Frame Preamble MAC for Multihop Wireless Sensor Networks: Design and Implementation

A. Bachir, S. Plancoulaine, D. Barthel, M. Heusse, A.Duda

Abstract

MAC protocols based on preamble sampling techniques offer significant energy savings for low data-rate multihop wireless sensor networks by efficiently reducing idle listening in lightly loaded networks. However, without any particular optimization, preamble sampling protocols still consume large amounts of energy due to the overhead induced by the long preamble transmitted prior data frames. To reduce the overhead induced by the use a full length preamble, many improvements have been proposed. These proposals targeted both the cost at the receiver and that at the transmitter. At the transmitter side, proposals tends to minimize the preamble size by means of explicit synchronization, whereas at the receiver, proposals aim at reducing the duration of listening by replacing the preamble with a series of frames.

In this paper, we focus on reducing the cost at the receiver side while showing that our solution is also compatible with solutions that reduce the cost at the transmitter side. We propose a new idea that reduces overhearing further on by avoiding reception of irrelevant frames. With our idea that is based on digest information, not only unicast frames can be filtered but also irrelevant (redundant) broadcast frames can be avoided a priori. We show that the use of the digest information together with the sequence number field substantially increases the energy savings of frame preamble MAC protocols. We propose two possible implementations, Micro Frame Preamble (MFP) and Zebra Frame Preamble (ZFP), in which avoiding reception of irrelevant broadcasts can be integrated. We provide an analytical framework, in which we model the lifetime of various frame preamble MAC protocols. We complete the theoretical analysis with ns2 simulation and experimental validation through implementation on the Chipcon CC 2500 evaluation kit. We analytically show that the solution based on micro frames yields

Abdelmalik Bachir, Martin Heusse and Andrzej Duda are with LIG Laboratory. BP 72, 38402 St. Martin Cedex, France. E-mails: {abdelmalik.bachir, martin.heusse, andrzej.duda}@imag.fr

Sylvain Plancoulaine and Dominique Barthel are with France Telecom R&D Division, 28 chemin du Vieux Chêne, BP98 38 243 Meylan Cedex, France. E-mails: {sylvain.plancoulaine, dominique.barthel}@orange-ftgroup.com

better performance. We discuss implementation choices and effect of power consumption of the micro controller on the overall performance of the solution. We also discuss potential improvements from using new architectures in which hardware circuitry can be used for constructing and transmitting preamble frames.

Index Terms

Sensor Networks, Energy-Efficient MAC protocols.

I. INTRODUCTION

One of the main objectives of MAC (Medium Access Control) protocols for wireless sensor networks is to minimize power consumption while providing reliable low-rate data transmission. Previous studies have identified multiples sources of energy dissipation at the MAC layer, the most important being the following [1]:

- *Idle Listening*: it happens when a node does not know when it will be the receiver of a frame, so it keeps its radio on while listening to the channel waiting for potential data frames. A node may waste considerable energy when the radio is on, even when it is neither receiving nor transmitting data [2]–[4].
- *Collisions*: they mainly concern CSMA-based MAC protocols. They may happen when a receiver is within the transmission range of two or more transmitters that are simultaneously transmitting so that the receiver does not capture any frame. When a collision happens, the energy spent in the transmission and reception of the collided frames is simply wasted.
- *Overhearing*: it happens when a sensor node receives and decodes an irrelevant frame. Such frames may be destined to other nodes, or the node has already received them. We studied both cases and proposed techniques that can be used by MAC protocols to avoid overhearing [5], [6].
- *Protocol Overhead*: it may have several different origins such as the energy drained in computing and transmitting control frames. For example, RTS and CTS control frames used in some protocols do not carry any application layer data although their transmission consumes energy.

Applications of sensor networks usually generate low traffic loads so that the communication channel is expected to be idle most of the time. In this case, idle listening is the most significant cause of energy dissipation: without any specific energy management, nodes waste considerable amounts of energy, because they keep their radios on for long time intervals while listening to an idle channel. To mitigate idle listening in energy-efficient MAC protocols for sensor networks, nodes need to sleep for long periods of time instead of being permanently active. Such MAC protocols define a *duty-cycle* parameter to control the ratio of the activity to the sleep time [7], [8].

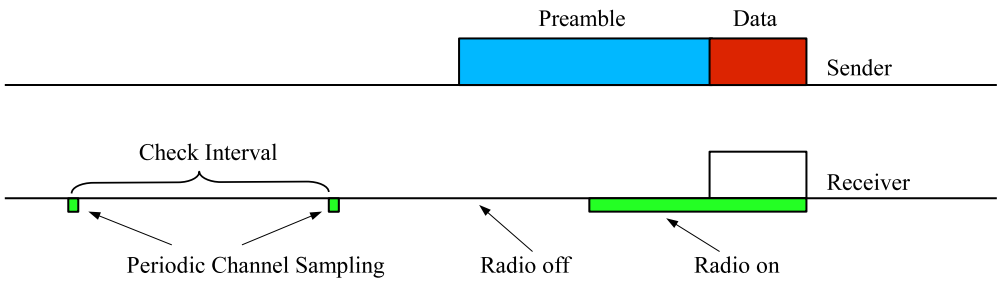


Fig. 1. Preamble sampling technique

Duty cycles of 1% may substantially reduce energy consumption. However, if a node wakes up only at some predetermined instants to limit the effect of idle listening, then it needs a means for detecting transmissions of other nodes. There are two ways for making this possible. In the first approach used in protocols like SMAC [1], TMAC [9], and others, nodes synchronize on a common sleep/wakeup schedule by exchanging synchronization messages. The second approach, used in protocols like WiseMAC [10] and B-MAC [11], does not setup a common sleep/wakeup schedule to avoid the synchronization overhead and further reduce idle listening in lightly loaded networks. Instead, each node chooses its own sleep/wakeup schedule independently of the others and transmits a preamble before each data frame. The preamble is long enough to make sure that all potential receivers will get their data. In this paper, we focus on the protocol family following the latter approach and we propose a method for further improving energy savings. We can find in the literature other terminologies for protocols based on a similar approach, e.g. Cycled Receiver [12], LPL (Low Power Listening) [11], Channel Polling [13], and Sparse Topology and Energy Management [14].

Fig. 1 shows an example of a preamble sampling protocol operation. According to the duty-cycle parameter, nodes switch their radios on each *Check Interval* to sample the channel. If a node finds the channel idle, it goes back to sleep immediately. However, if it detects a preamble transmission, then it keeps its radio on until it receives the subsequent data frame. Right after the reception of the data frame, the node sends an ACK frame, if needed, and goes back to sleep. To be effective, preamble transmission needs to be at least as long as the check interval (the period between two consecutive instants of node wakeup). In this way, a node makes sure that all potential receivers wake up during its preamble transmission, so they get the subsequent

data frame.

Without any particular optimization, preamble sampling protocols may consume large amounts of energy, because on the average nodes receive half of the preamble each time they receive a data frame. To reduce the overhead induced by the use a full length preamble, many improvements have been proposed. These proposals targeted both the cost at the receiver and that at the transmitter. At the transmitter side protocols propose techniques to minimize the preamble size. Details of these protocols are presented in Section VI. At the receiver side, protocols aim at reducing the duration of listening to the preamble. In [15] data frames are repeated in the preamble to minimize overhearing. By putting copies of the data frame in the preamble, a node that wakes up to sample the channel receives a copy of the data frame. From the destination field of the received frame, it knows whether it is the destination of the data. When the node is not the destination of the data frame, it switches off its radio to save energy. However, when the node is the destination of the data frame, it keeps receiving the data and has to transmit an ACK message back to the transmitter to acknowledge reception. As the preamble is just a repetition of the data frame, the receiver has no idea on when the whole transmission will end. Therefore, it keeps receiving until the transmitter stops its transmission of preamble plus the last data frame. When the transmission ends, the receiver sends its ACK message back to the transmitter. Keeping receiving to the whole transmission until the end presents a significant overhearing for the receiver. To cope with this, papers [16]–[18] propose to add a sequence number in the frames transmitted in the preamble. In [16], the sequence number is added to the data frames transmitted in the preamble. This allows the receiver to learn when the whole transmission ends and thus make it possible for it to switches off its radio in the mean time to save energy. It wakes up again only to transmit the ACK message. In [17], [18], the sequence number is added to small frames, referred to as wake up frames. The rational behind using wake-up frames instead of data frames is further overhearing reduction.

In this paper, we focus on reducing the cost at the receiver side and show in Section VI that the proposed solution is also compatible with solutions that reduce the cost at the transmitter side. We propose a new idea that reduces overhearing further on by avoiding reception of irrelevant frames. With our idea that is based on digest information, not only unicast frames can be filtered but also irrelevant (redundant) broadcast frames can be avoided a priori. We show that the use of the digest information together with the sequence number field substantially increases the

energy savings of frame preamble MAC protocols. We propose two possible implementations, Micro Frame Preamble (MFP) and Zebra Frame Preamble (ZFP), in which avoiding reception of irrelevant broadcasts can be integrated. We provide an analytical framework, in which we model the lifetime of various frame preamble MAC protocols. We complete the theoretical analysis with ns2 [19] simulation and experimental validation through implementation on the Chipcon CC 2500 evaluation kit [20]. We analytically show that the solution based on micro frames yields better performance. We discuss implementation choices and effect of power consumption of the micro controller on the overall performance of the solution. We also discuss potential improvements from using new architectures in which hardware circuitry can be used for constructing and transmitting preamble frames.

II. Frame Preamble Medium Access Control

A. Key Idea

The main drawback of traditional preamble sampling techniques comes from the overhead induced by the use of the long preamble prior to data transmission. To overcome this shortcoming, the preamble should carry some information to make it possible for receivers to minimize overhearing. A practical solution to insert information in the preamble is to replace it by a series of frames that we call in this paper preamble frames. We also use the generic term Frame Preamble to refer to all the MAC schemes in which the preamble is composed by a series of frames.

The efficiency of the implementation of Frame Preamble MAC in reducing overhearing depends on preamble-frame structure. Two main features which are (i) minimizing preamble reception, and (ii) avoiding reception of irrelevant data provide substantial additional energy savings. In the next section, we describe some fields that can be used in Frame Preamble MAC protocols.

B. Design Features

As all traditional frames, preamble frames also have physical layer fields, namely: the physical layer preamble, the synchronization word, the frame length indicator, and the CRC field (see Fig. 2).

In addition to those physical layer fields, a preamble frames may include all or some of the following MAC fields. Fig. 2 shows an example of a preamble frame structure[1].

1) The Type Field: This field specifies the type of frame being received. It is used to distinguish between various frame types, data, preamble, ACK, etc.

2) The Destination Address Field: This field is required to allow filtering of irrelevant unicast frames that are not addressed to the node.

3) The Source Address Field: This field may be considered optional. It is required to let the receiver know the address of the transmitter so that it can send the ACK message back to it after a successful unicast data reception. There are two options for including this field in Frame Preamble MAC protocols. Either it is included in all the preamble frames or it is only included in the data frame. Note that the performance of the resulting Frame Preamble MAC depend on the combination of the Frame Preamble variant used and on whether the source address field is included in the preamble frames or not.

