Technical Report: Energy Evaluation of Preamble Sampling MAC Protocols for Wireless Sensor Networks

Giorgio Corbellini, Cédric Abgrall, Emilio Calvanese Strinati, Andrzej Duda

Technical Report: Energy Evaluation of Preamble Sampling MAC Protocols for Wireless Sensor Networks

Giorgio Corbellini*†, Cédric Abgrall*, Emilio Calvanese Strinati*, and Andrzej Duda† *CEA-LETI, MINATEC, Grenoble, France †Grenoble Institute of Technology, CNRS Grenoble Informatics Laboratory UMR 5217, France
Email: [Giorgio.Corbellini, Cedric.Abgrall, Emilio.Calvanese-Strinati]@cea.fr, Andrzej.Duda@imag.fr

Abstract

The technical report presents a simple probabilistic analysis of the energy consumption in preamble sampling MAC protocols. We validate the analytical results with simulations. We compare the classical MAC protocols (B-MAC and X-MAC) with LA-MAC, a method proposed in a companion paper. Our analysis highlights the energy savings achievable with LA-MAC with respect to B-MAC and X-MAC. It also shows that LA-MAC provides the best performance in the considered case of high density networks under traffic congestion.

I. Introduction

Wireless Sensor Networks (WSN) have recently evolved to support diverse applications in various and ubiquitous scenarios, especially in the context of Machine-to-Machine (M2M) networks [1]. Energy consumption is still the main design goal along with providing sufficient performance support for target applications. Medium Access Control (MAC) methods play the key role in saving energy [2] because of the part taken by the radio in the overall energy budget. Thus, the main goal in designing an access method consists of reducing the effects of both *idle listening* during which a device consumes energy while waiting for an eventual transmission and *overhearing* when it receives a frame sent to another device [2].

To save energy, devices aim at achieving low duty cycles: they alternate long sleeping periods (radio switched off) and short active ones (radio switched on). As a result, the challenge of MAC design is to synchronize the instants of the receiver wake-up with possible transmissions of some devices so that the network achieves a very low duty cycle. The existing MAC methods basically use two approaches. The first one synchronizes devices on a common sleep/wake-up schedule by exchanging synchronization messages (SMAC [3], TMAC [4]) or defines a synchronized network wide TDMA structure (LMAC [5], D-MAC [6], TRAMA [7]). With the second approach, each device transmits before each data frame a *preamble* long enough to ensure that intended receivers wake up to catch its frame (Aloha with Preamble Sampling [8], Cycled Receiver [9], LPL (Low Power Listening) in B-MAC [10], B-MAC+ [11], CSMA-MPS [12] aka X-MAC [13], BOX-MAC [14], and DA-MAC [15]). Both approaches converge to the same scheme, called *synchronous preamble sampling*, that uses very short preambles and requires tight synchronization between devices (WiseMAC [16], Scheduled Channel Polling (SCP) [17]).

Thanks to its lack of explicit synchronization, the second approach based on *preamble sampling* appears to be more easily applicable and more scalable than the first synchronous approach. Even if methods based on *preamble sampling* are collision prone, they have attracted great research interest, so that during last years many protocols have been published. In a companion paper, we have proposed LA-MAC, a Low-Latency Asynchronous MAC protocol [18] based on preamble sampling and designed for efficient adaptation of device behaviour to varying network conditions.

In this report, we analytically and numerically compare B-MAC [10], X-MAC [13], and LA-MAC in terms of energy consumption. The novelty of our analysis lies in how we relate the energy consumption to traffic load. In prior energy analyses, authors based the energy consumption on the average Traffic Generation Rate (TGR) of devices [17] as well as on the probability of receiving a packet in a given interval [13]. In contrast to these approaches, which only focus on the consumption of a "transmitter-receiver" couple, we rather consider the global energy cost of a group of neighbor contending devices. Our analysis includes the cost of all radio operations involved in the transmission of data messages, namely the cost of transmitting, receiving, idle listening, overhearing and sleeping.

The motivation for our approach comes from the fact that in complex, dense, and multi-hop networks, traffic distribution is not uniformly spread over the network. Thus, the energy consumption depends on traffic pattern, *e.g. convergecast*, *broadcast*, or *multicast*, because instantaneous traffic load may differ over the network. In our approach, we estimate the energy consumption that depends on the instantaneous traffic load in a given localized area. As a result, our analysis estimates the energy consumption independently of the traffic pattern.

