Link Cost and Reliability of Frame Preamble MAC Protocols

Abdelmalik Bachir, Ludovic Samper and Dominique Barthel
France Telecom R&D
Meylan, France
{Abdelmalik.Bachir, Ludovic.Samper, Dominique.Barthel}@orange-ft.com

Martin Heusse and Andrzej Duda
LSR-IMAG Laboratory
Grenoble, France
{Martin.Heusse, Andrzej.Duda}@imag.fr

***Abstract*— Previous studies have shown that preamble MAC protocols have higher energy efficiency compared to traditional low power MAC protocols based on common sleep/wakeup schedules. One efficient implementation of preamble MAC protocols is Frame Preamble MAC in which the MAC preamble is replaced by a series of frames. In this paper, we consider four Frame Preamble MAC protocols, namely: persistent MFP (Micro Frame Preamble), non persistent MFP, persistent DFP (Data Frame Preamble), and non persistent DFP. We study their energy cost and communication reliability by assuming a simple binary symmetric channel error model.**

I. INTRODUCTION

We consider wireless sensor networks composed of a large number of battery operated nodes. Nodes share a common radio channel and are organized as a multihop ad hoc network — communication between nodes requires relaying packets by intermediate nodes. The medium access control (MAC) sets up rules for using the common channel. In sensor networks, the MAC protocol should minimize the energy consumption, because such networks should be long-lived without battery replenishment or replacement.

Previous studies have shown that the standard IEEE 802.11 DCF protocol [1], which is widely used in multihop ad hoc networks, is unsuitable for sensor networks because of considerable energy consumption in idle listening [2]: idle listening 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. As a node may waste considerable energy when the radio is on [3]–[5], many protocols propose to repeatedly put nodes into sleep mode (radio off) to mitigate idle listening [6], [7].

The use of sleep periods for mitigating the energy waste due to idle listening implies the need for a method to know when other nodes potentially transmit. There are two ways for avoiding message loss when a node transmits a frame while the destination node is sleeping. In the first approach used in protocols like SMAC [2], TMAC [8], and others, nodes synchronize on a common sleep/wakeup schedule by exchanging synchronization messages to set their sleep/wakeup schedule. The second approach used in protocols like WiseMAC [9], BMAC [10], and others like [11], [12], does not define a common schedule for sleep and wakeup periods to avoid synchronization overhead and to further reduce idle listening in low-rate data networks. In the second approach, each node chooses its own sleep/wakeup schedule independently of the others and a node transmits a preamble before each data frame, which is long enough to make sure that all potential receivers will get their data. Hereafter, the protocols using the second approach are referred to as Preamble Protocols.

Preamble Protocols are well suited to lightly loaded energy-limited sensor networks as they can save more energy compared to common sleep/wakeup protocols [9], [10], [13]. However, they also present some drawbacks that come from the overhead of preamble transmission/reception. In a previous work [14], we have considered this problem and proposed MFP (Micro-Frame Preamble) to reduce the overhead of preamble reception: we replace the preamble by a series of small frames called Micro-Frames. We have shown that the use of micro-frames significantly increases the energy savings at the receiver in an error free channel.

Besides minimizing energy consumption, a good MAC protocol should also provide reliable communication between neighbor nodes. However, these are antagonistic requirements in general, because reliability may increase energy consumption. In this paper, we study energy cost and communication reliability of Preamble Protocols by assuming a simple binary symmetric channel error model. As a representative for Preamble Protocols we use Frame Preamble MAC in which we replace the preamble that precedes the data frame by a series of frames. This can be a series of micro-frames, in which case we call the protocol micro-frame preamble (MFP), or a series of duplicate copies of the data frame, in which case we call the protocol data-frame preamble (DFP). We combine these two transmission schemes with two reception schemes, which can be either persistent and not, to consider four protocols: persistent-MFP, persistent-DFP, non-persistent-MFP, and non-persistent-DFP.