4) The Sequence Number Field: This field indicates the number of preamble frames that will be transmitted before the data frame. When this field is used, the node that wakes up to sample the channel receives a preamble frame from which it can deduce when the data frame will arrive thereby it can switch its radio off in the mean time to save energy. The sequence number field significantly minimizes preamble overhearing and thus increases the energy saving of Frame Preamble MAC protocols. In addition, the sequence number offers another benefit, which is reducing the carrier sense overhead. The node uses the sequence number and the destination address of the received microframe to estimate the schedule of data transmission including the ACK frame. In this way, it postpones its transmission until the end of the ongoing transmission, which saves the energy drained in potential repeated attempts for accessing the channel during the ongoing transmission. Note that the node re-executes a new backoff procedure at the end of each transmission to minimize potential contention.

5) The Digest Field: The digest field is envisaged to make it possible for a node to identify and thus to avoid receiving irrelevant broadcast frames before their entire reception. In many networking solutions, protocols use flooding to propagate an information throughout the network, perform robust information delivery, etc. In such cases, a node may receive multiple copies of

[1]We present the details of PHY headers as implemented on Chipcon CC2500 described later.

the same frame contents from all its neighbors. As only the first data contents is useful, the node saves significant energy amount by ignoring the subsequent broadcasts carrying the same data contents.

The digest field may be either a unique identifier (a combination between a sequence number and the source address of the node that started the broadcast) or a hash of the data contained in the broadcast frame. When a node wants to transmit a broadcast frame, it calculates the digest field to be included in the preamble frame and inserts it in a table to avoid receiving a similar broadcast from another node. This table logs digests of the broadcast frames that have been recently seen so that the node may switch its radio off when it expects a redundant reception.

According to this, a node that receives a preamble frame which the destination address is broadcast has to check in its table whether there is an entry with the same digest value carried in the preamble frame. If such an entry exists, then the expected broadcast data is redundant. Else, the expected broadcast data is new and the node should receive it. Once that broadcast is received, the node updates its table to avoid receiving redundant transmissions of the just received data frame.

One can argue that the use of hash functions to calculate the digest values may lead to conflicts that cause a node to ignore a data frame that has not previously received — this happens when two different data have the same hash result. We think that such a situation is hardly likely to happen because of the following reasons. First, digest field entries in the table of the MAC protocol are not permanent, but cleaned after a timeout value. Second, we can choose a suitable hash function and digest size so that collisions are very rare. A frame will be missed only if it involves two simultaneously active broadcasts with the same hash value during the timeout. Note that because broadcasts are not acknowledged, they are usually unreliable anyway.

For an efficient implementation, the CRC (Cyclic Redundancy Check) circuitry can be re-used to calculate the digest of data frames. The advantages of such a solution are twofold. First, it saves the overhead of calculating costly hash functions, which increases energy savings. Second, it does not require additional dedicated hardware for hash calculation, which does not increase the cost of the sensor node.

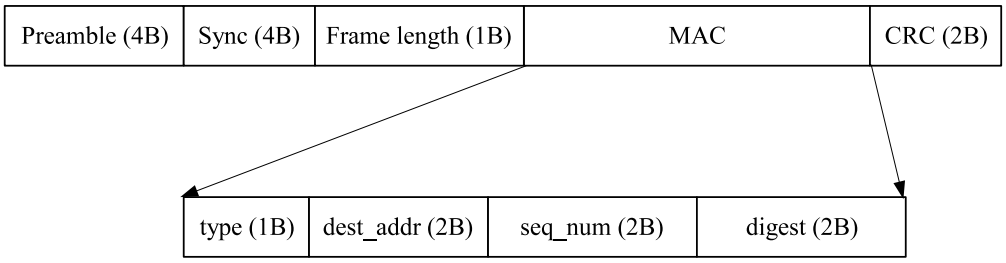


Fig. 2. Preamble frame structure (PHY and MAC) as implemented in the CC2500 module. The MAC structure depends on the variant of the frame preamble protocol used. This figure shows the fields used in the case of MFP. Field sizes, in bytes, are in parentheses.

C. Various Implementation Flavors

1) Data Frame Preamble: In this variant proposed in [16], the preamble frames are copies of the data. In this protocol, referred to as DFP (Data Frame Preamble) in this paper, each preamble data-frame only includes the sequence number field but not the digest field, therefore no irrelevant frames are filtered.

2) Micro Frame Preamble: In this variant that we call MFP (Micro Frame Preamble), the preamble is composed of small frames called micro frames. Each micro frame contains a sequence number and a digest field. Thus, MFP both minimizes overhearing and avoids irrelevant broadcast receptions.

3) Zebra Frame Preamble: In this variant the preamble is composed of an alternation of micro frames and data frames: each preamble data-frame is preceded by a micro-frame. This variant aims at combining both the advantages of MFP (no need to spend a long time for receiving a whole data frame in the preamble to decide whether it is relevant or not), and DFP (no need to wake up again to receive the data frame).

III. Theoretical Evaluation

Both of MFP, DFP and ZFP consume the same amount of energy in transmission as the time a node spends in transmission (which corresponds to the transmission of the preamble and the subsequent data) is the same for all. Therefore, to identify the optimal configuration for the Frame Preamble MAC protocol, i.e. under which circumstances it saves more energy to use MFP, DFP or ZFP, we propose to compare the average times a node spends in reception with MFP, DFP and ZFP.

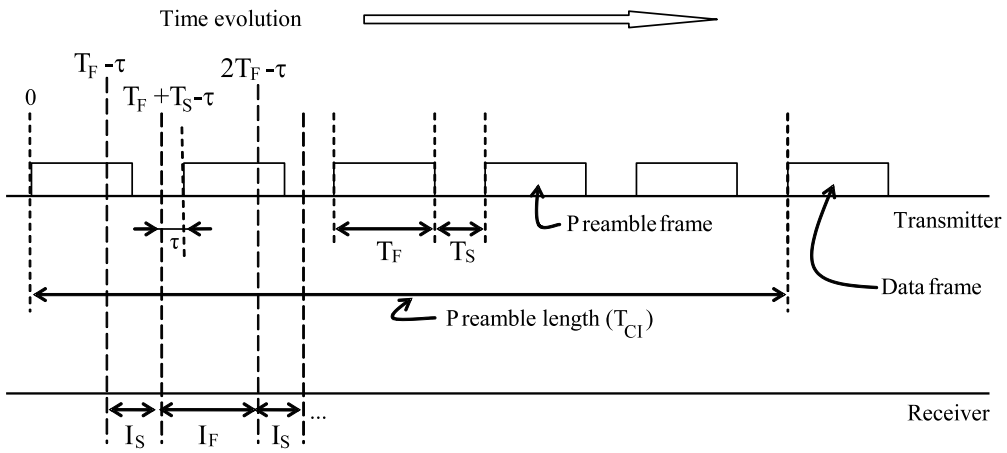


Fig. 3. Transmission of preamble frames with a gap between each two consecutive frames.

In this section, we analytically evaluate the performance of several representative preamble sampling protocols. We take LPL (Low Power Listening) [11] as a reference of preamble sampling protocols. We start by modeling the reception duration of a node when it uses Frame Preamble (MFP, DFP, or ZFP) or LPL. Then, we use this result to calculate the average lifetime duration of a node. Note that for the rest of the analysis, we do not consider the power drained in Sleep mode as it is generally negligible compared to that of other modes (4 orders of magnitude lower, see Table II for measured current consumption of CC2500).

A. Reception Cost

Let $T_{\circ}^{r}(t)$ be a random variable that expresses the time a node spends in Receive mode listening to the preamble. The $\circ$ designates the preamble protocol being used, which can be LPL, MFP, DFP, of ZFP.

1) LPL: In preamble sampling protocols, each node chooses its schedule independently of the others. Therefore, it may wake up at any random instant in $[0, T_{\mathrm{CI}}]$, where T_{CI} is the duration of the Check Interval. When a node wakes up and detects a preamble being transmitted on the channel, it keeps listening to the preamble until it receives a data frame. Therefore, we have:

$$T_{\mathrm{LPL}}^{r}(t) \quad = \quad \tau + U_{[0,T_{\mathrm{CI}})}(t). \tag{1}$$

where τ is the time needed to go from Sleep mode to Receive or to Transmit modes[2] and $U_{[0,u)}(t)$ stands for a uniform random variable in $[0, u)$. Therefore, the energy drained in Receive mode is:

$$\mathcal{E}^r_{\mathrm{LPL}} = (E[T^r_{\mathrm{LPL}}(t)] + T_{\mathrm{data}})P_r = \left(\tau + \frac{T_{\mathrm{CI}}}{2} + T_{\mathrm{data}}\right) P_r. \tag{2}$$

where $E[T^r_{\mathrm{LPL}}(t)]$, the mean time spent in preamble reception, T_{data} is the time needed to receive one data frame, and P_r is the power drained when the radio is in Receive mode.

2) Homogeneous Frame Preamble MAC: In this class of protocols, the preamble frames are of the same type such as in MFP and DFP. As each node chooses its schedule locally, the node may wake up at any time during the transmission of the preamble: during I_S or I_{F} intervals as shown in Fig. 3. We denote the duration of I_S by T_S (the inter preamble frames gap) and the duration of I_{F} by T_F (the preamble frame transmission time).

As shown in Fig. 3, if the node wakes up during I_{F}, it misses the beginning of a preamble frame and it cannot decode it correctly[3]. In this case, the node suspects it has just lost a preamble frame, so it keeps listening to the channel until it receives the subsequent preamble frame. If the node finds the channel idle for a duration longer than T_S, then it goes back to sleep concluding that no frames are being transmitted. However, if the node wakes up during I_S, then it receives the subsequent preamble frame.

Therefore, if the node wakes up during I_S (resp. I_{F}) interval, then the time it spends in Receive mode to correctly decode a microframe is $\mu_S(t)$ (resp. $\mu_F(t)$):

$$\begin{cases} \mu_F(t) = \tau + U_{[0,T_F)}(t) + T_S + T_F. \\ \mu_S(t) = \tau + U_{[0,T_S)}(t) + T_F. \end{cases} \tag{3}$$

The probabilities q_S and q_F that a node wakes up during I_S and I_F intervals are the following, respectively:

$$\begin{cases} q_S = \dfrac{T_S}{T_S + T_F}. \\ q_F = \dfrac{T_F}{T_S + T_F}. \end{cases} \tag{4}$$

[2]Note that τ is not negligible and smaller values for τ substantially increase the performance of preamble sampling protocols.

[3]We assume that a node will fail to decode a frame if it misses the reception of its first bit. This assumption can be easily relaxed — we can easily assume that the node will fail to decode a frame correctly if it misses the x first bits of the preamble. In both cases, the analysis remains the same, only the durations of the intervals I_{F} and I_S change.

From (Eq. 3) and (Eq. 4), we deduce the time needed to receive an entire preamble preamble, which is equal to:

$$T_{\mathrm{F}}^{r}(t) = q_F \mu_F(t) + q_S \mu_S(t) \quad (5)$$

Therefore, the average duration to receive a preamble frame is:

$$\begin{aligned} E[T_{\mathrm{F}}^{r}(t)] &= q_F E[\mu_F(t)] + q_S E[\mu_S(t)] \\ &= \left(\tau + \frac{T_S}{2} + \frac{3T_F}{2}\right) \end{aligned} \quad (6)$$

- Application to DFP.