II. Background

We propose to evaluate the energy consumption of a group of contending sensor nodes under three different preamble sampling MAC protocols: B-MAC, X-MAC, and LA-MAC. In complex, dense, and multi-hop networks, the instantaneous traffic distribution over the network is not uniformly spread. For example, in the case of networks with the *convergecast* traffic pattern (all messages go to one sink), the traffic load is higher at nodes that are closer to the sink in terms of number of hops. Due to this *funnelling effect* [19], devices close to the sink exhaust their energy much faster than the others.

The evaluation of the energy consumption in wireless sensor networks is difficult and the energy analyses published in the literature often base the energy consumption of a given protocol on the traffic generation rate of the network [17]. In our opinion, this approach does not fully reflect the complexity of the problem, so we propose to analyze the energy consumption with respect to the number of messages that are buffered in a given geographical area. This approach can represent different congestion situations by varying the instantaneous size of the buffer.

In our analysis, we consider a "star" network composed of a single receiving device (*sink*) and a group of N devices that may have data to send. All devices are within 1-hop radio coverage of each other. We assume that all transmitting devices share a global message buffer for which B sets the number of queued messages, B is then related to network congestion. Among all N devices, N_s of them have at least one packet to send; those nodes with the receiver are called *active* devices. Remaining devices have empty buffers and do not participate in the contention, nevertheless, they are prone to the *overhearing effect*. Thus, there are $N_o = N - N_s$ *over-hearers*. According to the global buffer state B, there are several combinations of how to distribute B packets among N sending devices: depending on the number of packets inside the local buffers of active devices, N_s and N_o may vary for each combination. For instance, there can be B active devices with each one packet to send or less than B active devices with some of them having more than one buffered packet.

In the remainder, we explicitly separate the energy cost due to transmission E_t, reception E_r, polling (listening for any radio activity in the channel) E_l, and sleeping E_s. E_o is the overall energy consumption of all overhearers. The overall energy consumption E is the sum of all these energies. The power consumption of respective radio states is P_t, P_r, P_l, and P_s for transmission, reception, channel polling, and sleeping. The power depends on a specific radio device. We distinguish the polling state from the reception state. When a node is performing channel polling, it listens to any channel for activity—to be detected, a radio transmission must start after the beginning of channel polling. Once a radio activity is detected, the device immediately switches its radio state from polling to receiving. Otherwise, the device that is polling the channel cannot change its radio state. The duration of a message over the air is t_d. The time between two wakeup instants is called a *frame* and lapses $t_f = t_l + t_s$, where t_l and t_s are respectively the channel polling duration and the sleep period. These values are related to the duty cycle.

III. Preamble Sampling MAC Protocols

In this section, we provide the details of the analyzed preamble sampling protocols. Figure 1 presents the operation of all protocols.

A. B-MAC

In B-MAC [10], all nodes periodically repeat the same cycle during their lifetime: wake up, listen to the channel, and then go back to sleep. When an active node wants to transmit a data frame, it first transmits a preamble long enough to cover the entire sleep period of a potential receiver. After the preamble the sender immediately transmits the data frame. When the receiver wakes up and detects the preamble, it switches its radio to the receiving mode and listens to the channel until the complete reception of the data frame. Even if the lack of synchronization results in low overhead, the method presents several drawbacks due to the length of the preamble: high energy consumption of transmitters, high latency, and limited throughput. We denote by t_p^B the duration of the B-MAC preamble.

B. X-MAC

In CSMA-MPS [12] and X-MAC [13], nodes periodically alternate sleep and polling periods. After the end of a polling period, each active node transmits a series of short preambles spaced with gaps. During a gap, the transmitter switches to the idle mode and expects to receive an ACK from the receiver. When a receiver wakes up and receives a preamble, it sends an ACK back to the transmitter to stop the series of preambles, which reduces the energy spent by the transmitter. After the reception of the ACK, the transmitter sends a data frame and goes back to sleep. After data reception, the receiver remains awake for a possible transmission of a single additional data frame. If another active node receives a preamble destined to the same receiver it wishes to send to, it stops transmitting to listen to the channel for an incoming ACK. When it overhears the ACK, it sets a random back-off timer at which it will send its data frame. The transmission of a data frame after the back-off is not preceded by any preamble. Note however that nodes that periodically wake up to sample the channel need to keep listening for a duration that is larger than the gap between short preambles to be able to decide whether there is an ongoing transmission or not. The duration of each short preamble is t_p^X and the ACK duration is t_a^X.