II. FRAME PREAMBLE MAC PROTOCOLS

A. Overview of traditional Preamble protocols

In Preamble MAC protocols, nodes do not share a global sleep/wakeup schedule: each node chooses its sleep/wakeup schedule independently of the others. As shown in Fig. 1, a node wakes up periodically and senses the channel for a short time to check whether there is any signal. If the node detects a signal being transmitted then it stays active trying to receive

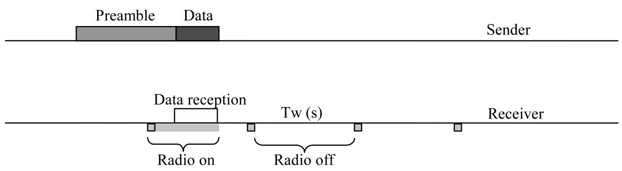


Fig. 1. Preamble sampling technique

the data frame that follows. If a node wants to transmit a data frame then it sends a preamble first. The goal of the preamble is to make sure that the receiver will be awaken and receives the data frame. To be effective, the preamble needs to be as long as the check interval, which is the period between two consecutive instants of node wakeups.

B. Motivation of Frame Preamble protocols

The main drawback of Preamble MAC protocols is that the node that detects the preamble keeps listening until it receives the data (Fig. 1). This is because the preamble does not convey any information about when the data will be transmitted. Some protocols try to take advantage of this listening period to estimate noise floor. Although, this helps the node to perform efficient CCA (Clear Channel Assessment), we argue that this reception is not really needed, in particular when the subsequent data is irrelevant. An irrelevant frame is a frame that is not received by the node of interest.

To overcome this drawback, we propose to put additional information in the preamble so that a node makes a timely decision, i.e. without keeping listening to the preamble until the data. We achieve this by replacing the traditional preamble by a series of frames that can be decoded at the receiver. We call the resulting scheme *Frame Preamble* MAC. The Frame Preamble protocol is a generalization of the Micro-Frame Preamble (MFP) introduced previously [14].

In this paper, we propose four variants of Frame Preamble MAC protocols. These variants come from the different strategies nodes may use when they transmit and receive frames. We discuss the four variants, which are a combination of two transmission and two reception schemes, in the two following Sections II-C and II-D.

C. Transmission Schemes: MFP vs. DFP

The frames used in Frame Preamble MAC can be either Micro-Frames carrying some information about the data frame or simply duplicate copies of the data frame itself.

In the MFP (Micro Frame Preamble), instead of a preamble, a node transmits micro-frames that contain information about the subsequent data frme. In this way, the node that wakes up to check the channel receives a micro-frame from which it learns when the data will be transmitted and whether this data is worth receiving (relevant or not). Specifically, each micro-frame contains a *sequence number*, a *destination address*, and a *hash* of the data payload. The sequence number indicates the number of the remaining micro-frames transmitted before the data frame. The node can then deduce when the data frame will be transmitted, so it may switch its radio off to avoid wasting energy in receiving subsequent micro-frames. The node uses the destination address field to avoid waking up to receive subsequent unicast data frames addressed to another node. The node uses the hash field to identify redundant broadcast frames and avoid receiving them again. The node stores the hash of each broadcast frame it receives in a table. In this way, the node knows that the subsequent broadcast frame is redundant if its table contains already the same hash value. More details on the protocol operation can be found elsewhere [14].