  In the case of DFP, the preamble is composed of DFP frames. The transmission duration of a DFP frame is T_{DFP}. Note that the transmission duration of a DFP frame is slightly larger than that of data frame T_{data}, because a DFP frame includes an additional field: the sequence number. As in DFP preamble frames are copies of the data frame, the reception of one preamble frame is sufficient for a successful reception. Therefore, we have:

$$\begin{aligned} \mathcal{E}_{\mathrm{DFP}}^{r} &= E[T_{\mathrm{DFP}}^{r}(t)] P_r \\ &= \left(\tau + \frac{T_S}{2} + \frac{3T_{\mathrm{DFP}}}{2}\right) P_r. \end{aligned} \quad (7)$$

- Application to MFP.

  Following the same methodology used for DFP, we obtain the average preamble reception time with MFP. According to (Eq. 5), we have:

$$E[T_{\mathrm{MFP}}^{r}(t)] = \tau + \frac{T_S}{2} + \frac{3T_{\mathrm{MFP}}}{2}. \quad (8)$$

  where T_{MFP} is the transmission duration of a MFP frame. To calculate the mean energy drained in reception with MFP, we to take into account the feature of irrelevant data filtering. To evaluate the impact of receiving irrelevant data frames, we assume that the proportion of relevant data frames is α. The value of α may depend on many parameters, e.g. the application's traffic pattern. In the case of nodes running only a flooding application, the reception of only one message is sufficient, so all other copies of the same message subsequently forwarded by neighbors become irrelevant. Therefore, if a node has n neighbors, then all the $(n-1)$ messages forwarded afterward are redundant. In this case, $\alpha = 1/n$.

  In the case of MFP, a node that wakes up to sample the channel receives a MFP frame after which it goes to sleep until the data transmission. As the node wakes up again to receive

the data frame only in α percent of cases, the mean energy drained to receive a data frame with MFP is:

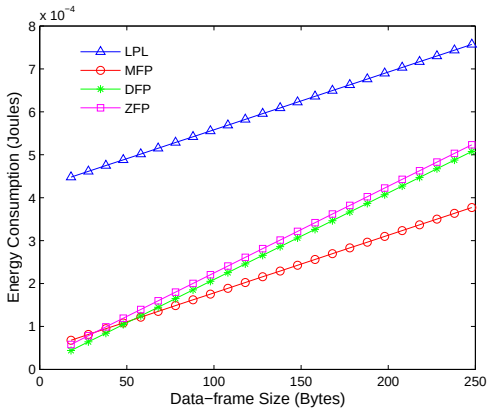
$$\begin{aligned}\mathcal{E}^r_{\mathrm{MFP}} &= \Big[E[T^r_{\mathrm{MFP}}(t)] + (\tau + T_{\mathrm{data}})\alpha\Big] P_r \\ &= \left[\tau + \frac{T_S}{2} + \frac{3T_M}{2} + (\tau + T_{\mathrm{data}})\alpha\right] P_r. \end{aligned} \quad (9)$$

where $(\tau + T_{\mathrm{data}})$ is the time needed to receive a data frame (it includes τ, because we assume that the node goes to Sleep mode after receiving a MFP frame).

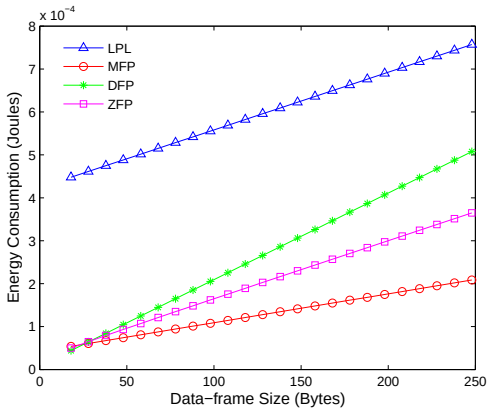
3) Hybrid Frame Preamble MAC: In this class of protocols, the preamble frames are a series of alternation of two types of frames, e.g. the alternation of MFP and DFP frames in ZFP. We apply the same methodology to calculate the energy drained in reception in ZFP. Details for the derivations are in the Appendix. We have

$$\begin{aligned}\mathcal{E}^r_{\mathrm{ZFP}} &= \Bigg(\tau + \frac{1}{2(2T_S + T_{\mathrm{MFP}} + T_{\mathrm{DFP}})}\Big\{ \\ &\quad T_S(T_S + 2T_{\mathrm{MFP}} + 2\alpha(\tau + T_{\mathrm{DFP}})) + \\ &\quad T_{\mathrm{MFP}}(T_{\mathrm{MFP}} + 2T_S + 2T_{\mathrm{DFP}}) + T_S(T_S + 2T_{\mathrm{DFP}}) + \\ &\quad T_{\mathrm{DFP}}(T_{\mathrm{DFP}} + 2T_S + 2T_{\mathrm{MFP}} + 2\alpha(\tau + T_{\mathrm{DFP}}))\Big\}\Bigg) P_r. \end{aligned} \quad (10)$$

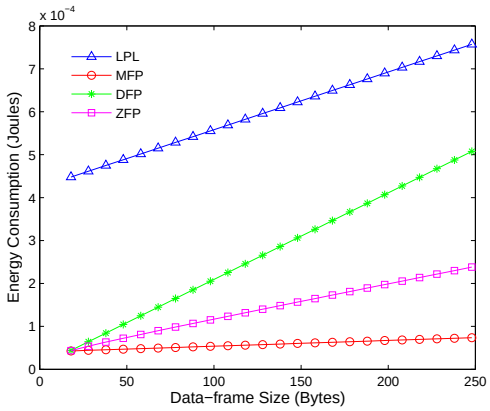
4) Discussion: To identify the best method to use, we plot values of $\mathcal{E}^r_{\mathrm{LPL}}$, $\mathcal{E}^r_{\mathrm{DFP}}$, $\mathcal{E}^r_{\mathrm{MFP}}$, and $\mathcal{E}^r_{\mathrm{ZFP}}$. We set the bandwidth of the radio to $250kb/s$, the MFP frame size to 18 bytes, the DFP frame size is two bytes larger than the data frame size because of the sequence number field, the turn-around time τ to 88μs and the the inter preamble frame space T_S to 0 s. We varied the value of the DFP frame size and we plot the result with different values of α (see Fig. 4). Overall, we show that MFP is the best in all the situations, except when data frames are all relevant (i.e. $\alpha = 1$) and have a size less then 40 bytes (see Fig. 4(a)) in which case DFP performs better. Thus, we can imagine a hybrid protocol that uses DFP when it sends frames less than 40 bytes and MFP when the data-frame size is larger than 40 bytes in the case that upper layers consider that all data frames are relevant to receive. For the rest of the paper, we restrain the analysis to MFP as it is the frame preamble variant that offers the best performance and to LPL as it is the reference protocol.



(a) $\alpha = 1.00$



(b) $\alpha = 0.50$



(c) $\alpha = 0.10$

Fig. 4. Mean Energy Consumed per Received Message According to the Filtering Rate α. Results are obtained with a check interval of 20ms. Note that these results hold for different check intervals and the amount of energy savings are amplified compared to LPL for larger check intervals.

B. Channel Sampling Cost

The average time a node spends in channel sampling with LPL is the following:

$$T^s_{\rm LPL} \;=\; \tau + T_{\rm CS}. \tag{11}$$

where $T_{\rm CS}$ is the time needed to determine whether there is a preamble being transmitted on the channel.

In MFP, the channel sampling duration is slightly different, because of the possible inter MFP frames gap (I_S interval in Fig. 3). Indeed, a node must sense the channel for a duration of T_S to determine whether the channel is free. Therefore, the duration of channel sampling in MFP is the following:

$$T^s_{\rm MFP} \;=\; \tau + T_S + T_{\rm CS}. \tag{12}$$

The amount of energy drained in channel sampling with LPL and MFP is:

$$\mathcal{E}^s_{\rm LPL} \;=\; T^s_{\rm LPL} P_{\rm samp} = (\tau + T_{rmCS})\, P_{\rm samp} \tag{13}$$

$$\mathcal{E}^s_{\rm MFP} \;=\; T^s_{\rm MFP} P_{\rm samp} = (\tau + T_S + T_{\rm CS})\, P_{\rm samp}. \tag{14}$$

where $P_{\rm samp}$ is the power drained when the radio is in channel sampling mode. This power is almost equal to the power in Receive mode. For the rest of the analysis, we assume $P_{\rm samp} = P_r$.

C. Transmission Cost

The energy drained in the transmission of one message in LPL is the following:

$$\mathcal{E}^t_{\rm LPL} \;=\; \mathcal{E}^s_{\rm LPL} + (\tau' + T_{\rm CI} + T_{\rm data}) P_t. \tag{15}$$

where P_t is the power drained when the radio is in Transmit mode and τ' is the transition time from Receive mode to Transmit mode.

In MFP, the energy drained in the transmission of one message depends on $N_{\rm MFP}$, the number of MFP frames transmitted in the preamble. As the radio potentially goes to Sleep mode between MFP frame transmissions, the number of MFP frames transmitted in the preamble is:

$$N_{rmMFP} = \left\lceil \frac{T_{\rm CI}}{T_S + T_M} \right\rceil. \tag{16}$$

Therefore, the energy drained during a transmission in MFP depends on the inter MFP frames time. If the inter MFP frames time is larger than the transition time τ, then the transmitter

goes to Sleep mode between the transmission of two consecutive MFP frames. Otherwise, the transmitter does not go to Sleep mode and we assume that the energy drained during the inter MFP frames gap is equal to that drained in Transmit mode. Therefore, we have:
if $T_S < \tau$, then:

$$\mathcal{E}^t_{\rm MFP} \;=\; \mathcal{E}^s_{\rm MFP} + \left(\tau' + \left\lceil \frac{T_{\rm CI}}{T_S + T_{\rm MFP}} \right\rceil T_{\rm MFP} + T_{\rm data}\right) P_t. \tag{17}$$

if $T_S \geq \tau$, then:

$$\mathcal{E}^t_{\rm MFP} \;=\; \mathcal{E}^s_{\rm MFP} + \left(\tau' + \left\lceil \frac{T_{\rm CI}}{T_S + T_{\rm MFP}} \right\rceil (\tau + T_{\rm MFP}) + T_{\rm data}\right) P_t. \tag{18}$$