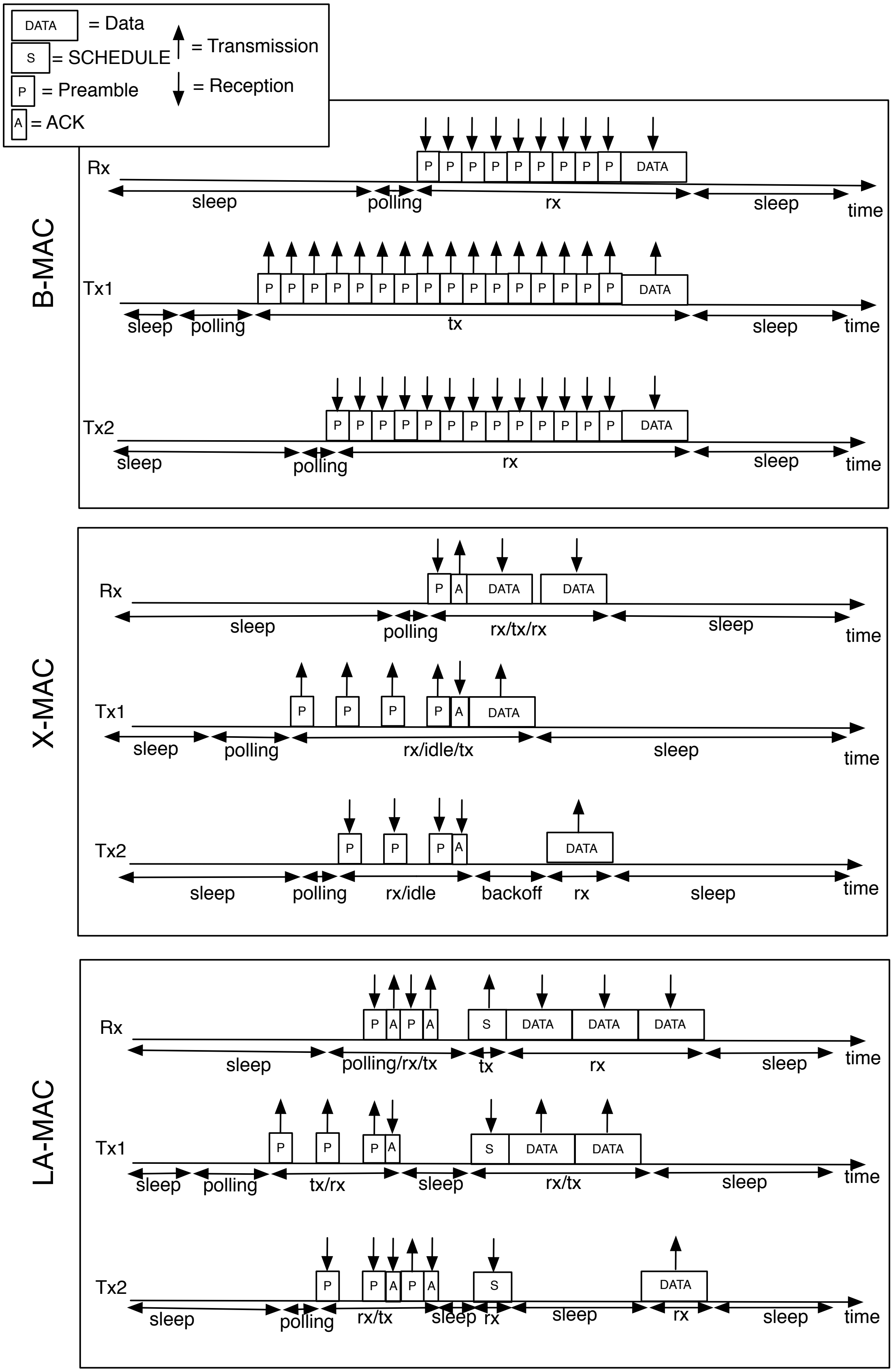

Figure 1: Comparison of analyzed MAC methods.

C. LA-MAC

LA-MAC [18] is a scalable protocol that aims at achieving low latency and limited energy consumption by building on three main ideas: efficient forwarding based on proper scheduling of children nodes that want to transmit, transmissions of frame bursts, and traffic differentiation. It assumes that the network is organized according to some complex structure (tree, DAG, partial mesh) and takes advantage of the network structure to support efficient multi-hop forwarding—a parent of some nodes becomes a coordinator that schedules transmissions in a localized region.

