passive EPCOS 868.30 MHz filter. It costs €2.51, or 50% of the wakeup radio budget, consumes no power and attenuates GSM signals by as much as 30 dB. In practice this means that GSM signals can only interfere when the GSM transmitter is closer than the T-node sending the wakeup signal.

Figure 1 T-node platform with wakeup radio circuitry



A wakeup signal transmitted at 3 mW from 10 m away is attenuated to a level of –51 dBm, which is equivalent to 10 μV peak to peak on the antenna line. After an attenuation of 3.5 dB caused by the frequency filter, this reduces to a mere 5 μV. For proper detection by a microcontroller, the signal must be amplified. In the ideal case, the output of the amplifier is directly connected to an (edge-triggered) interrupt pin, which requires a 0.7 V peak-to-peak signal. Therefore, the signal leaving the frequency filter must be amplified by at least a factor of 140,000 to reach the required 10 m range. Such a high amplification of an 868 MHz signal is impossible within our power budget of 150 μW, as the required current-feedback operational amplifiers generally consume hundreds of milliwatts at best. Therefore, the signal is first converted down to the original OOK frequency by means of the two diodes shown in Figure 1. This costs no power and significantly reduces the gain-bandwidth requirements on the amplification circuit.

To amplify the 862 Hz baseband signal from 5 μV to 0.7 V, we still need an amplification factor of 140,000. This could be realised by a single operational amplifier, which would require a gain-bandwidth product of at least 862 Hz × 140,000 = 120 MHz. Such amplifiers exist but still consume power in the order of 100 mW, rendering them impractical for use in a low-power wakeup radio. Therefore, we resort to a multistage design consisting of four low-power, low-gain amplifiers. Cascading amplifiers comes at the price of a slight loss in efficiency. The four-stage feedback amplifier used in the wakeup radio circuit results in a loss of 6 dB, which must be compensated for. Hence, the total amplification must be $4 \times 140{,}000 = 560{,}000 = 27.36^4$. The amplification factor of each stage is determined by a pair of resistors, which are set to 330 kΩ and 12 kΩ (yielding a factor of 27.5). We selected the National LPV358 operational amplifier, because it provides ample bandwidth (152 kHz) and consumes only 24 μW. This brings the total power consumption of the analog part of the wakeup radio to 96 μW.

3.4 The digital domain

The primary task of the microcontroller in the wakeup radio is to filter out residual interference left by the analog part, to detect a wakeup signal and to notify the AT-mega128L processor of the T-node about the signal by raising an interrupt. It is important to make these tasks as simple as possible so we can use a low-power microcontroller; the power budget has shrunken to about 50 μW, because the analog part consumes about 100 μW out of the total 150 μW.

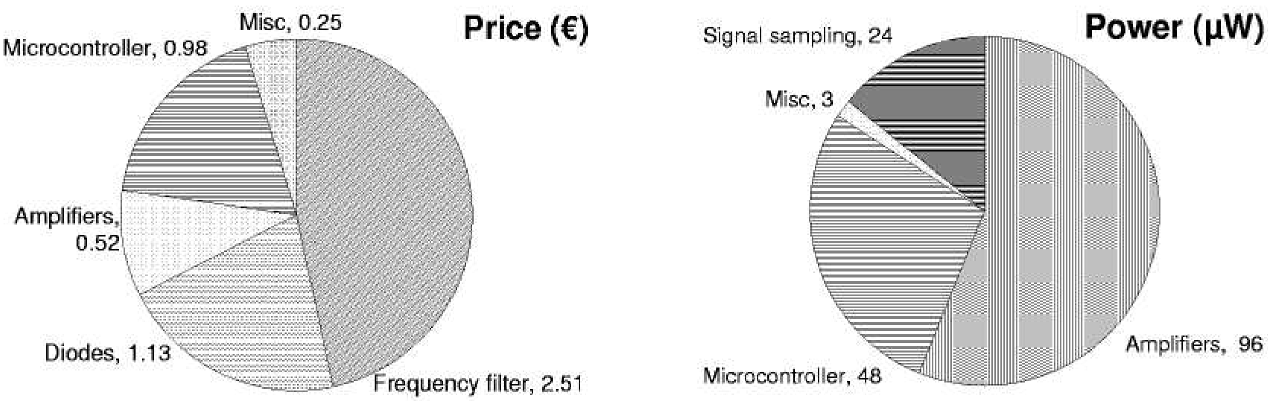
As argued above, the main source of interference is GSM equipment transmitting in the 900 MHz band. GSM also uses OOK modulation, but at a rather low frequency of 200 Hz. We determined that a CC1000 radio can switch at a maximum frequency of about 900 Hz, setting it enough apart to allow for easy discrimination. To maximise effectiveness, we set the CC1000 to use a frequency of 862 Hz, which is almost relatively prime (sharing only a factor of 2) and can be generated from the T-node's on-board 32 KHz crystal (using a divider of 38).