D. Modeling Node Lifetime

We define $\mathcal{L}_\circ$, the lifetime duration of a node as

$$\mathcal{L}_\circ \;=\; \frac{E_{\rm initial}}{\mathcal{P}_\circ}. \tag{19}$$

where $\mathcal{P}_\circ$ (joules/sec) is the average power consumed by the sensor node and $E_{\rm initial}$ (joules) is its initial energy. The triangle '$\circ$' can be either MFP or LPL. For the sake of conciseness and simplicity, we only consider the power consumed by the radio — the overhead of the microcontroller is very small as shown in Section V-B. Therefore, we have

$$\mathcal{P}_\circ \;=\; \mathcal{P}^t_\circ + \mathcal{P}^r_\circ + \mathcal{P}^s_\circ. \tag{20}$$

where $\mathcal{P}^t_\circ$ (resp. $\mathcal{P}^r_\circ$, $\mathcal{P}^s_\circ$) is the average power drained in transmission (resp. reception, sampling). The average power drained during preamble sampling is

$$\mathcal{P}^s_\circ \;=\; \frac{\mathcal{E}^s_\circ}{T_{\rm CI}}. \tag{21}$$

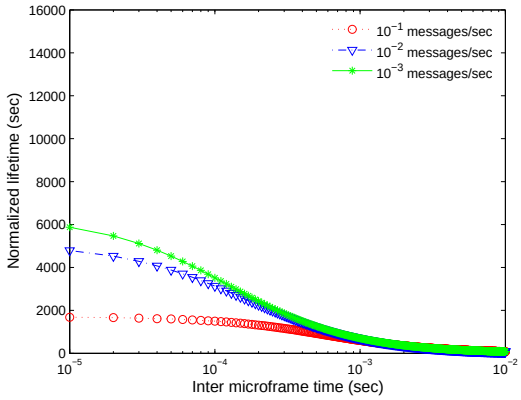
Similarly, we compute the average power drained during transmission $\mathcal{P}_t$

$$\mathcal{P}^t_\circ \;=\; \frac{\mathcal{E}^t_\circ}{T_{\rm traffic}}. \tag{22}$$

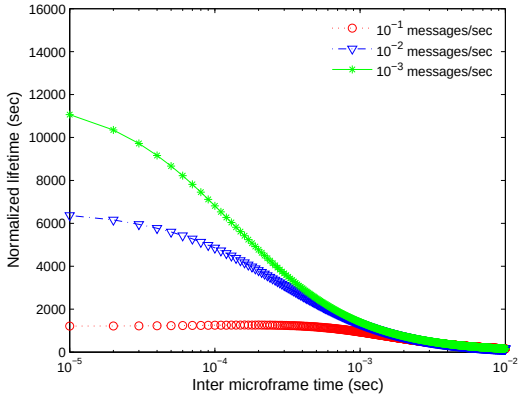
and the average power drained during reception $\mathcal{P}_r$

$$\mathcal{P}^r_\circ \;=\; \frac{\mathcal{E}^r_\circ \cdot n}{T_{\rm traffic}}. \tag{23}$$

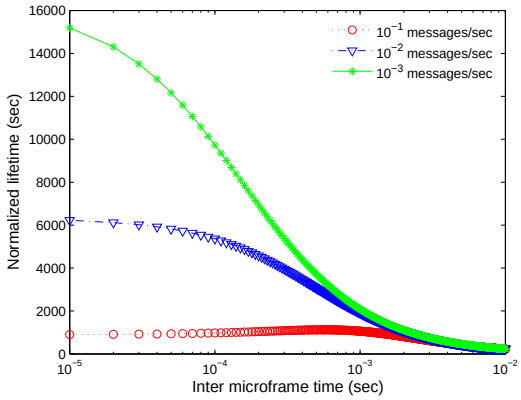
where $T_{\rm traffic}$ is the average inter data-frame transmission time that characterizes the application traffic load. For the derivation of the average power drained during reception (Eq. 23), we have assumed that nodes run a simple flooding protocol in which each node forwards each new



(a) Check interval =50ms



(b) Check interval =100ms



(c) Check interval =150ms

Fig. 5. Normalized lifetimes of a node using MFP in function of different inter microframes gaps and various traffic loads.

message exactly once. Therefore, a node with n neighbors will initially receive n broadcast messages. However, if the node is running MFP then it will filter the $n-1$ redundant broadcasts from its neighbors according to the α parameter as explained in (Eq. 9).

Note that the lifetime of nodes running MFP depends on many parameters, among which the inter microframe time T_S. Finding the best value T_S^* that maximizes the lifetime of nodes running MFP is equivalent to finding a solution to the following equation:

$$\frac{d\mathcal{L}_{\mathrm{MFP}}}{dT_S} = 0 \tag{24}$$

For the sake of simplicity, we choose not to analytically solve the above equation. Instead, we plot Fig. 5 that shows the normalized lifetime (corresponding to the lifetime of a node with an initial energy of $E_{\mathrm{initial}} = 1$ Joule) of nodes in function of the inter microframe time. For this plot, we have used the following parameters: the transition times τ and τ' are equal to 88.4μs and 9.6μs respectively[4], the transmission times of a 18-byte microframe and a 256-byte data frame at 250kb/s are equal to $T_{\mathrm{MFP}} = 576\mu$s and $T_{\mathrm{data}} = 8.48$ms, respectively, and the carrier sense time T_{CS} is equal to 32μs[5]

Fig. 5 shows that a node that uses MFP benefits from longer lifetimes when the traffic load is low and the check interval is large. This is expected, because lower traffic loads imply less transmissions and larger check intervals imply longer sleep periods. Fig. 5 also shows that shorter inter microframe gaps improve lifetimes, because the sampling time is smaller (T^s_{MFP} decreases when T_S decreases). The energy saved with smaller inter microframe gaps is larger when the traffic load is lower, because nodes spend more time in sampling than in transmitting messages. Note that even if there is an optimal value for T_S that can be calculated from (Eq. 24), we notice from the graphs in Fig. 5 that taking very small values for T_S yields the maximum lifetime in most of the cases. Therefore, for the sake of simplicity, we set T_S to 0 for the remainder of the analysis in this paper.

[4]See state transition timings of the CC2500 [20]

[5]We use the automatic termination of RX function after a programmable timer, which is made possible by the CC2500 [20]. In this case, the radio controller will terminate the reception if the first valid carrier sense sample indicates no carrier (RSSI below threshold). The minimum value needed for the carrier sense sample is 8 symbols, which corresponds to 32μs (the duration of a symbol is 4μs because the CC2500 can use OOK modulation for transmission at 250kbps [20])

E. Comparisons

To evaluate the performance of MFP and compare it with LPL, we plot Fig. 6 with the same parameters used in Fig. 5. The amount of energy saved in preamble sampling protocols depends on the duty cycle. To save more energy, nodes need to use lower duty cycles to spend more time in low-power mode. However, although lowering the duty cycle (i.e. increasing the check interval) does reduce idle listening (see Eq. (21)), it has some side effects: it increases the cost of transmissions (see Eq. (15)) and receptions (see Eq. (2)). Therefore, there is an optimal value for the check interval T^*_{CI} that achieves the best trade-off between the cost of transmission, reception, and channel sampling in function of the traffic generated by applications. The value of T^*_{CI} is the solution of the following equation:

$$\frac{d\mathcal{L}_\circ}{dT_{CI}} = 0 \tag{25}$$

Therefore, we have

$$T_{CI}^{*,\mathrm{LPL}} = \sqrt{\frac{(\tau + T_{\mathrm{CS}})P_{\mathrm{samp}}}{\frac{P_t}{T_{\mathrm{traffic}}} + \frac{P_r n}{2T_{\mathrm{traffic}}}}} \quad \text{for LPL} \tag{26}$$

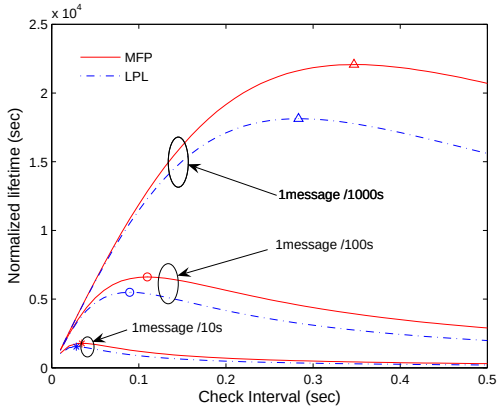
and,

$$T_{CI}^{*,\mathrm{MFP}} = \sqrt{\frac{(\tau + T_{\mathrm{CS}})P_{\mathrm{samp}}}{\frac{P_t}{T_{\mathrm{traffic}}}}} \quad \text{for MFP} \tag{27}$$

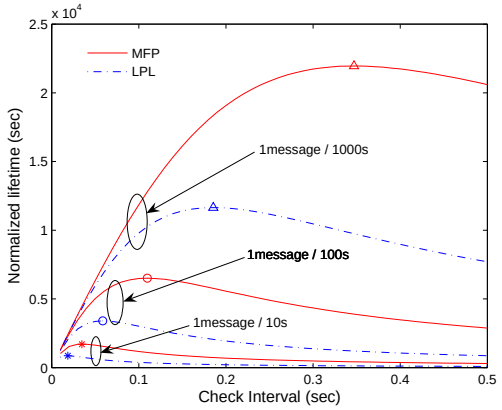
In Fig. 6, we show normalized lifetimes for MFP and LPL in function of the check interval. We can see that a node has a longer lifetime for MFP than for LPL for all the used check intervals. The figure also shows that optimal check intervals are larger for MFP, which means that MFP allows smaller duty cycles and thus longer lifetimes. We can also see that the lifetime of a node that uses LPL decreases when the number of its neighbors increases, because the node spends energy in receiving redundant messages transmitted by its neighbors (each message contains preamble plus data). However, when a node uses MFP, neither it does receive the preamble of these redundant messages nor does it receive the redundant data. Therefore, its lifetime does not decrease. We can also notice that even if there are no irrelevant receptions ($n = 1$), MFP presents a substantial gain of lifetime.

IV. SIMULATIONS

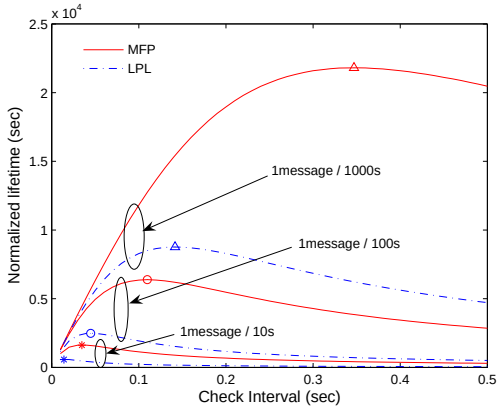
The analysis in Section III does not take collisions into account. Therefore, actual lifetimes may differ from the values obtained analytically. Thus, we have run simulations: we have



(a) $n = 1$



(b) $n = 5$



(c) $n = 10$

Fig. 6. Normalized lifetime of nodes for different check intervals. Microframes are transmitted without inter microframe gaps, i.e. $T_S = 0$.