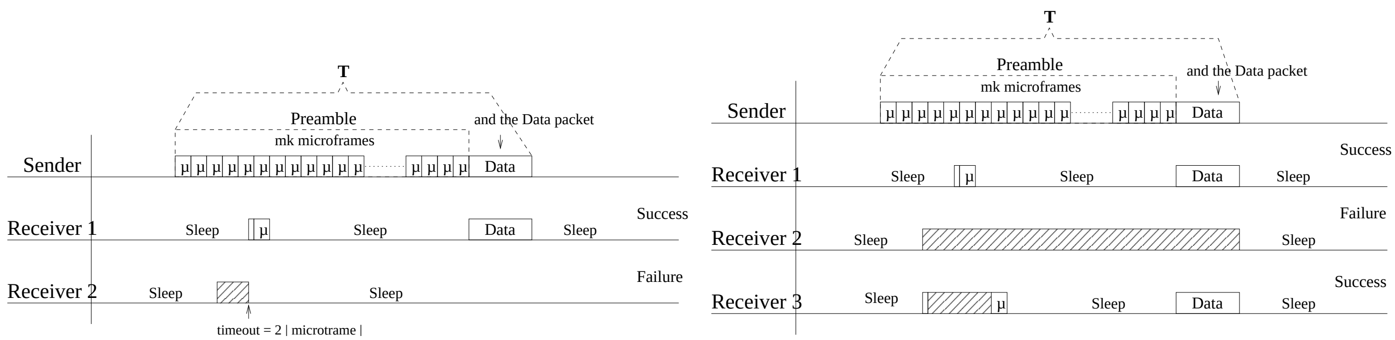
In the DFP (Data Frame Preamble), the frames transmitted instead of the preamble are duplicate copies of the data. The advantage of DFP is that the node that wakes up to check the channel immediately receives the data, so it does not need to wake up again to receive the data. It also has another advantage in the sense that duplicating the data in preamble frames increases the reliability of the transmission (as shown in Section IV-E). However, in DFP the node cannot avoid receiving irrelevant data, which may consume non-negligible energy if the reception time of data is large. Typically, this happens when large data frames are transmitted at low bandwidth.

D. Reception Schemes: Persistent vs. Non Persistent

In Preamble Protocols, a node wakes up periodically each check interval to sense the state of the channel. The time needed for doing this is the sampling time which duration depends on the technique used in preamble transmission. It should be large enough so that the node is able to decode the information being transmitted on the channel. For example, if the technique used in transmission is MFP, then the sampling time is at least equal to one micro-frame transmission time. This is the minimum sampling time to decode a micro-frame. Only when a node correctly decodes a micro-frame, it can know when the data frame will be transmitted and whether it is relevant.

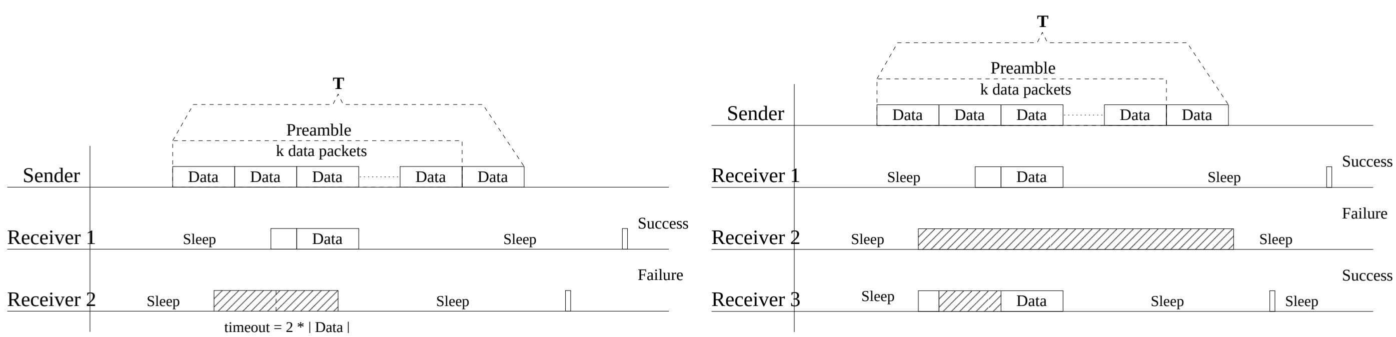
In general, a node needs more than the minimum sampling time to correctly decode a micro-frame. The sampling time depends upon the instant of node wakeup and the quality of the radio link. If we assume that radio links are perfect then the maximum sampling time a node needs is equal to twice the micro-frame transmission time. This happens when the node fails to receive a micro-frame because it has missed its first bit. In this case, the node should keep receiving until it decodes the subsequent micro-frame.