We selected the PIC12F683 microcontroller to do the wakeup signal detection, because it only consumes 48 μW when running at 32 KHz, its lowest operating frequency. This microcontroller features a built-in comparator as well as 10-bit AD converter, which both can be hooked up to the output of the analog part. Detecting the 862 Hz signal when running at 32 KHz leaves little room for advanced processing; each instruction takes four clock cycles, leaving just nine instructions per data sample. This however, is enough to count the number of transitions between high- and low-input levels (or vice versa) within a fixed period; when this count exceeds a threshold, we assume that a wakeup call was present. In particular, the microcontroller is instructed to repeatedly draw seven samples (at 862 Hz) and raise an interrupt when the number of transitions exceeds 3. To limit latency on the one hand, and counter aliasing on the other hand, the wakeup signal consists of ten transitions ensuring a high chance of being detected. The simple counting-transitions policy has proven to be quite effective (see Section 6).

3.5 Overall specifications

Now that we discussed both the analog and digital part of the wakeup radio design, we can put the pieces together and provide the overall specifications. Figure 2 provides a breakdown both for the price and the power consumption. The total price amounts to €5.39, which slightly exceeds the €5 target. The frequency filter is by far the most expensive component, but is essential in filtering out most of the ambient noise. The total power consumption also exceeds its target slightly and amounts to 171 μW. This is mainly due to the microcontroller consuming additional power when operating the AD converter. The ADC draws 3 mW during the sample acquisition period of 5 μs and 150 μW during the digitalisation period of 60 μs. These costs are incurred for every sample, averaging out to a total of 24 μW at 1000 Hz. Note that 57% of the power budget is spent in the analog part of the design on amplifying the baseband signal from 5 μV to 0.7 V. The sensitivity of the AD converter is 5 mV, showing that there is room for improvement, for example, by leaving out one amplifier stage saving 24 μW. All in all, the design of the wakeup radio meets its requirements rather well.

Figure 2 Cost breakdown for price (€5.39) and power consumption (171 mW) of the wakeup circuitry



4 Performance analysis

Wakeup radio is good at combining low latency with low energy consumption for low data rate traffic.

In this section, we analyse how well wakeup radio stands up to state-of-the-art energy-efficient MAC protocols. We compare against *B-MAC* (Polastre et al., 2004), the standard protocol that ships with TinyOS, *WiseMAC* (El-Hoiydi and Decotignie, 2004), which improves on B-MAC by maintaining poll schedules, and *SCP-MAC* (Ye et al., 2006) designed for extremely low data rates, making it a challenging protocol to beat. Since the purpose of this performance comparison is to determine the feasibility of our wakeup-radio design, we revert to a mathematical analysis using simple models capturing first-order effects only.

Table 1 lists the main parameters used in the performance analysis. The hardware-related parameter values are based on the standard T-node platform (measured), and the design of the wakeup circuitry outlined in the previous section (taken from data sheets). The number of neighbours (topology information) and data rate (application specific) will be varied to study the crossover points of wakeup radio and MAC protocols. The performance metrics of interest are power (energy consumption) and latency; we do not model throughput since that is not a critical resource, or otherwise switching the radio off would not be a possibility in the first place.

Table 1 Parameters used in the performance analysis; typical values are taken from the T-node platform featuring a CC1000 radio and wakeup circuitry

Symbol	*Description*	*Value*
P_{sleep}	Power when in sleep mode	600 μW
P_{RX}	Power when receiving	45 mW
P_{TX}	Power when transmitting	60 mW
P_{cs}	Power when performing a carrier sense	15 mW
T_{cs}	Time for performing a carrier sense	2.5 ms
T_{byte}	Time for sending/receiving 1 byte	416 μs
P_{wu}	Power when in wakeup mode	171 μW
F_{wu}	Frequency of the wakeup signal	862 Hz
N	Number of neighbours	Variable
F_{msg}	Frequency of message sends	Variable
T_{msg}	Time needed for sending a msg + ACK	21 ms
T_{hdr}	Time needed for sending just a header	7 ms

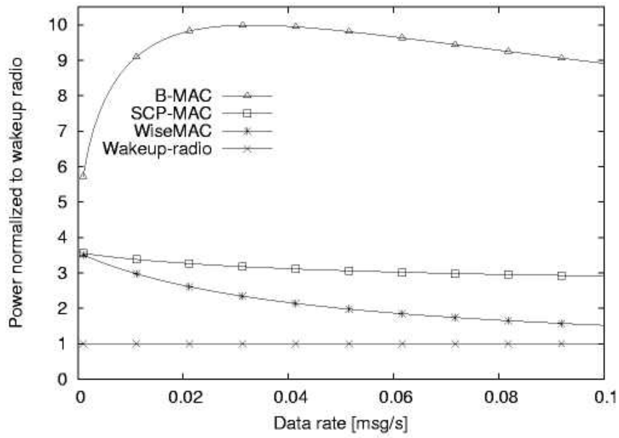
The performance of the wakeup radio is rather easy to model. The latency of a (one-hop) message transfer consists of the time needed to send the wakeup signal (T_{tone}), and the actual transfer time over the primary radio (T_{msg}). This transfer time includes waking up the radio, receiving the DATA message (including headers) and sending back an acknowledge frame. For simplicity, we do not consider transmission errors and collisions, hence, we do not model retransmissions.