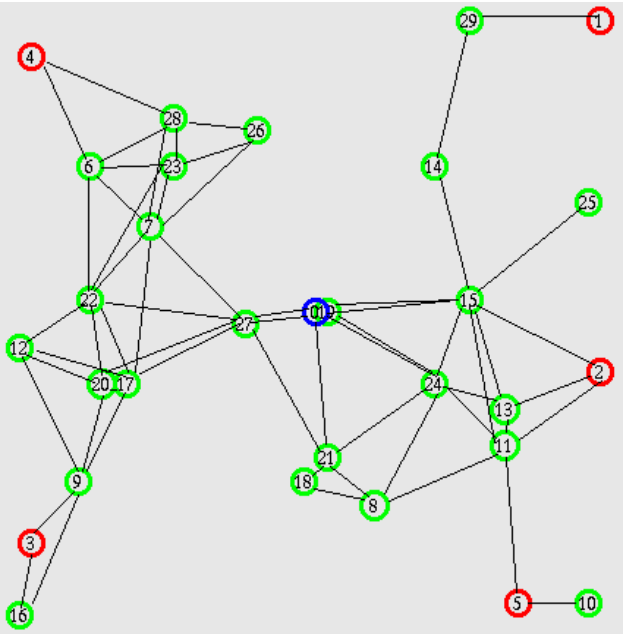


Fig. 7. Chosen random topology: Node 0 is the sink, Nodes 1 through 5 are the sources, the other nodes (6 through 29) are intermediate nodes that relay traffic.

implemented MFP and LPL in ns-2 [19] and compared their performance in large networks with random topologies. We have generated different topologies in which we randomly choose x and y coordinates of each node according to a uniform distribution. We make sure that each of the generated networks is connected, i.e. there is a path between any two nodes. Fig. 7 shows an example of such a network.

To get into the details of the protocol behavior, we have run our simulation on a chosen topology presented in Fig. 7. It includes nodes with a very low degree of connectivity (e.g. nodes 1, 10, and 25 have only one neighbor) as well as a high degree of connectivity (e.g. node 15 has eight neighbors). In addition, there are several routes from the sources (nodes 1 through 5) to the sink (node 0).

We use the parameters of PHY and MAC layers for the CC2500 evaluation board [20]. We set the size of a microframe to 18 bytes that corresponds to 576μs at 250Kb/s. As ns-2 is a packet level simulator, a node that misses the first bit of a frame (e.g. because it is sleeping) will also miss all the subsequent bits of the same frame. That is why we require that carrier sense duration be longer than one microframe transmission time to guarantee that a node missing the first bit

of a microframe can detect the first bit of a potential subsequent microframe (or a data-frame).

We consider three MAC protocols: LPL, MFP with and without filtering of redundant messages that we call MFP-filter and MFP-nofilter, respectively. We compare the gain obtained from avoiding unneeded listening to the preamble (the case of MFP-nofilter) with the gain obtained from both avoiding keeping listening to the preamble and filtering out redundant messages (the case of MFP-filter).

For the routing and application layers, we use simple flooding. Each source periodically generates a message and broadcasts it to other nodes. For each simulation run, we record the number of messages each node has transmitted, correctly received, and the number of collided messages it has observed. Note that we only count the number of distinct received messages, i.e. we do not count redundant copies of the same message received by the same node. Such redundant messages result from the forwarding of the same message by the neighbors of the node. We do not count redundant messages in order to be fair in comparison with MFP-filter that does not receive redundant messages.

We define two particular instants for gathering statistics during simulation: the *first-node* and the *all-nodes* instants. The former corresponds to the instant of the death of the first node whereas the latter is the instant of the death of the last node, i.e. when all the nodes are dead.

We have run extensive simulations for a wide range of parameters. Each point of the following plots is the average over 10 simulation runs.

A. Network-Scope Evaluation

In the first group of simulation, we set the data frame size to 150 bytes, the traffic generation interval to 100 seconds and we vary the check interval from 50 to 750ms. For each value of the check interval, we measure the lifetime for each node. For this purpose, we introduce the *Information Transport Efficiency* (ITE) that quantifies the amount of information passed through each node during its lifetime. First, we count the number of relevant data frames the node has sent or received. Then, we quantify the ITE as the number of relevant frames passed through the node (or the number of bits) per joule. Note that, the ITE considers only the useful information: redundant broadcast frames do not count. The ITE is more accurate for quantifying the lifetime of a node than just measuring the time it takes a node to run out of energy, because the lifetime of a node also depends on the quantity of information passed through it — extending the lifetime

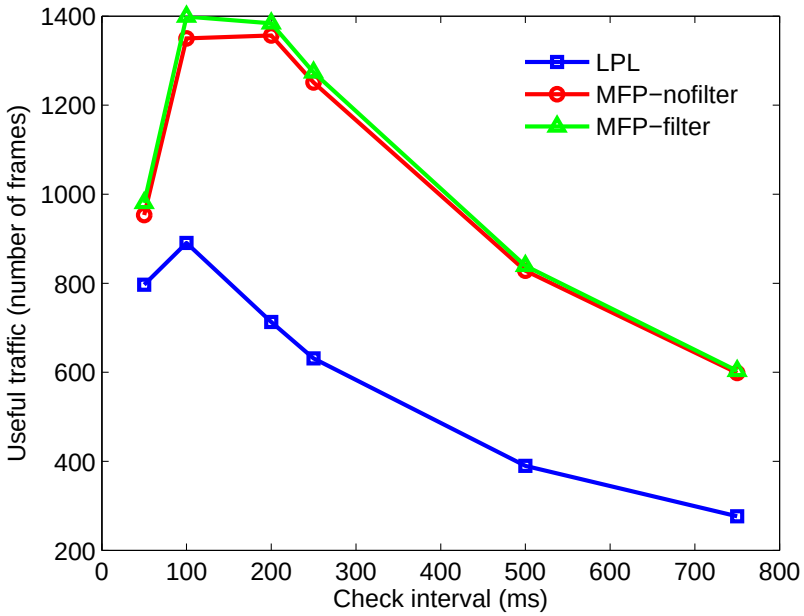


Fig. 8. The mean lifetime of nodes for various check intervals. It is measured in terms of the number of relevant frames that passed through the node per 1 joule. The lifetime is averaged over all the nodes.

of a node means allowing more information to pass through it per energy unit. Hereafter, the term normalized lifetime is measured according to the ITE principle. Specifically, it refers to the number of relevant frames passed through a node per 1 joule.

Fig. 8 shows the evolution of the mean lifetime of nodes in function of the check interval. Each point on the curves presents the lifetime averaged over all the nodes and measured at the *all-nodes* instant. Fig. 8 shows that MFP significantly increases the mean lifetime of nodes for all check intervals. In addition, filtering irrelevant messages increases the mean lifetime furthermore: the MFP-filter curve is continuously above that of MFP-nofilter. For the parameters of Fig. 8, filtering redundant messages does not provide significant lifetime increase compared to that of avoiding receiving the preamble — the duration of listening to the preamble is by far longer than that of receiving the data, especially when data frames are quite small and transmitted at high speeds (e.g. 250kb/s). The average lifetime measured at the *first-node* instant (not plotted in the figure) is also maximized when the check interval is between 100ms and 200ms.

Fig. 9 shows the percentage of lifetime extension measured both at the *first-node* and *all-nodes* instants for MFP-filter and MFP-nofilter compared to LPL. We can see that the gain obtained by

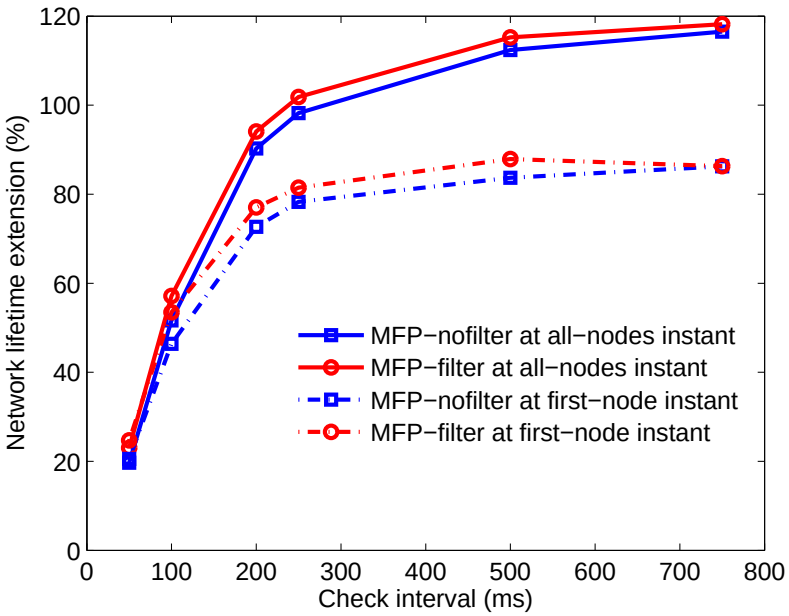


Fig. 9. The mean lifetime extension of MFP compared to LPL.

using MFP is more important at the *all-nodes* instant, because simulation continues and nodes can improve their energy savings. At the *first-node* instant, the gain corresponds to that of the first node, i.e. when the most vulnerable node exhausts its energy.

Fig. 10 shows collision rate, which is the ratio of the number of received collided packets to the number of all the received packets, measured at the *first-node* instant, i.e. from the beginning of the simulation until the first-node instant. Note that after this instant collision rates decrease because the network becomes less dense as nodes start disappearing.

Collision rates measured at the *all-nodes* instant (not plotted in Fig. 10) exhibit a similar behavior, but with lower rates. Collisions affect nodes lifetime, because in our simulations a node keeps its radio in Receive mode while there is activity on the channel until it correctly decodes a frame or the channel is idle again. If there is a collision, the node spends time in Receive mode listening to the collided frames, which drains energy and thus decreases the lifetime of the node. The energy drained in listening to collisions depends on the transmission duration of the whole message (preamble plus data). The larger the message, the longer the duration of listening to collisions. The preamble length and thus the check interval should be

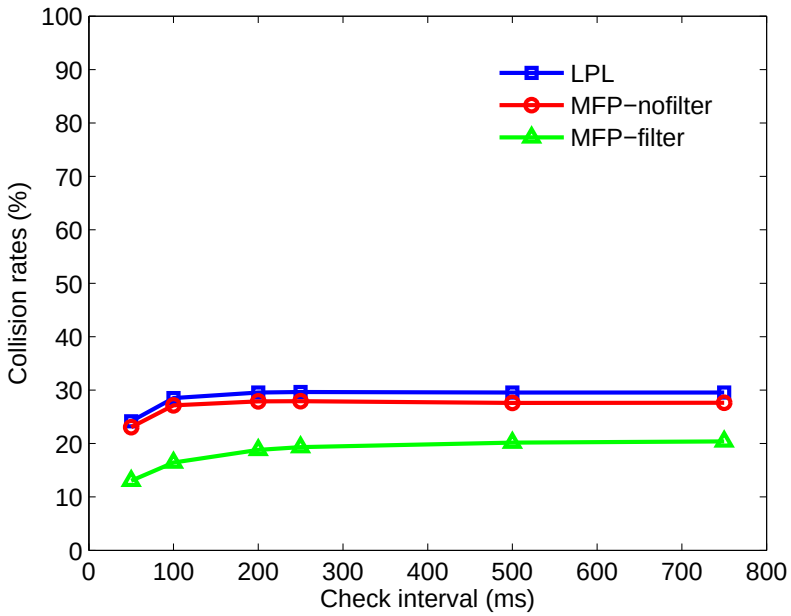


Fig. 10. The mean collision rates.