However, if the radio link is not perfect, then transmission errors may occur and the node may even fail to receive the subsequent micro-frame. In this case, the node has two options: either (1) to be *persistent* and continue receiving until it decodes a frame or the channel becomes idle, or (2) to be *non-persistent* and stop receiving after a timeout value, which is twice a micro-frame transmission time. Fig. 2 shows an example of persistent MFP (Fig. 2(b)), and non-persistent MFP (Fig. 2(a)). Note that the notion of persistence applies to DFP as well: we therefore consider two other protocols: persistent DFP (Fig. 3(b)) and non-persistent DFP (Fig. 3(a)).



(a) Non Persistent (b) Persistent

Fig. 2. MFP



(a) Non Persistent (b) Persistent

Fig. 3. DFP

Our intuition is that the non persistent methods should be used in channels with bursty errors or with high error rates. In bursty channels, the node goes back to sleep to avoid keeping receiving during the burst of errors and wakes up later on to sample the channel again. In channels with high error rates, the non persistent method saves the node the energy of keeping receiving without success as the probability of a correct reception is low. The persistent method is efficient in channels with low error rates. Persisting in reception under these circumstances saves the transmitter the cost of retransmitting.

III. System Model

A. Problem Statement and Assumptions

We consider a wireless link between two nodes. We want to find the energy cost of transmitting one data frame over this link and estimate the corresponding reliability. The energy cost of the link is equal to the amount of energy drained at transmitter plus that of the receiver. We define reliability as the probability that the receiver correctly decodes the data frame. Both the receiver and the transmitter use a Frame Preamble MAC. For the analysis in this paper, we only consider the four protocols described earlier in Sections II-C and II-D. The goal of this study is to identify the best protocol to use for the required reliability in given channel conditions.

We assume a Binary Symmetric Channel (BSC) in which each bit has a constant and independent error probability. We call p the probability that a micro-frame is corrupted. We assume that a micro-frame has a unit size and its transmission time also has a unit duration. For the sake of simplicity, we express our results in function of these unities. We assume that the size of data frames is m times larger than that of micro-frames and there are $m \times k$ micro-frames in the preamble, where k is some constant. As we use micro-frames as unit sizes and unit durations, the transmission of a data frame also has the duration of m and the check interval the duration of mk (see also Fig. 2). We assume that DFP protocols send k data frames in the preamble (see Fig. 3). Constant k enables us to relate two types of protocols and to compare different variants.

B. Reliability

In Frame Preamble protocols, the receiver sends an ACK message back to the transmitter right after data transmission to acknowledge a successful reception. If the transmitter does not receive the ACK, then it retransmits again until it receives the ACK or the maximum number of transmissions is reached. Each retransmission includes the whole Frame Preamble plus the data. We call n the maximum number of transmissions and p_f the probability of a failed single transmission. Therefore, the reliability R, which is the probability of a successful communication in n single transmissions is

$$R = 1 - p_f^n. \quad (1)$$

C. Transmission Cost

The energy drained in transmission is proportional to the amount of time the transmitter spends in transmit mode. It is also proportional to the current and voltage but for the sake of simplicity, we assume these to be constant. We distinguish between a *transmission* and a *single transmission*. A single transmission consists of the transmission of the preamble and the data whereas a transmission consists of several single (re)transmissions. We call T the duration of a single transmission:

$$T = mk + m \tag{2}$$

We call T_{tx} the duration of a transmission:

$$\begin{aligned} T_{tx} &= (1-p_f)T + p_f(1-p_f)2T + \cdots + \\ &\quad + p_f^{n-2}(1-p_f)(n-1)T + p_f^{n-1}nT \\ &= (1-p_f)\sum_{i=1}^{n-1} p_f^{i-1} iT + p_f^{n-1}nT \\ &= \frac{1-p_f^n}{1-p_f}T \end{aligned} \tag{3}$$