The method periodically adapts local organization of channel access depending on network dynamics such as the number of active users and the instantaneous traffic load. In LA-MAC, nodes periodically alternate long sleep periods and short polling phases. During polling phases each receiver can collect several requests for transmissions included inside short preambles. After the end of its polling period, the node that has collected some preambles processes the requests, compares the priority of requests with the locally backlogged messages and broadcasts a SCHEDULE message. The goal of the SCHEDULE message is to temporarily organize the transmission of neighbor nodes to avoid collisions. If the node that ends its polling has not detected any channel activity and has some backlogged data to send, it starts sending a sequence of short unicast preambles containing the information about the burst to send. As in B-MAC and X-MAC, the strobed sequence is long enough to *wakeup* the receiver. When a receiver wakes up and detects a preamble, it clears it with an ACK frame containing the instant of a *rendezvous* at which it will broadcast the SCHEDULE message. If a second active node overhears a preamble destined to the same destination it wants to send to, it waits for an incoming ACK. After ACK reception, a sender goes to sleep and wakes up at the instant of the *rendezvous*. In Figure 1, we see that after the transmission of an ACK to Tx_1 , Rx device is again ready for receiving preambles from other devices. So, Tx_2 transmits a preamble and receives an ACK during the same *rendezvous*. Preamble clearing continues until the end of the channel polling interval of the receiver.

IV. Energy Analysis

We focus on evaluating energy consumption of a network composed by N transmitters and one sink, the receiver. We provide a separated analytic evaluation of the energy consumption for three preamble sampling protocols: B-MAC, X-MAC, and LA-MAC.

We explicit the analytic expressions of energy consumption *E(B)* starting from the case of empty buffers *B=0* until the generalized expression for unknown values of B.

A. Empty Global Buffer (B=0)

If $B = 0$, all protocols behave in the same way: nodes periodically wake up, poll the channel for t_l seconds, then go back to sleep because of the absence of channel activity and messages to send. Overall network consumption is proportional to network population and only depends on the time that each node spends in polling and sleeping modes:

$$E^{ALL}(0) = (N+1) \cdot (t_l \cdot P_l + t_s \cdot P_s) \tag{1}$$

B. Global Buffer with One Message (B=1)

If there is only one message to send, there are two active devices: the sender, that has a message in the buffer (N_s = 1) and the destination. Other devices ($N_o = N - 1$) have empty buffers, therefore, their energy consumption only depends on channel activity of active nodes that they can overhear and the amount of time that they spend in sleeping mode.

B-MAC ($B = 1$)

When message sender wakes up, it polls the channel for t_l seconds and then starts sending a long preamble that anticipates data transmission. Even if data are assumed unicast, the destination field is not included in preambles; therefore, all neighbor nodes that progressively wake up need to hear both the preamble and the header of the following data to be able to know the identity of the intended destination. The cost for transmission is:

$$E_t^B(1) = (t_p^B + t_d) \cdot P_t \tag{2}$$

Devices are not synchronized and wake-up schedules are uniformly distributed across time, thus, each one hears an average time equal to the half duration of a long preamble before starting data reception. The cost of reception includes the cost of receiving the half duration of a long preamble added to the cost of receiving data. Energy consumption of each node depends upon probability of quasi-synchronization *p*:

$$E_r^B(1) = (p \cdot t_p^B + (1-p) \cdot \frac{t_p^B}{2} + t_d) \cdot P_r \tag{3}$$

The overall polling cost of current case involves both polling procedures of sender and receiver: the first one polls the channel for an entire polling period (t_l seconds) whereas the second one only for a duration that depends on p. The cost of polling activity is:

$$E_l^B(1) = (1 + \frac{p}{2}) \cdot t_l \cdot P_l \tag{4}$$

The cost of sleeping activity concerning the couple transmitter-receiver depends on the time that they do not spend in any mode among polling, receiving, or transmitting:

$$E_s^B(1) = (2 \cdot t_f - (\frac{t_p^B}{2} \cdot (p+3) + 2 \cdot t_d + t_l \cdot (1 + \frac{p}{2}))) \cdot P_s \tag{5}$$

With B-MAC, there is no difference in terms of energy consumption between overhearing and receiving a message. Therefore, the cost of overhearing activity is as follows:

$$E_o^B(1) = N_o \cdot (E_r^B(1) + p \cdot \frac{t_l}{2} \cdot P_l + (t_f - (p \cdot (\frac{t_l}{2} + t_p^B) + (1-p) \cdot \frac{t_p^B}{2} + t_d)) \cdot P_s) \tag{6}$$

X-MAC ($B = 1$)

When the sender wakes up, it polls the channel for t_l seconds and starts sending a series of unicast preambles separated by a gap for *early* ACK reception. Once the sink has received a short preamble, it clears it with an *early* ACK to stop the transmission of preambles and receive data. At this time the sender can transmit its message. After data reception, R_x remains in polling mode for an extra back-off time t_b that is used to receive other possible messages [13]. All devices that have no messages to send and that overhear channel activity go to sleep.