$$\begin{aligned} T_{tone} &= \frac{10}{F_{wu}} \\ L_{wu} &= T_{tone} + T_{msg} \end{aligned} \quad (1)$$

$$\begin{aligned} P_{wu} = P_{wu} &+ F_{msg} \cdot ((T_{tone} + T_{msg}) \cdot P_{TX} \\ &+ T_{msg} \cdot P_{RX} + (N-1) \cdot T_{hdr} \cdot P_{RX}). \end{aligned} \quad (2)$$

The average power consumed by a node depends on the frequency at which messages are sent through the network, denoted by F_{msg}. Each message transfer adds energy to the basic costs of the wakeup circuitry (P_{wu}). The CC1000 radio is used for sending both the wakeup tone (for a duration of T_{tone}) and the DATA/ACK sequence (for a total duration of T_{msg}). Receiving the message also takes T_{msg} time. The final component in equation (2) models neighbours overhearing the message transfer; as an optimisation they switch off the CC1000 radio after the MAC header has been decoded (address filtering).

Figure 3 Normalised power consumption; unconstrained latency

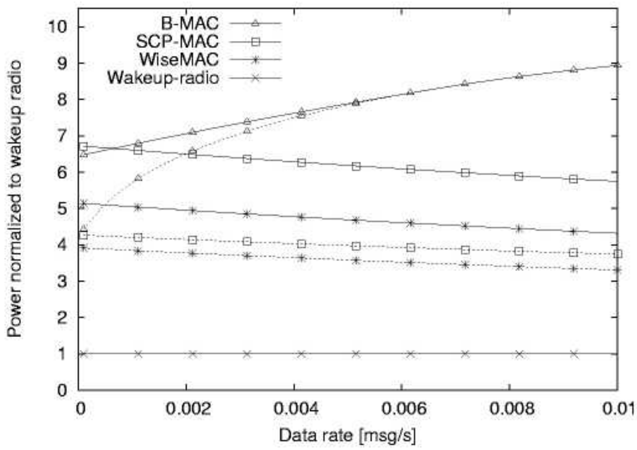


In a similar spirit, we have modelled the latency and energy consumption of the three reference MAC protocols especially designed for WSNs. Please refer to Appendix A for the models of B-MAC, WiseMAC and SCP-MAC. With these models it becomes possible to explore the relative performance of wakeup radio with respect to a number of scenarios. In particular we are interested in the effect of the rate at which messages flow through the network (F_{msg}). For now we fix the number of neighbours to ten for the primary radio and to five for the wakeup radio, reflecting the difference in range between the two. Figure 3 shows the average power consumption of the wakeup radio and the three MAC protocols over a wide traffic range; the consumption plots have been normalised against the wakeup radio (at 1.0) to visualise to basic gain in lifetime the wakeup radio achieves over the three MAC protocols.

Several observations can be made. First, the wakeup radio outperforms all MAC protocols due to its low power consumption in receive mode (171 μW) vs. that of a T-node in sleep mode (600 μW). Second, B-MAC performs worst of all, which was to be expected given that it is the oldest of the three. Third, WiseMAC does better than SCP-MAC, with the difference growing for larger data rates. This is caused by SCP-MAC incurring more overhearing overhead than WiseMAC due to the synchronised channel polling. Finally, we like to point out that the gains of the wakeup radio are relatively small, because we did not take latency into account.

Figure 4 shows the true benefit of the wakeup radio by constraining the latency and zooms in on the low data rate range. Two sets of curves are included. The first set shows the effect when the maximum latency of the MAC protocols is set to ten times that of the wakeup radio (237 ms = 10 × 23.7 ms). This accounts for the shorter range of the wakeup radio, causing messages to the sink to travel along paths with many more hops than with the CC1000 radio operating at maximum send power. Note that the wakeup radio now achieves gains about twice as high as when the MAC protocols could select arbitrarily large sleep intervals (except B-MAC, which uses its optimal interval). Also the relative performance between the three MAC protocols has changed considerably. B-MAC is still the worst, but the gap with especially SCP-MAC is less dramatic. The difference between SCP-MAC and WiseMAC has significantly increased due to SCP-MAC's slotted nature delaying each hop by the duration of a complete sleep interval, while WiseMAC will find its next neighbour to wake up, on average, after half an interval.