D. Reception Cost

We follow the same methodology as in Section III-C to derive the average time the receiver spends in receive mode. Let S (resp. F) be a random variable that expresses the time the receiver spends in receive mode in case of successful (resp. failed) single transmission. Therefore, the reception duration T_{rx} is

$$\begin{aligned} T_{rx} &= (1-p_f)S + (1-p_f)p_f[F+S] + \cdots \\ &\quad + (1-p_f)p_f^{n-1}[(n-1)F + S] + p_f^n nF \\ &= (1-p_f)\left(\sum_{i=0}^{n-1} p_f^i [iF + S]\right) + p_f^n nF \\ &= \frac{1-p_f^n}{1-p_f}[p_f F + (1-p_f)S] \end{aligned} \tag{4}$$

IV. Evaluation

For the evaluation of Frame Preamble protocols, we find the values of p_f, F, and S for each protocol variant: non-persistent-DFP, non-persistent-MFP, persistent-DFP, and persistent-MFP.

A. Non-Persistent DFP

In non-persistent DFP, the timeout values are twice the data transmission time. Thus, to correctly receive a single transmission, the receiver must correctly decode the data frame following its wakeup instant. Let us call q the probability that a data frame is corrupted:

$$q = 1 - (1-p)^m \tag{5}$$

Therefore,

$$p_f = q \tag{6}$$

In Preamble Protocols the receiver may wake up to sample the channel at any instant during preamble transmission. Specifically in DFP, the receiver may wake up at any instant during data frame preamble transmission. If the receiver misses the reception of the first bit of data frame preamble, then it keeps receiving until it catches the first bit of the subsequent data frame. This time is equal to U_m, which is a uniform random variable in $[0, m[$. For a successful single transmission, if the receiver does not correctly decode a data frame, because it has missed the first part of it, then the receiver must correctly decode the subsequent data frame. Therefore, we have:

$$S = U_m + m \tag{7}$$

However, in a failed single transmission, the reception duration F depends on the wakeup instant of the receiver. If the receiver wakes up during the last data frame of the preamble then it must fail to decode the subsequent data frame. In this case, the receiver goes back to sleep before timeout expires because the channel is back to idle before. In this case F is equal to $U_m + m$. However, if the receiver wakes up before the transmission of the last data frame of the preamble, then it goes back to sleep again when timeout expires. In this case, F is equal to $2m$. The probability that the receiver wakes up during the last data frame of the preamble is $1/k$. Therefore, we have:

$$F = \frac{k-1}{k} \times 2m + \frac{1}{k} \times (U_m + m) \tag{8}$$

B. Non-persistent MFP

To correctly receive a single transmission in non-persistent MFP, the receiver must correctly decode the micro-frame following its wakeup instant and correctly decode the data frame. Therefore, we have:

$$\begin{aligned} p_f &= 1 - (1-p)(1-q) \\ &= 1 - (1-p)^{m+1} \end{aligned} \tag{9}$$

For the sake of simplicity, we do not consider the case in which the receiver wakes up during the transmission of the last micro-frame. This means that the receiver either correctly decodes a micro-frame or its timer expires during the preamble for the lack of correctly decoding a micro-frame. We use this assumption, because it simplifies the analysis with negligible effects on the results. Therefore, in the case of a successful single transmission, the time S is equal to the duration of receiving a correct micro-frame plus that of receiving a correct data frame, which is:

$$S = U_1 + 1 + m \tag{10}$$

where U_1 is a uniform random variable in $[0, 1[$.

A single transmission fails, because either the receiver fails to receive a micro-frame in the preamble so that it does not wake up to catch the data frame, or the receiver correctly decodes a micro-frame, but fails to correctly receive the data. The probability of the first case is p. Therefore, we have:

$$F = p \times 2 + (1-p) \times (U_1 + 1 + m) \tag{11}$$

C. Persistent DFP

In persistent DFP, a single transmission fails, if the receiver fails to receive all data frames before the channel is back to idle. This includes all data frames of the preamble and the data frame. As its wakeup instant is random, the receiver may miss the reception of j data frames, where j is in $1, \ldots, k$. For example, if the receiver wakes up during the first data frame of the preamble, then it may keep listening during all the $k-1$ subsequent frames of the preamble plus the data. In this case, the number of missed frames is equal to k. The probability that the receiver wakes up during the transmission of any preamble frame is $1/k$. Therefore, we have:

$$\begin{aligned} p_f &= \frac{1}{k}q^k + \frac{1}{k}q^{k-1} + \cdots + \frac{1}{k}q \\ &= \frac{q}{k}\left(\frac{1-q^k}{1-q}\right) \end{aligned} \quad (12)$$