The expected number of preambles that are needed to *wake up* the receiver is γ^X:

$$\gamma^X = (\frac{t_l - t_a^X - t_p^X}{t_f})^{-1}, \tag{7}$$

where t_a^X is the duration of an *early* ACK message, and t_p^X the duration of a preamble message of the series. We remind that before the receiver wakes up and captures a preamble, there are $(\gamma^X - 1)$ preambles whose transmission energy is wasted. In X-MAC, the total amount of energy that results from the activity of transmitting one message depends on the average number of preambles that must be sent (γ^X) and the cost of *early* ACK reception. Provided that wake-up schedules of nodes are not synchronous, it may happen that when the receiver wakes up, the sender is already performing channel polling (transmitter and receiver are quasi-synchronized with probability p).

In the case of quasi-synchronization, the receiver stays an average duration equal to half of t_l in polling mode and then it is able to clear the very first preamble of the incoming series. With probability p, the cost of transmission only includes the cost of transmitting one preamble and the cost of receiving the *early* ACK that follows.

Otherwise, (with probability 1-p) the receiver wakes up after the end of the polling process of the sender; thus, the receiver compels the sender to waste energy for the transmission of γ^X preambles and the wait for an *early* ACK (while waiting for *early* ACK, a node is in polling mode) before it can hear one preamble. Transmission cost is:

$$\begin{aligned} E_t^X(1) = &\quad (1-p) \cdot \gamma^X \cdot t_p^X \cdot P_t + p \cdot t_p^X \cdot P_t + t_a^X \cdot P_r + t_d \cdot P_t \\ = &\quad ((1-p) \cdot \gamma^X + p) \cdot t_p^X \cdot P_t + t_a^X \cdot P_r + t_d \cdot P_t \end{aligned} \tag{8}$$

The cost of receiving activity does not depend on p and it includes the transmission of one *early* ACK plus the reception of both data and preamble.

$$E_r^X(1) = (t_d + t_p^X) \cdot P_r + t_a^X \cdot P_t \tag{9}$$

With probability $1 - p$ (no synchronization) the receiver wakes up while the sender is already transmitting a preamble (or it is waiting for an *early* ACK). Otherwise, (with probability p) the receiver stays in polling mode for an average duration of t_l.

If the active couple is quasi-synchronized, there is a period of time that both T_x and R_x simultaneously spend polling the channel, then, when the sender starts the transmission of the series of preambles, the receiver switches its radio to receiving mode. Within the whole channel polling cost for the sender, are included both the time spent polling the channel and the time that it waits for *early* ACK without any answer (event that happens $\gamma^X - 1$ times with probability $1 - p$).

$$\begin{aligned} E_l^X(1) = &\quad ((t_l + (1-p) \cdot (\gamma^X - 1) \cdot t_a^X) + ((1-p) \cdot \frac{t_p^X + t_a^X}{2} + p \cdot \frac{t_l}{2}) + t_b) \cdot P_l \\ = &\quad ((1-p) \cdot (\frac{t_p^X + t_a^X}{2} + (\gamma^X - 1) \cdot t_a^X) + (\frac{p}{2} + 1) \cdot t_l + t_b) \cdot P_l \end{aligned} \tag{10}$$

The sleeping activity of the active couple is twice a frame duration less the time that both devices spend in one of the active modes:

$$\begin{aligned} E_s^X(1) = \quad & (2 \cdot t_f - (t_l + ((1-p) \cdot \gamma^X + p) \cdot (t_p^X + t_a^X) + t_d) - \\ & +(p \cdot \tfrac{t_l}{2} + t_p^X + t_a^X + (1-p) \cdot \tfrac{t_p^X + t_a^X}{2} + t_d + t_b)) \cdot P_s \\ = \quad & (2 \cdot t_f - 2 \cdot t_d - p \cdot \tfrac{t_l}{2} - t_p^X - t_a^X - (1-p) \cdot \tfrac{t_p^X + t_a^X}{2} - t_l) \cdot P_s + \\ & -(((1-p) \cdot \gamma^X + p) \cdot (t_p^X + t_a^X) - t_b) \cdot P_s \end{aligned} \tag{11}$$

In the same way as other devices, over-hearers can wake up at a random instant.