Figure 4 Normalised power consumption; latency constrained by 237 ms (solid lines) and 1 s (dashed lines)



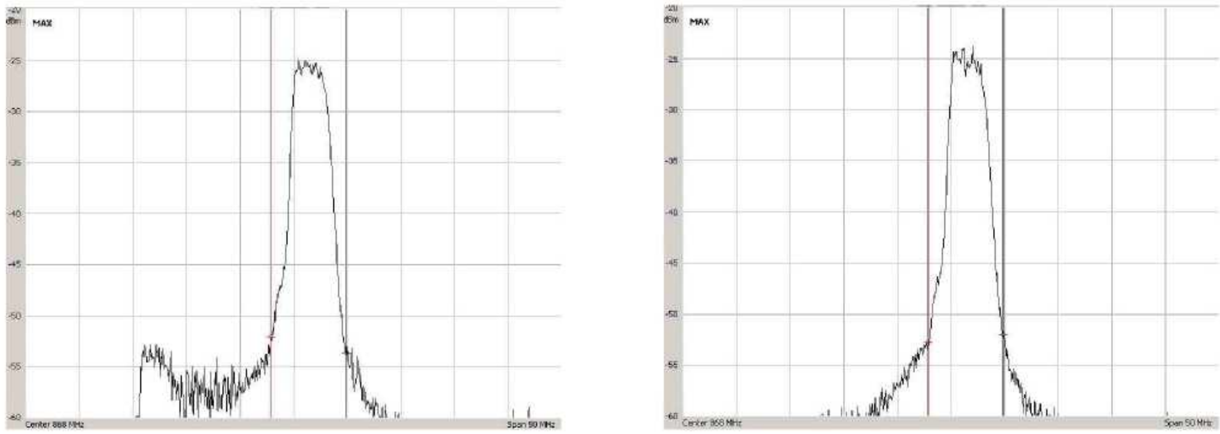
The second set of curves in Figure 4 shows the performance when the (one-hop) latency is relaxed to 1 s. This delay seems reasonable when considering scenarios in which data is harvested on the fly. Especially SCP-MAC benefits from this relaxation, but the wakeup radio still outperforms all protocols by nearly a factor of 4. This is quite an achievement given that SCP-MAC and WiseMAC have been well engineered to operate at low power consumption levels.

5 Implementation

We will now detail some implementation issues that arose while taking the wakeup radio from its design to a working prototype. The first issue that arose was on the sender side, where we use the CC1000 radio to generate the wakeup signal using OOK. We had a choice of two options for generating this 862 Hz OOK signal. The first option is to switch the radio on and off completely. The second option is to control the send power (amplitude modulation) and switch between minimum and maximum level. We experimented with both alternatives and Figure 5 shows two snapshots obtained with a Rohde & Schwarz FS315 spectrum analyser from a series of measurements. Note that the spectrum in the case of on-off switching contains some noise in the 10 MHz sideband directly below the 868 MHz carrier, which is not present when switching the output power between high and low. Since this noise will not be attenuated much by the frequency filter centred at the carrier, it might cause distortions in the baseband signal and we decided to go for modulating the send power of the CC1000 radio.

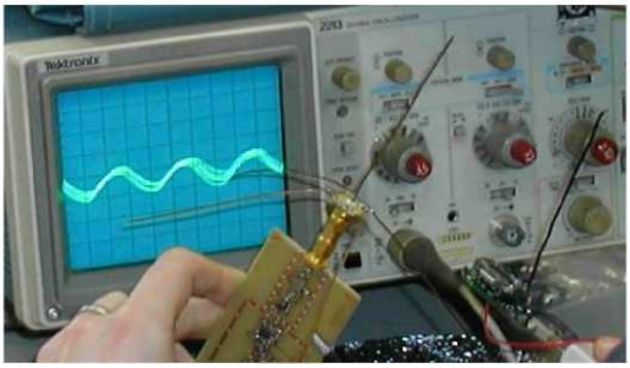
The next step in the implementation process was to design and fabricate a printed circuit board for the receive circuitry. For practical reasons, we decided to equip this board with a separate whip antenna, and leave connecting to the T-node antenna for a future revision. We encountered some serious problems in the analog part of the wakeup circuitry. In particular, we had problems in getting the four-stage amplifier cascade to work, despite the validation of its design through running a SPICE circuit simulation. After trying out several types of operational amplifiers (op-amps) and experimenting with different gains (controlled through resistors), we had to settle for a three-stage cascade losing about a factor of 4 in amplification compared to the reference design. The consequence is that the range of the wakeup radio prototype is limited to about 2 m instead of the 10 m target. Fortunately, the quality of the signal after amplification is good, as can been seen in Figure 6 displaying a screen shot from a Tektronix 2213 oscilloscope.