To find the distributions of F and S, we introduce $X \in \{0, \ldots, k-1\}$, a discrete random variable that expresses the number of corrupted data frames received during preamble transmission. We have:

$$X = (X|_{failure})p_f + (X|_{success})(1-p_f). \quad (13)$$

where, $X|_{failure}$ (resp. $X|_{success}$) is a discrete random variable that expresses the number of received corrupted frames in the preamble knowing that the single transmission failed (resp. succeeded). Therefore, we express F and S as:

$$F = U_m + m \times (X|_{failure}) + m \quad (14)$$
$$S = U_m + m \times (X|_{success}) + m \quad (15)$$

To obtain $P[X=j]$ for $j = 0, \ldots, k-1$, we use the following relation:

$$P[X=j] = P[X \geq j] - P[X \geq j+1] \quad (16)$$

To find $P[X \geq j]$, we introduce $Z \in \{1, \ldots, k\}$, a random variable that expresses the position of the frame during which the receiver wakes up. For example, if the receiver wakes up during the transmission of the first data frame of the preamble, which has position 1, then $Z = 1$. Z is uniform, i.e. $P[Z = j] = 1/k$. Thus, we have:

$$\begin{aligned} P[X \geq j] &= \sum_{i=1}^{k} P[X \geq j | Z = i] P[Z = i] \\ &= \underbrace{\frac{1}{k}q^j + \cdots + \frac{1}{k}q^j}_{\text{if the receiver wakes up before position } k-j} \\ &\quad + \underbrace{\frac{1}{k}0 + \cdots + \frac{1}{k}0}_{\text{otherwise}} \\ &= \frac{k-j}{k}q^j \end{aligned} \quad (17)$$

Therefore

$$\begin{aligned} P[X=j] &= \frac{k-j}{k}q^j - \frac{k-(j+1)}{k}q^{j+1} \\ &= \frac{q^j}{k}\Big[k(1-q) + q - j(1-q)\Big] \end{aligned} \quad (18)$$

Now, let us express $X|_{failure}$. $X|_{failure} \in \{0, \cdots, k-1\}$. $X|_{failure}$ is uniform in $[0, k-1]$. This is because the probability that the receiver keeps listening during j data-frames knowing that the single transmission fails is exactly the probability that the receiver wakes up during the transmission of the preamble data frame of position $j+1$. This probability is $1/k$. Hence $P[X|_{failure} = j] = 1/k$. Therefore, we deduce $X|_{success}$, which is:

$$X|_{success} = \frac{X - (X|_{failure})p_f}{1-p_f} \quad (19)$$

D. Persistent MFP

The probability of a successful single transmission in case of persistent MFP depends only on the data frame that follows the micro frames. We have,

$$p_f = q \quad (20)$$

To calculate F and S, we introduce $Y \in \{0, \cdots, mk-1\}$, which is a random variable that expresses the number of received micro-frames, corrupted or not:

$$F = U_1 + Y + m \quad (21)$$
$$S = U_1 + Y + m \quad (22)$$

U_1 stands for the part of the micro-frame the receiver has to listen to until the beginning of the next micro-frame and m is the duration of the data frame. Note that Y does depend on the success of the reception as the last one depends only on the data frame. Therefore, $F = S$ as shown in the two equations above. To find the distribution of Y, we use the following relation:

$$P[Y=j] = P[Y \geq j] - P[Y \geq j+1] \quad (23)$$

To calculate $P[Y \geq j]$, we follow the same methodology used in Section IV-C. That is, we introduce Z which expresses the position of the micro-frame during which the receiver wakes up. Z is uniform in $\{1, \cdots, mk\}$, then $P[Z = j] = 1/mk$. We have,