However, differently from active devices, as soon as they overhear some activity they immediately go back to sleep. Therefore, their energy consumption depends on the probability that such nodes wake up while the channel is busy or not. The probability that at the wake-up instant the channel is free, depends upon several factors such as polling duration, buffer states, and the number of senders. In Fig. 2, we show a tree containing all possible wake-up schedule combinations that may happen. In the tree, we consider as reference instant, the time at which the transmitter wakes up (root of the tree). With probability p, the transmitter (T_x) and the Receiver (R_x) are quasi-synchronized; not synchronized (with probability $(1-p)$), otherwise. With probability $p \cdot p$ both the receiver and a generic over-hearer are quasi-synchronized with the transmitter, this is the Case 1 in the tree. In the remainder, we explicit the expressions for all possible combinations contained in the tree. Overall energy consumption resulting from the overhearing process is the sum of all combinations weighted by relative probabilities (cf. Eq. 21).

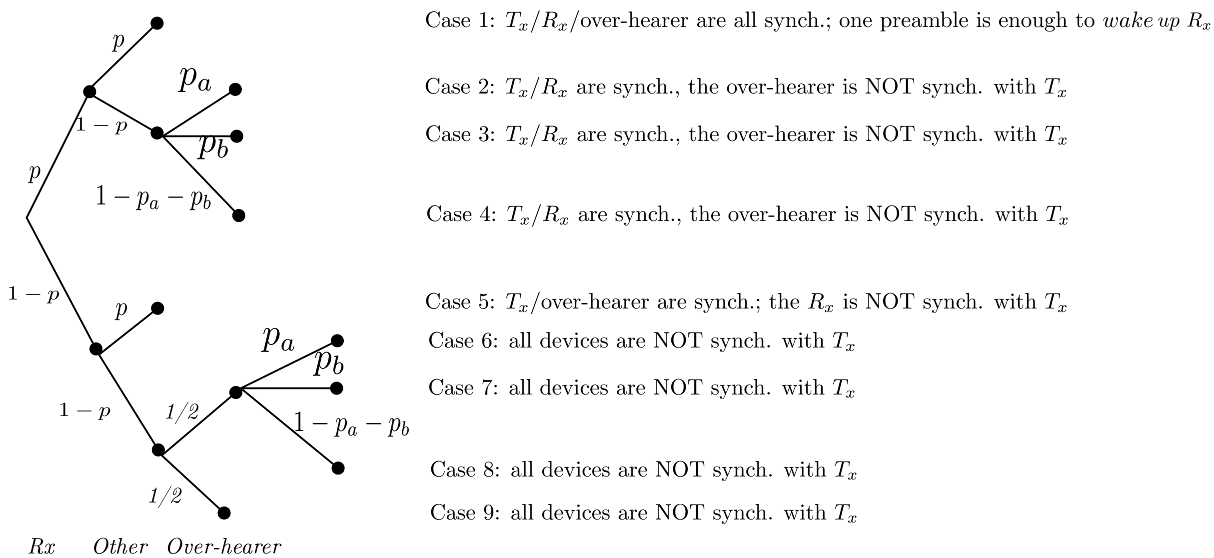

Figure 2: Scenario with global buffer size B=1, X-MAC protocol. Tree containing all possible wake-up schedule combinations of T_x, R_x and over-hearers. Branches are independent, thus, the probability at leaf is the product of probabilities of the whole path from the root to the leaf.

- Case 1: Sender, receiver and over-hearer are all quasi-synchronized (see Fig. 3). The over-hearer receives a preamble for the sink, then it goes back to sleep. Energy consumption of the overhearing action concerning Case 1 is:

$$E_{Case_1,o}^X = \frac{t_l}{2} \cdot P_l + t_p^X \cdot P_r + (t_f - \frac{t_l}{2} - t_p^X) \cdot P_s \tag{12}$$

- Cases 2, 3, and 4: The receiver is synchronized with the sender, whereas the over-hearer is not. When the over-hearer wakes up, it may overhear different messages such as a preamble (Case 2), an *early* ACK (Case 3), a data (Case 4) as well as a clear channel (Case 4 again). Possible situations are summarized in Fig. 4.
  - Case 2: If the over-hearer wakes up during a preamble transmission, it remains in polling mode without sensing any activity until the *early* ACK that follows the preamble is sent; then, the over-hearer goes to sleep. The probability for the over-hearer to wake up during a preamble is $p_a = t_p^X / t_f$. Energy consumption resulting from Case 2 is as follows:

$$E_{Case_2,o}^X = \frac{t_p^X}{2} \cdot P_l + t_a^X \cdot P_r + (t_f - \frac{t_p^X}{2} - t_a^X) \cdot P_s \tag{13}$$