Figure 5 Spectrum analyser plots of wakeup signals in the 868 MHz band



(862Hz on-off switching) (862Hz send power modulation)

Figure 6 Wakeup radio on oscilloscope (see online version for colours)



Decoding the smooth and regular signal coming out of the amplifier is the job of the PIC12F683 microcontroller running in its ultra low-power mode at 32 KHz. For maximum flexibility the 862 Hz wakeup signal is (over-)sampled through the on-chip AD converter at 1000 Hz. As mentioned in Subsection 3.4, the low processor speed only allows for the execution of nine instructions per sample. The signal detection procedure is outlined as pseudo-instructions below:

```
1  Start AD conversion
2  Is AD conversion ready?
3  If not, retry 2
4  Subtract new sample from previous
5  Complement the result if negative
6  Compare the result to the threshold
7  If larger, increment signal changes
8  Store the new sample for next round.
```

Note that the code fragment contains no explicit timing; it follows simply from the time the microcontroller takes to execute these eight instructions inlined back to back. As it turned out, the AD conversion process (instructions 1 and 2) takes almost 1 ms to complete, or about four times as long as expected. Part of the reason is that it writes information to four different registers, one after the other. The result is that samples are drawn at an effective rate of only 570 Hz (once every 14 instructions). The mismatch between the sampling rate and the frequency of the wakeup signal can be solved in two ways. The clock rate of the Programmable Intelligent Computer (PIC) can be increased to the next level at 125 KHz, or the frequency of the wakeup signal can be lowered (say to 431 Hz). Both options have their down side. When increasing the clock frequency, the power consumption increases from 48 μW to 600 μW, tying that of the sleep mode of the T-node. When lowering the frequency of the wakeup signal it becomes much more prone to interference from GSM signals with an OOK modulation frequency of 200 Hz. Since the analog part is not functioning optimally, we decided not to take the risk of picking up additional interference and will run the PIC at 125 KHz. The promising alternative of using an *external* comparator (drawing just 20–30 μW) connected to one of the I/O pins of the PIC is left for future work; it would be very power-efficient, but require careful tuning of the (fixed) threshold discriminating between high (on) and low (off) levels. Note that the PIC also contains an *internal* comparator, but operating it consumes at least 90 μW due to the need to keep it on continuously because of its long setup times; this makes it a rather unattractive alternative.

6 Experiments

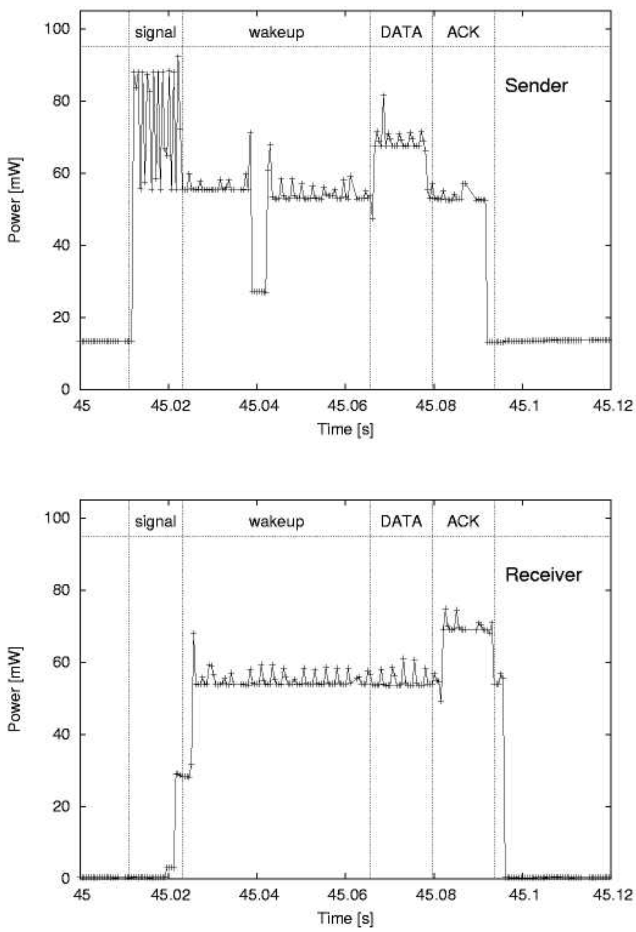
With our working prototype wakeup radio we performed a number of tests to determine its performance. These tests were performed in our lab, an ordinary office building with the usual set of WLAN networks, GSM cell phones and DECT phones in operation, making for a challenging environment. In a first test, the receiver would simply toggle one of the T-node's Light-Emitting Diodes (LEDs) when detecting a wakeup signal. This allowed us to determine the maximum range (3 m) at which the receiver could detect wakeup signals. At that distance, however, many wakeups would be missed, so in order to avoid retransmissions one should limit the range to about 2 m.