$$\begin{aligned} P[Y \geq j] &= \sum_{i=1}^{k} P[Y \geq j | Z = i] P[Z = i] \\ &= \underbrace{\frac{1}{mk}p^{j-1} + \cdots + \frac{1}{mk}p^{j-1}}_{\text{if the receiver wakes up before position } mk-j} \\ &\quad + \underbrace{\frac{1}{mk}0 + \cdots + \frac{1}{mk}0}_{\text{otherwise}} \\ &= \frac{mk-j}{mk}p^{j-1} \end{aligned} \quad (24)$$

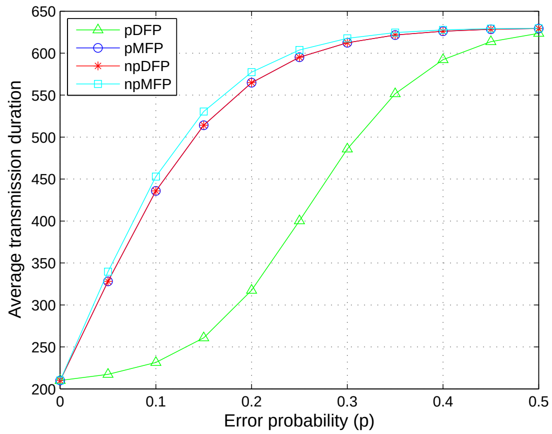


Fig. 4. Average transmission duration

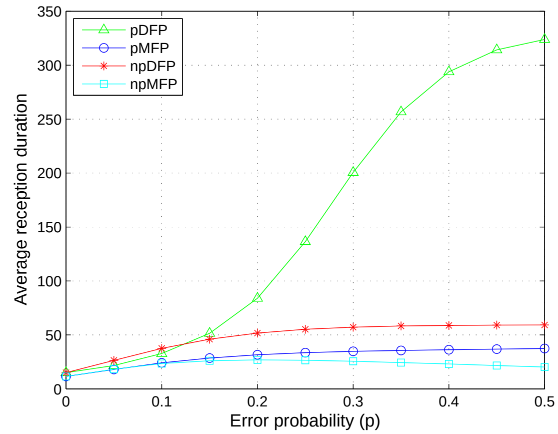


Fig. 6. Average reception duration

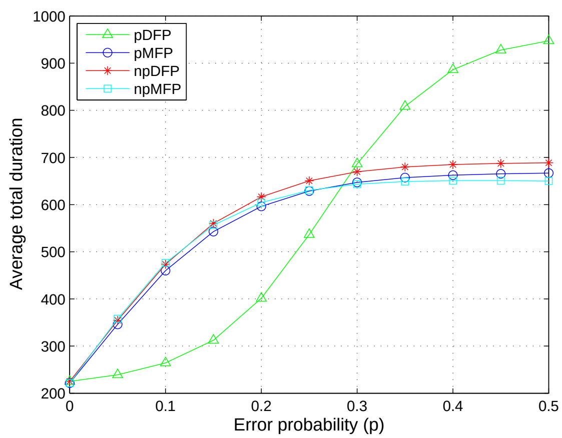


Fig. 5. Average total duration

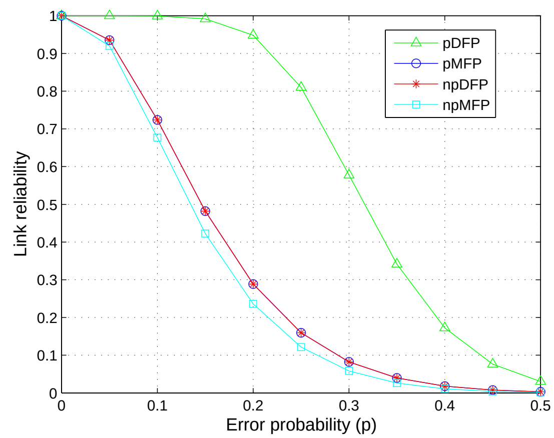


Fig. 7. Link reliability

Therefore,

$$P[Y = j] = \frac{1}{mk}\left[(mk - j)p^{j-1} - (mk - j - 1)p^{j}\right] \quad (25)$$

E. Numerical Examples

In this section, we present numerical results concerning the average transmission and reception durations for the four protocols defined above (Sections II-C and II-D). These metrics are important metrics, beacuse they are allow estimation of the sensor lifetime. In the following, we use these metrics to quantify the link energy cost defined as the average reception duration plus the average transmission duration as shown in Fig. 5. We also plot the corresponding communication reliability.