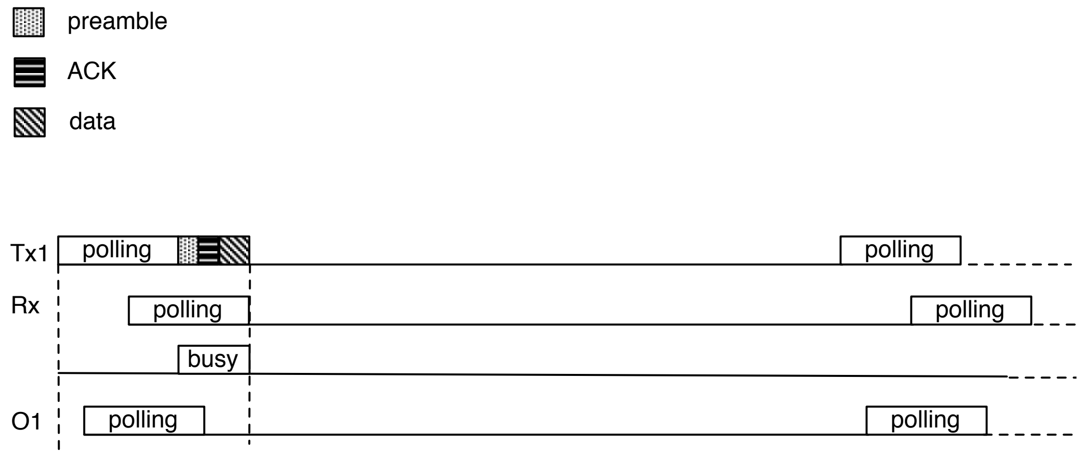

Figure 3: X-MAC protocol, global buffer size $B = 1$. Overhearing situations for Case 1.

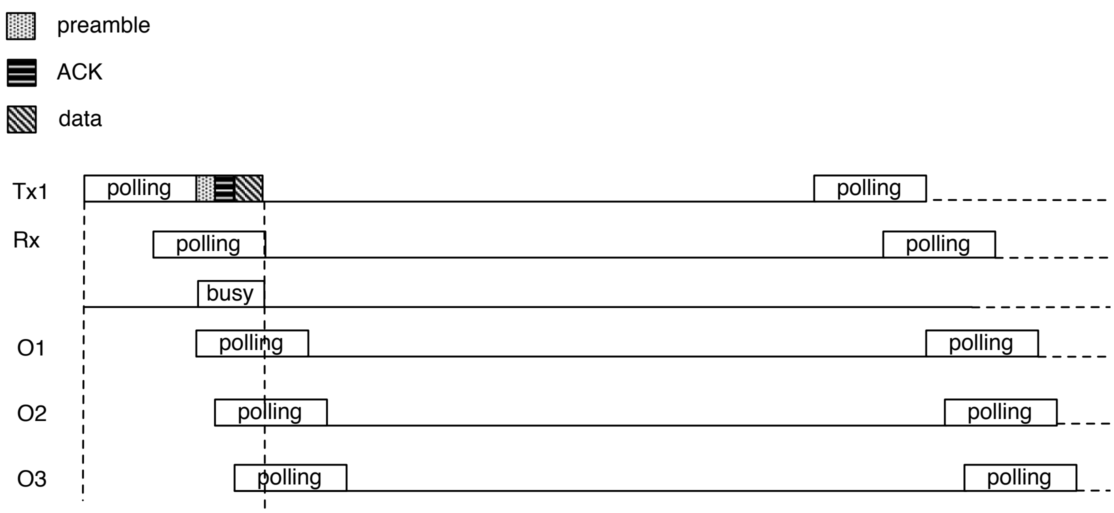

Figure 4: X-MAC protocol, global buffer size $B = 1$. Overhearing situations for Cases 2, 3, and 4.