In a second test, we let the receiver run without sending any wakeup signal to determine its sensitivity to ambient noise and interference. Much to our surprise the LED would not toggle for hours, once the analog circuit stabilised after about 50 ms. This shows that the wakeup circuitry is very effective at blocking interference from other wireless equipment. Even making a telephone call by GSM from within a 1 m range did not trigger a false alarm.

Next, we upgraded the software to do a series of communications, each involving the transmission of a 10 ms wakeup signal, followed by a 15 ms period to activate the T-node at the receiving end, the transmission of a DATA packet over the CC1000 radio, and a closing ACK by the receiving T-node. If an acknowledgement was not received by the sender within 30 ms, it would retry sending the DATA message up to five times. This measure was taken to counter an artefact of using two antennas mounted perpendicular to each other on the receiver side (*R*), which caused the T-node antenna to be misaligned with respect to the sender (*S*) antenna resulting in rather poor reception over the CC1000 backchannel (from *R* to *S*). Each ACK received back at sender *S*, however, implies that the wakeup signal was received in order at *R*, allowing us to estimate the number of missed wakeups rather accurately. At a range of 1 m, about 80–90% of the wakeup signals are indeed received at *R*; the exact fraction was difficult to determine because the sender node had to be positioned manually for each run.

Finally, we hooked the sender and receiver nodes up to our testbed infrastructure capable of logging power consumption with a resolution of 5 kHz of up to 24 T-nodes operating in parallel. Figure 7 shows an excerpt of two such traces, capturing a complete wakeup-DATA-ACK sequence. At time t = 45.011 s, sender *S* starts sending the wakeup signal, toggling between maximum and minimum send power causing its power consumption to fluctuate between 55.2 and 87.9 mW. After 12 ms (at t = 45.023 s), *S* stops sending the wakeup signal, and switches its radio from OOK modulation (wakeup mode) into FSK modulation (normal mode). During this switch the radio is turned off, accounting for the dip in power consumption around t = 45.040 s. At t = 45.066 s, *S* sends out a DATA message, which takes 13 ms, and switches to RX mode to receive the ACK. Finally at t = 45.094, *S* can switch off the radio and prepare for the next communication sequence. Note that the power consumption of *S* does not drop to zero in between communication actions, because we did not bother putting the processor into deep sleep as we do for the receiver *R*. The total duration of the wakeup-DATA-ACK sequence is 82 ms, which is rather large compared to the actual data transfer time of 28 ms. We are confident, however, that the time spent in signalling and waking up can be considerably shortened as we have been using conservative timings in this experimental phase of development. For example, the receive trace shows that the wakeup signal is actually detected some milliseconds before the end, at t = 45.020 s when *R*'s power consumption rises to about 55.2 mW. Since the precision of the power consumption tracer is in the order of 200 μW, we performed a separate measurement of the power consumed by the receiver in wakeup mode (i.e. with its T-node switched off) using a multimeter. We measured the receive circuitry to consume 801 μW, which is well in line with the theoretical 723 μW derived from the data sheets.

Figure 7 Power traces from sender (top) and receiver (bottom) exchanging one wakeup-DATA-ACK sequence



7 Conclusions

In many WSN scenarios, battery-powered nodes must operate for years, which necessitate the need for advanced power management of the radio. WSN-specific MAC protocols are capable of duty cycling the radio at very low rates, but only at the expense of introducing long sleep intervals. Wakeup radio on the other hand promises low latency *and* low energy consumption through the use of a secondary, ultra low-power radio. This was confirmed by an analytical comparison with respect to three state-of-the-art MAC protocols for WSNs.

In this paper, we presented the first working prototype wakeup radio for use in sensor networks. It was designed to leverage as much of the primary (CC1000) radio as possible. In particular, the antenna can be shared, and the send path is reused; only the receive path requires additional hardware, but by using standard components the cost is limited to about €5. The wakeup radio operates in the European license-free 868 MHz band, and includes special measures to avoid interference from GSM signals in the nearby 900 MHz band. To keep the design as simple as possible the wakeup signal is modulated using OOK (at 862 Hz), which allows for straightforward decoding (i.e. counting pulses) by a dedicated PIC microcontroller.

When implementing the design into a working prototype, we faced two problems that compromised performance. The first problem was in the analog part, where a reduction in the amplification factor caused a range limitation to about 2 m vs. the 10 m design target. The second problem was the PIC microcontroller not being able to obtain the ADC readings fast enough, forcing it to be run at a higher clock speed. This caused the power consumption of the PIC to jump from 48 μW to 600 μW (and the overall power consumption from 171 μW to 819 μW). Experiments with the prototype wakeup radio showed that it is insensitive to noise and interference, and did not raise any false alarms. The flip side is that some of the wakeup signals are missed; at a range of 1 m about 80–90% of all messages (wakeup-DATA-ACK) are successfully transferred, with a latency of 80 ms.