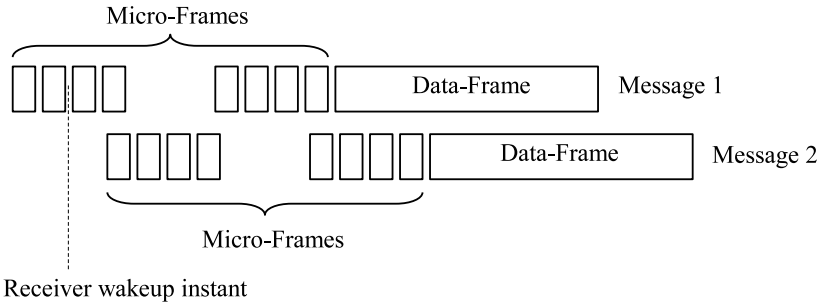


Fig. 11. Message collision in ns-2 simulation. Messages 1 and 2 are transmitted by nodes that are out of transmission range of each other — the collision occurring at the receiver is caused by the hidden node problem.

smaller to reduce the amount of energy wasted in listening to collisions.

Collision rates for MFP-nofilter and LPL are fairly similar, however those for MFP-filter are much smaller. Our simulation of MFP does not guarantee that MFP-filter has the same execution sequence as MFP-nofilter and LPL. Fig. 11 explains why this may happen. Assume that Message 1 is transmitted before Message 2 and the receiver is able to correctly decode a microframe of the Message 1 preamble. This is possible, if the receiver wakes up before

TABLE I

PERFORMANCE FOR CHOSEN NODES.

Node Identifier	Degree	Collision	Transmission	Reception	Lifetime extension
Node 1	1	0%	60%	40%	40%
Node 15	8	60%	20%	20%	80%
Node 25	1	0%	50%	50%	80%
Node 0	5	20%	40%	40%	120%
Node 27	7	40%	30%	30%	140%

Message 2 is transmitted. Then, assume that Message 1 is redundant for the receiver. Thus, MFP-filter will filter it out and switch the radio off during its transmission. Consequently, if the receiver uses MFP-filter protocol, then it will not observe this collision as its radio will be off. However, if the receiver uses MFP-nofilter or LPL, then it will observe a collision as it will wake up to receive the data frame of Message 1, which will be corrupted by Message 2.

B. Node-Scope Evaluation

In this section, we present simulation results for some chosen nodes to provide more details on the performance of our protocol. In a random topology, nodes may fall into different categories: low or high degree of connections, low or high traffic load, vulnerable position in the graph or not, etc.

As shown in Table I, lifetime extension ranges from 40% for node 1 to 140% for Node 27. Node 15 is the most vulnerable as it is the first to run out of energy. Node 15 dies before the others because it has the largest number of neighbors (eight) and thus the highest activity rate. Node 25, albeit not as vulnerable as Node 15, presents the same lifetime extension, because it has only one neighbor: Node 15. Therefore, when Node 15 dies, Node 25 becomes isolated and does not forward any traffic. Thus, its lifetime extension remains the same as for Node 15.

Table I confirms the relation between node degree and lifetime extension. In general, nodes with high degrees such as Node 27 have important lifetime extension. However, when a high degree node experiences high collision rates, which is the case for example for Node 15, the lifetime extension may be lower; for instance, Node 15 has 8 neighbors, but with 60% of its energy drained during collisions, its lifetime extension is only 40%.

For some nodes, the lifetime extension measured at the *all-nodes* instant is smaller than that measured at the *first-node* instant. Such nodes forward a decreasing amount of traffic after the *first-node* instant, so their lifetime extension is smaller. For instance, at the *first-node* instant, Node 15 runs out of energy and Nodes 1, 14, and 29 become disconnected from the network and form an isolated subnetwork. From that instant, the only traffic going through this subnetwork comes from source node 1, which is smaller than the traffic before the *first-node* instant. Thus, the improvement observed afterwards decreases.

V. Implementation

A. Feasibility of MFP on Existing Radios

To show that MFP is feasible on existing radio modules, we have implemented MFP on the Chipcon CC2500 [20] evaluation board. The module contains a CC2500 chip, a short range low power radio transceiver with a 2.4 GHz modem. The transceiver is controlled by an 8051 low power 8 bits-24MHz microcontroller [21] with 2304B of RAM and 16KB of flash memory. The radio and the microcontroller communicate via a Serial Peripheral Interface (SPI).

The CC2500 already implements an efficient preamble sampling protocol called WOR (Wake On Radio). WOR is implemented on the radio chip to save the microcontroller the burden of periodically switching the radio between Receive and Sleep modes. This enables programmers to put the microcontroller in Idle (low power) mode while the radio is periodically sampling the channel.

We have implemented a simple flooding application and set the check interval to 300ms [6].

To evaluate energy savings, we have measured the instantaneous current consumption of the sensor node to identify in which mode it is operating. Table II shows the current drained for each mode.

Fig. 12 shows the current consumption during periodic channel sampling. Sampling is done by the radio module so that it is possible to put the microcontroller in Idle (low power) mode while sampling. The current drained to sample the channel is 14mA on average[7].

[6]CC2500 offers a limited set of check intervals, so we used the value of 300ms.

[7]The average values we measured are slightly larger than the typical values reported in the CC2500 data sheet. In the figure, peaks exceed 14mA because we measure instantaneous current.

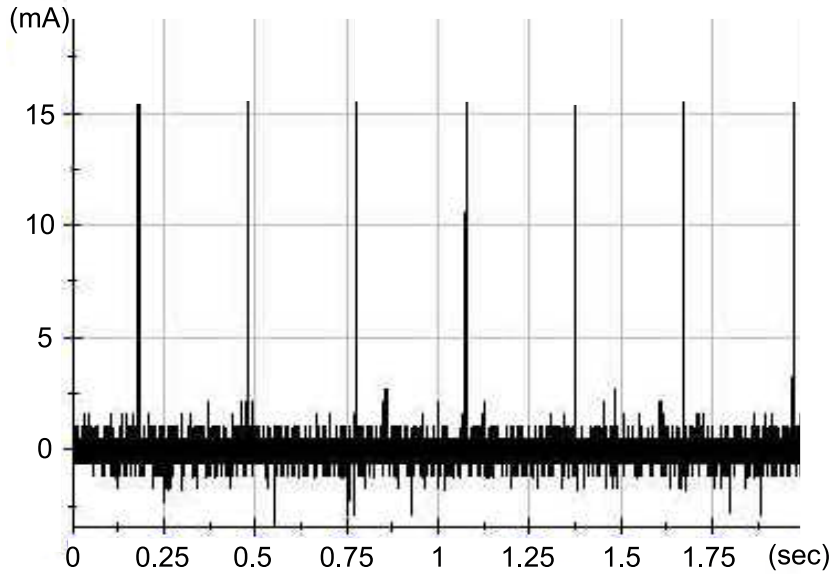


Fig. 12. Current consumption during periodic channel sampling done by the radio module. We have performed these current measurements only on the radio module.

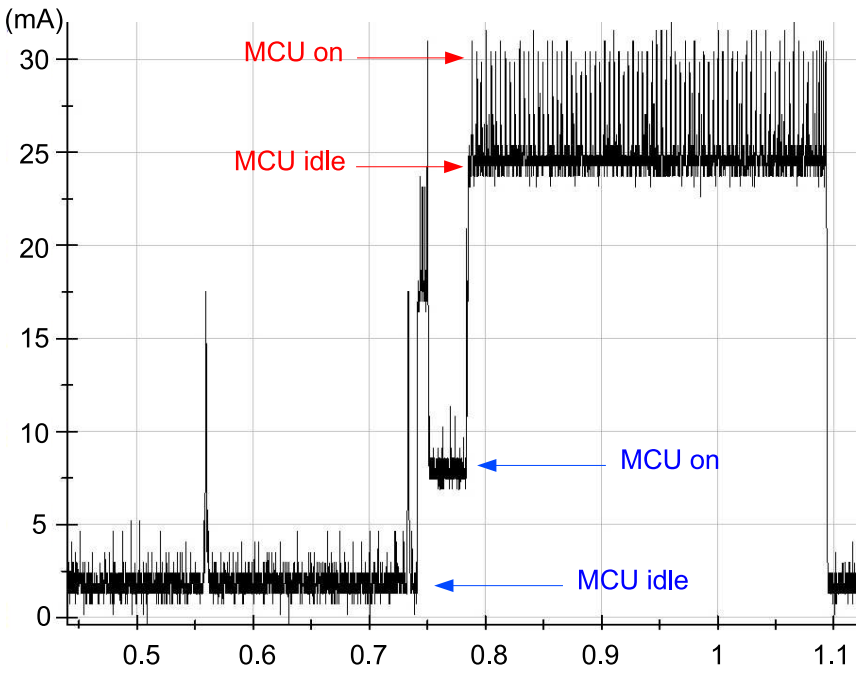


Fig. 13. Current consumption during frame forwarding using MFP taken at the radio and the microcontroller together.

TABLE II

MEASURED CURRENT CONSUMPTION OF CC2500

Radio (sleep)	900 nA
Radio (idle)	1.5 mA
Radio (transmit)	22 mA
Radio (receive)	14 mA
Microcontroller (active)	8 mA
Microcontroller (idle)	2 mA

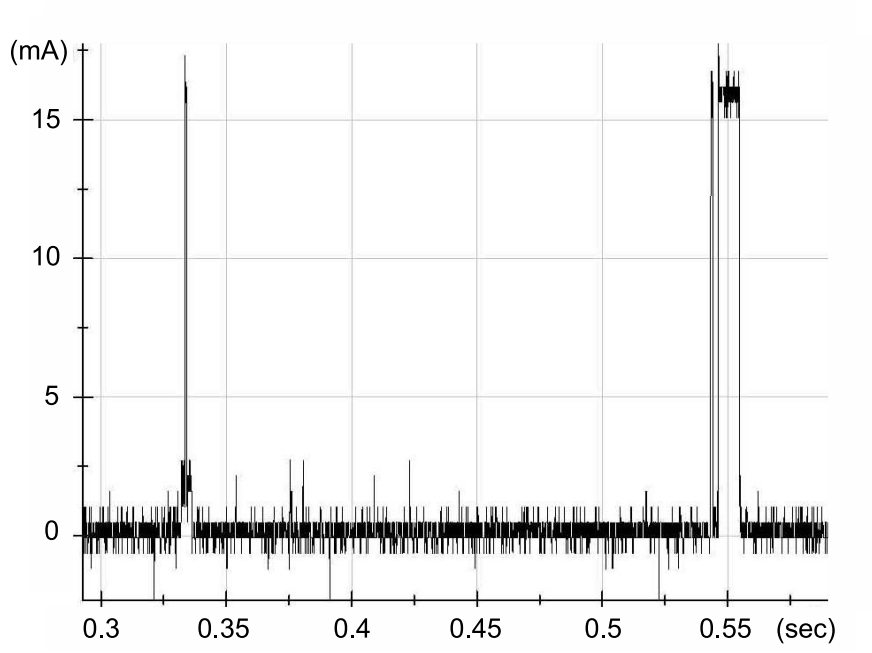


Fig. 14. Details of a data frame reception. Measurements are performed only on the radio.