To plot Figures 4-7, we use Eqs. (1), (3), and (4) with the values of p_f, F, and S derived for each protocol variant. We set the number of retransmissions n to 3, we consider data frames ten times larger than micro-frames, i.e. $m = 10$, we set the check interval to 200, i.e. $k = 20$, and we vary p, the probability that a micro-frame is corrupted from 0 to 0.5. Note also that durations are expressed in time units as described previously (see Section III-A).

Fig. 4 shows that the average transmission durations of p-MFP and np-DFP are equal. This is because these two protocols have the same probability of a successful single transmission, which is equal to the probability of a correct reception of a data frame. Therefore, in both cases the transmitter retransmits the same amount of times. This result is also confirmed in Fig. 7 which shows that p-MFP and np-DFP have the same reliability.

We also notice that the average transmission duration of np-MFP is slightly larger than that of the other protocols. This is due to a larger probability of a single transmission failure than in the other cases, because it does not only depend on a correct reception of a data-frame, but also on a correct reception of a micro-frame in the preamble. Thus, on the average the transmitter retransmits more times than in other variants.

Fig. 4 also shows that p-DFP has the shortest average reception duration. This is because it has the lowest probability that a single transmission fails. Fig. 7 confirms this results as it shows that p-DFP has the highest reliability.

Fig. 6 shows that np-MFP has the lowest average reception duration. For low error probabilities, from $p = 0$ to $p = 0.2$, this duration increases when p increases, because in general, for these error rates the receiver correctly decodes a micro-frame, but fails to decode a data frame. However, for higher error probabilities, from $p = 0.3$ to $p = 0.5$, this duration decreases when p increases, because in this case the receiver mostly fails to decode a correct micro-frame so it does not wake up later on to listen for a data frame.

Fig. 6 also shows that the average reception duration in p-DFP increases when the error rate increases: in this case, the receiver has to listen to more data frames in the preamble to correctly decode one of them. Note that this duration is limited on the average to half of the preamble plus the data frame. This limit is reached when the error rate is extremely high, i.e. p is close to 1. In this case, the average reception duration for p-MFP is also the same as that for p-DFP. The average reception duration in p-MFP increases when p increases, but less sharply than that in p-DFP. This is because, the receiver in p-MFP has a larger probability to decode a micro-frame in the preamble, so it can switch its radio off in the meantime to wake up only to catch the data frame.

V. Conclusion

Preamble MAC Protocols contribute to increasing the lifetime of sensor networks by providing an efficient low power MAC. In this paper, we have focused on modeling performance of the Frame Preamble MAC, an efficient implementation of Preamble MAC protocols. As representative Frame Preamble MACs, we have defined four protocols: persistent MFP, non persistent MFP, persistent DFP, non persistent DFP. For these protocols, we have investigated the relationship between energy cost and communication reliability over a wireless link by assuming a simple binary symmetric channel error model. We have provided a comprehensive mathematical analysis that derives the cost of transmission, reception, and the corresponding reliability according to transmission error rates.

VI. Acknowledgments

Special thanks to Mischa Dohler for proof-reading the paper.

References

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

[2] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. *Proceedings of the IEEE Infocom*, pages 1567–76, New York, NY, July 2002.

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

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

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

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

[7] I. Demirkol, C. Ersoy, and F. Alagoz. MAC Protocols for Wireless Sensor Networks: a Survey. *IEEE Communications Magazine, to appear*, 2005.

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

[9] 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.

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

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

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

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

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