In conclusion, the current prototype is a firm, first step in developing a practical low-cost wakeup radio. The next steps are to address the limited range and high power consumption, and to test the performance in real-life scenarios. For now, we anticipate that with a few (hardware) alterations to the design it will be possible to reduce the power down to 200 μW, if not lower, which would provide the second-generation implementation the desired four-fold gain in lifetime over advanced MAC protocols duty cycling the main radio.

Acknowledgements

We would like to thank Gertjan Halkes for proofreading the draft version of this paper, and for scripting gnuplot and postscript to our needs.

References

da Silva Jr., J., Shamberger, J., Ammer, M., Guo, C., Li, S., Shah, R., Tuan, T., Sheets, M., Rabaey, J., Nikolic, B., Sangiovanni-Vincentelli, A. and Wright, P. (2001) 'Design methodology for PicoRadio networks', *Design, Automation and Test in Europe (DATE)*, Munich, Germany, pp.314–325.

El-Hoiydi, A. and Decotignie, J-D. (2004) 'WiseMAC: an ultra low power MAC protocol for multi-hop wireless sensor networks', *First International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004), LNCS 3121*, Springer-Verlag, pp.18–31.

Gu, L. and Stankovic, J. (2005) 'Radio-triggered wake-up for wireless sensor networks', *Real-Time Systems*, Vol. 29, No. 2, pp.157–182.

Langendoen, K. (2008) 'Medium access control in wireless sensor networks', in Wu, H. and Pan, Y. (Eds): *Medium Access Control in Wireless Networks, Volume II: Practice and Standards*, Nova Science Publishers, Inc., pp.535–560.

Miller, M. and Vaidya, N. (2005) 'A MAC protocol to reduce sensor network energy consumption using a wakeup radio', *IEEE Transactions on Mobile Computing*, Vol. 4, No. 3, pp.228–242.

Pletcher, N. (2004) *Micro Power Radio Frequency Oscillator Design*, Master's Thesis, UC, Berkeley.

Polastre, J., Hill, J. and Culler, D. (2004) 'Versatile low power media access for wireless sensor networks', *2nd ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*, Baltimore, MD, USA, pp.95–107.

Rajendran, V., Obraczka, K. and Garcia-Luna-Aceves, J. (2003) 'Energy-efficient, collision-free medium access control for wireless sensor networks', *1st ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, Los Angeles, CA, USA, pp.181–192.

Schurgers, C., Tsiatsis, V., Ganeriwal, S. and Srivastava, M. (2002) 'Optimizing sensor networks in the energy-latency-density design space', *IEEE Transactions on Mobile Computing*, Vol. 1, No. 1, pp.70–80.

Shih, E., Bahl, P. and Sinclair, M. (2002) 'Wake on wireless: an event driven energy saving strategy for battery operated devices', *8th ACM International Conference on Mobile Computing and Networking (Mo-biCom02)*, Atlanta, GA, USA, pp.160–171.

van Hoesel, L. and Havinga, P. (2004) 'A lightweight medium access protocol (LMAC) for wireless sensor networks', *1st International Workshop on Networked Sensing Systems (INSS 2004)*, Tokyo, Japan.

Ye, W., Silva, F. and Heidemann, J. (2006) 'Ultra-low duty cycle MAC with scheduled channel polling', *4th ACM Conference on Embedded Networked Sensor Systems (SenSys 2006)*, Boulder, CO, USA, pp.321–334.

Appendix A: MAC models

In this appendix, we detail the performance models of the following three WSN-specific MAC protocols: B-MAC, WiseMAC and SCP-MAC. The models refer to parameters introduced in Table 1.

B-MAC uses Low-Power Listening to reduce energy consumption. Nodes regularly wake up to check whether the channel is clear or not. If clear, they return to sleep immediately. Otherwise, nodes switch to receive mode listening for the next message to arrive. Senders prepend each message with a preamble that covers B-MAC's sleep interval ($T_{\text{sleep}}^{\text{B-MAC}}$) to ensure that the receiver, regardless of when it wakes up, will detect the preamble and wait for the subsequent message to arrive. The consequence of stretching preambles is that latency increases proportionally:

$$L_{\text{B-MAC}} = T_{\text{sleep}}^{\text{B-MAC}} + T_{\text{msg}} \tag{3}$$

$$\begin{aligned} P_{\text{B-MAC}} = {} & P_{\text{sleep}} + T_{\text{cs}} / T_{\text{sleep}}^{\text{B-MAC}} \cdot P_{\text{RX}} \\ & + F_{\text{msg}} \cdot ((T_{\text{sleep}}^{\text{B-MAC}} + T_{\text{msg}}) \cdot P_{\text{TX}} \\ & \quad + (T_{\text{sleep}}^{\text{B-MAC}} / 2 + T_{\text{msg}}) \cdot P_{\text{RX}} \\ & \quad + (N-1) \cdot (T_{\text{sleep}}^{\text{B-MAC}} / 2 + T_{\text{hdr}}) \cdot P_{\text{RX}}). \end{aligned} \tag{4}$$