Fig. 13 presents the current consumption during frame forwarding: the node receives a frame to be flooded and forwards it to its neighbors. Initially, the node periodically samples the channel. When the node wakes up to sample the channel at time 0.55s, it receives a microframe from which it learns about the arrival time of the data frame. As the node uses MFP, it switches its radio off until instant 0.74s to save energy. It then wakes up at 0.74 to receive the data frame. Because of potential clock drift, the node actually wakes up slightly ahead of the time for the data frame. During this wakeup, the node receives a second microframe and then the data frame. Fig. 14 shows some details: the reception of two microframes and the data frame. Note that time

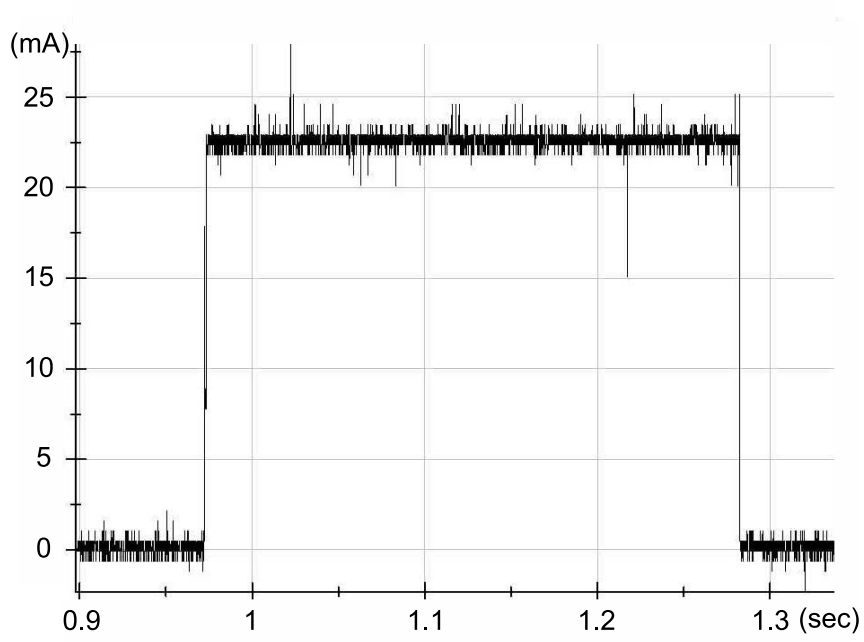


Fig. 15. Details of a transmission (microframes and a data frame). Measurements are performed only on the radio.

scales are different on each graph.

Our implementation has brought into focus the problem of the clock drift. This practical detail does not look important unless one tries to implement the MFP protocol. When a node receives a microframe indicating that the forthcoming data frame should be received, it calculates from the sequence number the sleep duration until the data frame. The clock needs to be precise enough to switch on the radio right on time to receive the data frame. The node should wake up early enough to make up for the clock drift and correctly receive the data frame. The accuracy of the oscillator used by the node determines the amount of the extra time the node spends in receive mode before the data frame arrives. There is a trade-off to make between using a high precision oscillator such as quartz crystal (ranging from 1ppm to 100ppm) or a lower precision oscillator such as a RC (e.g. ≈ 15000ppm [21]). A RC oscillator is less expensive and occupies less space than a quartz crystal. In our case, we use RC oscillator as it is integrated in the module.

After the reception of the data frame, the node should forward it to its neighbors. Before each message transmission (microframes plus data), the node chooses a random backoff to avoid collisions due to simultaneous access to the channel. Note that contrary to the IEEE 802.11 DCF [22], but similarly to the IEEE 802.15.4 CAP [23], the channel is not continuously sensed

during the backoff. The radio is switched off during the whole backoff period to save energy. When the backoff timer expires, the node performs carrier sense and transmits at once if the channel is free. Otherwise, if the channel is busy, the node re-executes the backoff procedure.

The backoff appears in Fig. 13 during the interval from 0.755s to 0.78s. The current drained during the backoff procedure is higher by 8mA, because we put the microcontroller in active mode during the backoff to show the difference between the microcontroller's active and idle modes on the same figure. After carrier sense, the node begins to transmit by first sending a series of microframes and then the data frame. During the transmission phase, the current drained by the node (radio plus microcontroller) oscillates between 24mA and 30mA. The value of 24mA corresponds to the radio in Transmit mode (22mA) (see Figure. 15) and the microcontroller in Idle mode (2mA), whereas the value of 30 corresponds to the radio in Transmit mode and the microcontroller in Active mode (8mA). In our implementation, we reduce the time during which the microcontroller is in Active mode; for example, the microcontroller fills in the transmission buffer of the radio with frames to be transmitted and goes back to Idle mode when the radio's transmission buffer if full. At the same time, the radio is reading its transmission buffer and transmitting frames. As the microcontroller fills the buffer (6MHz SPI) quicker than the radio is able to transmit (bandwidth 250kb/s), the microcontroller can go back to Idle mode while the radio is still transmitting frames.

B. Microcontroller Overhead

As shown in the previous section, the energy efficiency of MFP depends on the power consumed by the radio and on that consumed by the microcontroller. In this section, we determine the energy consumption distribution of these two components in the three operation modes of MFP: sampling, transmission, and reception. The transmission of a frame requires that the radio remains in transmit mode that consumes the highest amount of current (22mA) for a long time (microframes plus the data). Therefore, the radio transmission is the dominant energy consuming operation as shown in Fig. 16. The radio reception operation is made very small by our MFP protocol that saves nodes the reception of preamble until the data. The radio sampling is the smallest energy consuming operation, which is expected because it is the main objective of preamble sampling protocols.

For the microcontroller, the energy consumption of the three considered operations (sampling,

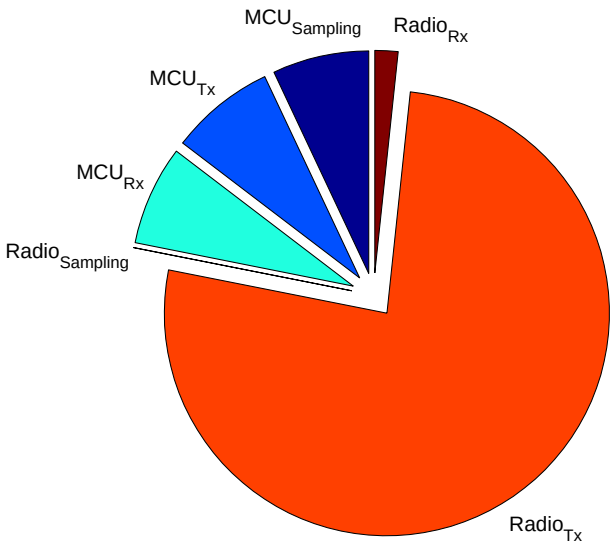


Fig. 16. Distribution of the energy consumption of both the radio and the microcontroller for the different MFP operations.

reception, and transmission) are almost equal, as shown in Fig. 16. The reason is that our implementation puts the microcontroller in Idle mode whenever possible, therefore the energy consumption of these operations is dominated by the consumption of the microcontroller's Idle mode. Therefore, reducing the energy consumption of the microcontroller's Idle mode substantially increases energy saving and thus nodes lifetime.

C. Possible Improvements

The MFP protocol has three major states: channel sampling, transmission, and reception. The implementation of MFP on the CC2500 drastically minimizes the microcontroller overhead by putting it in Idle mode whenever possible.

For the channel sampling operation, the microcontroller goes to Idle mode (or to any other low power mode) while the radio is performing periodic wake up according to a predefined check interval. Thus, the microcontroller does not consume any more energy in channel sampling mode than in Idle mode.

The reception operation starts when the radio receives a frame while periodically sampling the channel. Upon this reception, the radio wakes the microcontroller up to process the received

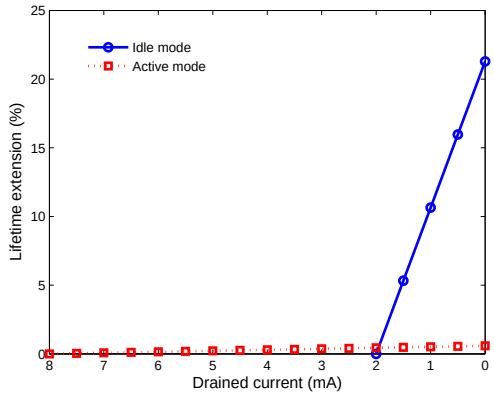


Fig. 17. The percentage of lifetime extension in function of different values for the current drained by the microcontroller in Idle and Active modes. Reducing the current drained in the Active mode only offers a negligible gain, whereas reducing that of the Idle mode significantly increases the lifetime.

frame. If the microcontroller finds a microframe, then it checks whether the subsequent data is relevant or not. If so, the microcontroller goes back to Idle and wakes up later only to receive the data.

For the transmission, the microcontroller constructs microframes and sends them to the radio's transmission buffer. At the same time, the radio reads data from the buffer and sends them to the air interface. As the microcontroller is faster than the radio, the buffer may be full during some periods. When the buffer is full, the microcontroller goes back to Idle mode. When the buffer is almost empty, the radio wakes the microcontroller up so that it continues to fill the buffer with frames to be transmitted.

In all these modes, the microcontroller is in low power mode most of the time especially when the traffic load is low. Fig. 17 shows that reducing the current drained in the Active mode only offers a negligible gain, whereas reducing that of the Idle mode significantly increases the lifetime. Therefore, it is better for MFP to use microcontrollers with a minimum power consumption Idle mode. This result emphazises the negligible overhead of calculating microframes by the microcontroller, therefore there is no need to add dedicated hardware for calculating microframes.

VI. Related Work

In this section, we focus on contributions related to preamble sampling. We show that MFP can replace these contributions in some situations and can be used jointly with them in other situations. In both cases, MFP increases the amount of energy savings.

BMAC [11] proposes an outlier detection based technique to improve the accuracy of CCA (Clear Channel Assessment). An accurate CCA has two major benefits. First, it reduces the number of false assessments (e.g. the channel is assessed to be busy while it is actually clear), which increases channel utilization and thus the network throughput. Second, it allows a node to accurately detect whether the channel is still active during a presumed preamble reception, which reduces the duration of receiving false preambles and thus increases energy savings. The accurate CCA operation of BMAC is compatible with MFP whether is it persitent in reception or not. If MFP is persistent is reception, i.e. a node persists in receiving until it receives a microframe or the channel is back to idle, an accurate CCA allows a node to detect when the channel is back to idle so that it stops receiving at the right time. If MFP is non-persitent in reception, i.e. a node does not persist in reception but gives up if it fails to receive a microframe within a certain timeout value, an accurate CCA only increases channel utilization (throughput) as an accurate CCA is not needed to detect false preambles.

El-Hoiyidi *et al.* [24] tackled this issue and proposed a way to reduce the preamble length for unicast frames. Although such a scheme contributes to saving energy both at transmitter and receiver, it did not propose any solution to avoid receiving the preamble all the way until data reception. Moreover, their proposal only applies to unicast frames and cannot be generalized to broadcast frames.