The sleep interval also has a significant effect on the power consumption of B-MAC. First, it sets the baseline at the power consumed in the radio's sleep state (P_{sleep}). Second, it determines the relative importance of the carrier sense each interval ($T_{\text{cs}} / T_{\text{sleep}}^{\text{B-MAC}}$). Third, it determines the overhead associated with each message transfer. A sender must transmit the preamble for the full length of the sleep interval, while the receiver and other neighbours listen on average for half of the interval before the start of the message is detected. Neighbours shut down their radio as soon as the MAC header is decoded.

Increasing the sleep interval reduces the polling costs, but increases the message overheads showing that there exists an optimum interval. This optimum can be determined by taking the derivative of equation (4) with respect to $T_{\text{sleep}}^{\text{B-MAC}}$, and setting that to zero:

$$-T_{\text{cs}} \cdot P_{\text{RX}} / {T_{\text{sleep}}^{\text{B-MAC}}}^2 + F_{\text{msg}} (P_{\text{TX}} + N/2 \cdot P_{\text{RX}}) = 0$$

This leads to the following best-case setting for B-MAC:

$$T_{\text{sleep}}^{\text{B-MAC}} = \sqrt{\frac{T_{\text{cs}} \cdot P_{\text{RX}}}{F_{\text{msg}} (P_{\text{TX}} + N/2 \cdot P_{\text{RX}})}}. \tag{5}$$

WiseMAC (WM for short) improves on B-MAC by keeping track of neighbour schedules, which allows a sender to wait until just before the receiver polls the channel, and then to transmit its message with a normal preamble. When communication is infrequent, WiseMAC compensates for possible clock drift and stretches the preamble accordingly. In the equations below, we do not include this compensation since we determined that the impact for a typical clock drift of 30 ppm is neglectable:

$$L_{\text{WM}} = T_{\text{sleep}}^{\text{WM}} / 2 + T_{\text{msg}} \tag{6}$$

$$\begin{aligned} P_{\text{WM}} = {} & P_{\text{sleep}} + T_{\text{cs}} / T_{\text{sleep}}^{\text{WM}} \cdot P_{\text{RX}} \\ & + F_{\text{msg}} \cdot (T_{\text{msg}} \cdot P_{\text{TX}} + T_{\text{msg}} \cdot P_{\text{RX}} \\ & \quad + (N-1) \cdot T_{\text{msg}} / T_{\text{sleep}} \cdot T_{\text{hdr}} \cdot P_{\text{RX}}). \end{aligned} \tag{7}$$

Note that the overhearing term includes an additional factor $T_{\text{msg}}/T_{\text{sleep}}$, which accounts for the fact that nodes only overhear messages when they happen to poll by chance during an ongoing transmission; this probability reduces when the sleep interval increases. In contrast to B-MAC, there is no optimal sleep interval, because the message transfer no longer depends on it (cf. equations 4 and 7). This does not, however, imply that the sleep interval can be stretched to arbitrary length, because nodes must still wakeup at least once every $1/F_{\text{msg}}$ seconds to handle the data generated by the application.

The final protocol that we consider is SCP-MAC. In contrast to the two previous protocols, nodes are synchronised on a global schedule and wakeup collectively to perform a carrier sense (poll) at the beginning of each slot. By orchestrating senders to contend for channel access prior to a poll, SCP-MAC can operate very efficiently for low traffic loads.

$$L_{\text{SCP}} = T_{\text{sleep}}^{\text{SCP}} + T_{\text{msg}} \tag{8}$$

$$\begin{aligned} P_{\text{SCP}} = {} & P_{\text{sleep}} + T_{\text{cs}} / T_{\text{sleep}}^{\text{SCP}} \cdot P_{\text{RX}} \\ & + F_{\text{msg}} \cdot (T_{\text{msg}} \cdot P_{\text{TX}} + T_{\text{msg}} \cdot P_{\text{RX}} \\ & \quad + (N-1) \cdot T_{\text{hdr}} \cdot P_{\text{RX}}). \end{aligned} \tag{9}$$

To keep the network synchronised, that is to compensate for clock drift, SCP-MAC demands a minimum traffic load. Again we can ignore this factor because of the rather high precision (30 ppm) of the crystals used in typical node hardware. We also do not model sender channel-access resolution since SCP-MAC uses a very short window compared to the actual transfer time T_{msg}. Note that with SCP-MAC the chance of overhearing is back to 1 due to all nodes listening at the same time.