WiseMAC [10] and SCP [13] reduce the preamble length by using short preambles for unicast frames. As they still use preambles, albeit not always of a full-length, microframes can replace the continuous preambles, thus achieving further energy savings.

Protocols that use preambles split into frames with a gap between consecutive frames, such as STEM-B [14], CSMA-MPS [25], TICER [12], WOR [20] and X-MAC [26], have the advantage of not always needing the full-length preamble; in the case of unicast transmissions, the receiver sends the ACK in the gap between the frames to stop the preamble transmission by the transmitter. However, in very lightly loaded networks, these protocols do not guarantee optimal energy

savings, because they increase idle listening at receivers. Indeed, when there is a gap between frames, a receiver should remain in Receive mode for a duration that is larger than the gap to sample the channel. The sampling duration therefore increases and nodes waste more energy in sampling. MFP avoids this long sampling duration by sending microframes without gaps. MFP is more suitable for low data rate networks.

Z-MAC [27] is a hybrid protocol that combines the strengths of CSMA and TDMA. Under low contention, Z-MAC switches to CSMA to achieve high channel utilization and low delays. Under high contention, Z-MAC switches to TDMA to achieve high channel utilization, fairness, and less collisions. Z-MAC is implemented in TinyOs [28] on top of B-MAC. Therefore, it can be used with MFP that is compatible with B-MAC. MFP improves energy savings of B-MAC, therefore it may do the same for Z-MAC.

VII. Conclusions

Protocols based on preamble sampling techniques are being increasingly used in wireless sensor networks because of their significant energy savings compared to other protocols. In this paper, we have shown that energy consumption can be further reduced by reducing irrelevant receptions. Our technique, called MFP (Micro Frame Preamble), replaces the continuous long preamble by a series of micro-frames containing a digest (an indicator of the data frame contents) and a sequence number. This information enables a node to switch off the radio and wake up only for the reception of a relevant data frame. MFP is not another MAC protocol for sensor networks; rather, it is a generic technique that can be used under various preamble protocols including, WiseMAC, LPL, Z-MAC, SCP and others.

Throughout the paper, we have presented an extensive evaluation of MFP. We have derived analytical formula for lifetime extension obtained with MFP, and performed ns-2 simulations to take other parameters, such as collisions, into account. We have shown that MFP can be successfully implemented on existing radio modules, such as the CC2500, without hardware modifications. We have also shown that the overhead of computing and passing microframes to the radio chip is negligible so that MFP can be implemented on a microcontroller with a limited processing power.

REFERENCES

[1] W. Ye, J. Heidemann, and D. Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. *In Proceedings of IEEE Infocom*, pages 1567–76, New York, NY, July 2002.

[2] Ember Corporation. EM250 Single-Chip ZigBee/802.15.4 Solution, Data Sheet. 2005.

[3] Freescale Semiconductor. MC13192/MC13193 2.4 GHz Low Power Transceiver for the IEEE 802.15.4 Standard. *Rev. 2.8*, 04/2005.

[4] J. Polastre, R. Szewczyk, D. Culler. Telos: Enabling Ultra-Low Power Wireless Research. *In Proceedings of IPSN/SPOTS*, pages 302–11, Los Angeles, CA, April 2005.

[5] A. Bachir, D. Barthel, M. Heusse and A. Duda. Abstract Frames for Reducing Overhearing in Wireless Sensor Networks. *In Proceedings of the IFIP Networking*, Coimbra, Portugal, May 2006.

[6] A. Bachir, D. Barthel, M. Heusse and A. Duda. Micro-Frame Preamble MAC for Multihop Wireless Sensor Networks. *In Proceedings of IEEE ICC*, Istanbul, Turkey, June 2006.

[7] K. Langendoen and G. Halkes. Energy-Efficient Medium Access Control. *Book chapter in the Embedded Systems Handbook, R. Zurawski (editor), CRC press*, 2005.

[8] I. Demirkol, C. Ersoy, and F. Alagoz. MAC Protocols for Wireless Sensor Networks: a Survey. *IEEE Communications Magazine*, 44(4):115–21, April 2006.

[9] T. van Dam and K. Langendoen. An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. *In Proceedings of ACM Sensys*, pages 171–80, Los Angeles, CA, November 2003.

[10] C. Enz, A. El-Hoiydi, J. Decotignie, V. Peiris. WiseNET: An Ultralow-Power Wireless Sensor Network Solution. *IEEE Computer*, 37(8):62–70, August 2004.

[11] J. Polastre, J. Hill and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. *In Proceedings of ACM SenSys*, 2004.

[12] E-Y. Lin, J. Rabaey, A. Wolisz. Power-Efficient Rendez-vous Schemes for Dense Wireless Sensor Networks. *In Proceedings of IEEE ICC*, Paris, France, June 2004.

[13] W. Ye, F. Silva and J. Heidemann. Ultra-Low Duty Cycle MAC with Scheduled Channel Polling. *In Proceedings of ACM SenSys*, Boulder, CO, November 2006.

[14] C. Schurgers et al. Optimizing Sensor Networks in the Energy-Latency-Density Design Space. *IEEE Transactions on Mobile Computing*, 1(1):70–80, January-March 2002.

[15] A. El-Hoiydi, J-D. Decotignie, C. Enz, E. Le Roux. Poster Abstract: WiseMac, an Ultra Low Power MAC Protocol for the WiseNET Wireless Sensor Networks. *In proceedings of ACM SenSys*, pages 302–3, Los Angeles, CA, November 2003.

[16] A. El-Hoiydi, and J-D. Decotignie. Low Power Downlink MACProtocols for Infrastructure Wireless Sensor Networks. *ACM Mobile Networks and Applications (MONET)*, 10(5):675–90, 2005.

[17] X. Shi, G. Stromberg, Y. Gsottberger, and T. Sturm. Wake-Up- Frame Scheme for Ultra Low Power Wireless Transceivers. *In Proceedings of IEEE Globecom*, Dallas, TX, November 2004.

[18] X. Shi, and G. Stromberg. SyncWUF: An Ultra Low-Power MAC Protocolfor Wireless Sensor Networks. *IEEE Transactions on Mobile Computing, year = 2007, volume = 6, number = 1, pages = 115-25, month = January,*.

[19] http://www.isi.edu/nsnam/ns/.

[20] Chipcon Corporation. CC2500 Single Chip Low Cost Low Power RF Transceiver, Data Sheet. 2005.

[21] Silicon Laboratories. C8051F320/1 Datasheet. *Preliminary Rev. 2.1*, 12/2003.

[22] IEEE 802.11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. 1999.

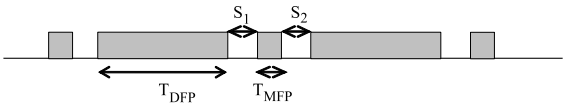


Fig. 18. Zebra Frame Preamble

[23] IEEE 802.15.4. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). 2003.

[24] A. El-Hoiydi and J. Decotignie and J. Hernandez. Low Power MAC Protocols for Infrastructure Wireless Sensor Networks. *In Proceedings of European Wireless*, Barcelona, Spain, February 2004.

[25] S. Mahlknecht and M. Boeck. CSMA-MPS: A Minimum Preamble Sampling MAC Protocol for Low Power Wireless Sensor Networks. *In Proceedings of IEEE Workshop on Factory Communication Systems*, Vienna, Austria, September 2004.

[26] M. Buettner et al. X-MAC: A Short Preamble MAC Protocol For Duty-Cycled Wireless Networks. *In Proceedings of ACM SenSys*, Boulder, CO, November 2006.

[27] I. Rhee et al. Z-MAC: a Hybrid MAC for Wireless Sensor Networks. *In Proceedings of ACM SenSys*, San Diego, CA, November 2005.

[28] J. Hill et al. System Architecture Directions for Networked Sensors. *ACM SIGOPS Operating Systems Review*, 34(5):93–104, 2000.

APPENDIX

We follow the same methodology used in Section III-A2 to derive the mean time a node spends in Receive mode. As in DFP, in ZFP a node does not need to wake up again to receive the data frame transmitted at the end. It is sufficient for the node to receive a DFP frame in the preamble. In ZFP, there are 4 intervals during which a node may wake up to sample the channel. These intervals are: I_{MFP}, I_{S_1}, I_{DFP}, and I_{S_2}. Therefore, we define the following:

According to the analysis presented in Section III-A and to Fig. 18, we have:

$$\begin{cases} q_{S_1} &= \frac{S_1}{S_1+S_2+T_{\mathrm{MFP}}+T_{\mathrm{DFP}}} \\ q_{S_2} &= \frac{S_2}{S_1+S_2+T_{\mathrm{MFP}}+T_{\mathrm{DFP}}} \\ q_{\mathrm{DFP}} &= \frac{T_{\mathrm{DFP}}}{S_1+S_2+T_{\mathrm{MFP}}+T_{\mathrm{DFP}}} \\ q_{\mathrm{MFP}} &= \frac{T_{\mathrm{MFP}}}{S_1+S_2+T_{\mathrm{MFP}}+T_{\mathrm{DFP}}} \end{cases} \tag{28}$$

and,

$$\begin{cases} \mu_{S_1}(t) & = & \tau + U_{[0,S_1]}(t) + a + \alpha(\tau + T_{\mathrm{DFP}}) \\ \mu_{\mathrm{MFP}}(t) & = & \tau + U_{[0,T_{\mathrm{MFP}}]}(t) + S_2 + T_{\mathrm{DFP}} \\ \mu_{S_2}(t) & = & \tau + U_{[0,S_2]}(t) + T \\ \mu_{\mathrm{DFP}}(t) & = & \tau + U_{[0,T_{\mathrm{DFP}}]}(t) + S_1 + T_{\mathrm{MFP}} + \alpha(\tau + T_{\mathrm{DFP}}) \end{cases} \tag{29}$$

Therefore, the reception duration in ZFP is:

$$T^r_{\mathrm{ZFP}}(t) = q_{S_1}\mu_{S_1}(t) + q_{\mathrm{MFP}}\mu_{\mathrm{MFP}}(t) + q_{S_2}\mu_{S_2}(t) + q_{\mathrm{DFP}}\mu_{\mathrm{DFP}}(t) \tag{30}$$

For the sake of simplicity, we assume that $S_1 = S_2$. Therefore, the average reception duration $E[T^r_{\mathrm{ZFP}}(t)]$ is:

$$\begin{aligned} E[T^r_{\mathrm{ZFP}}(t)] = \tau + \frac{1}{2(2T_S + T_{\mathrm{MFP}} + T_{\mathrm{DFP}})}\Big\{ & \\ & T_S(T_S + 2T_{\mathrm{MFP}} + 2\alpha(\tau + T_{\mathrm{DFP}})) + \\ & T_{\mathrm{MFP}}(T_{\mathrm{MFP}} + 2T_S + 2T_{\mathrm{DFP}}) + T_S(T_S + 2T_{\mathrm{DFP}}) + \\ & T_{\mathrm{DFP}}(T_{\mathrm{DFP}} + 2T_S + 2T_{\mathrm{MFP}} + 2\alpha(\tau + T_{\mathrm{DFP}}))\Big\} \end{aligned} \tag{31}$$