- Case 3: If an over-hearer wakes up during an *early* ACK transmission, it stays in polling mode without detecting any channel activity until data is overheard; afterwards it goes back to sleep. The probability for the over-hearer to wake up during an *early* ACK is $p_b = t_a^X / t_f$. Energy consumption concerning Case 3 is as follows:

$$E^X_{Case_3,o} = \frac{t_a^X}{2} \cdot P_l + t_d \cdot P_r + (t_f - \frac{t_a^X}{2} - t_d) \cdot P_s \tag{14}$$

- Case 4: The over-hearer either wakes up during data transmission or during the following silent period. In both events when the sender wakes up and senses the channel, it asserts that the channel is clear. From a consumption point of view these events are equivalent because if a message is already being transmitted by T_x when the over-hearer wakes up, it can not capture the begin of the transmission exactly like if there were not an ongoing transmission. Therefore, the over-hearer stays in polling mode for t_l seconds and goes back to sleep immediately after. The probability for this event to happen is $1 - p_a - p_b$. Energy consumption concerning Case 4 is as follows:

$$E^X_{Case_4,o} = t_l \cdot P_l + (t_f - t_l) \cdot P_s \tag{15}$$

- Case 5: Similarly to Case 1, if the over-hearer is quasi-synchronized with the transmitter, it overhears the first preamble even if the receiver is still sleeping; then, it goes back to sleep. The energy cost is as follows:

$$E^X_{Case_5,o} = E^X_{Case_1,o} \tag{16}$$

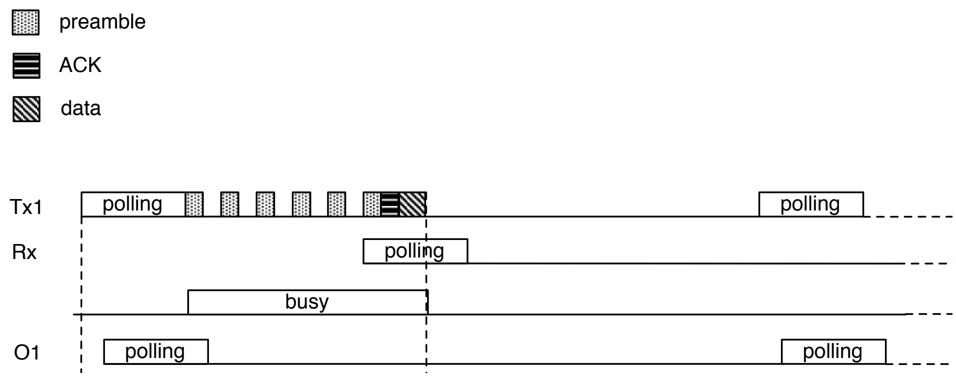

Figure 5: X-MAC protocol, global buffer size $B = 1$. Overhearing situation for Case 5.

- Cases 6, 7 and 8: If neither the receiver nor the over-hearer are synchronized with the sender, it may happen that the receiver wakes up before the over-hearer (cf. Fig. 6). Therefore, similarly to Cases 2, 3 and 4, different situations are possible: Cases 6, 7, and 8 are similar to 2, 3, and 4, respectively. The costs are as follows:

$$E^X_{Case_6,o} = E^X_{Case_2,o} \tag{17}$$

$$E^X_{Case_7,o} = E^X_{Case_3,o} \tag{18}$$

$$E^X_{Case_8,o} = E^X_{Case_4,o} \tag{19}$$

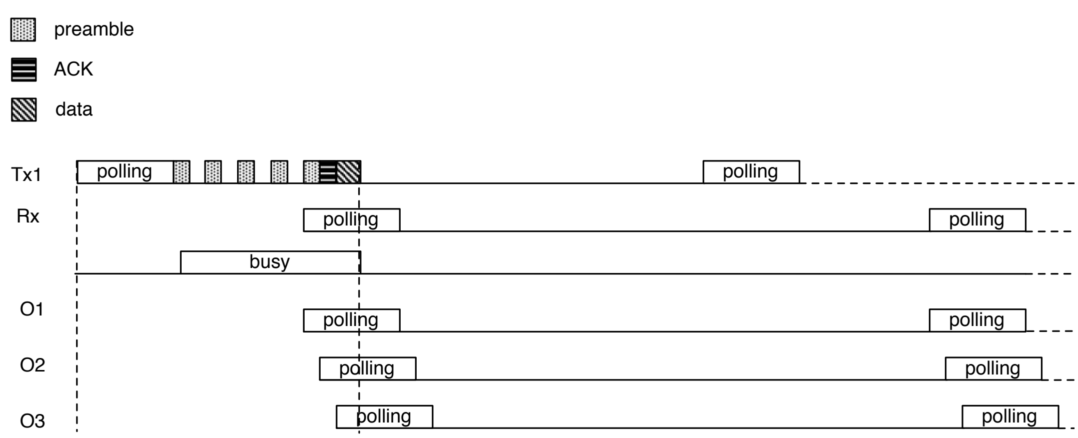

Figure 6: X-MAC protocol, global buffer size $B = 1$. Overhearing situations for Cases 6, 7, and 8.

- Case 9: If the over-hearer wakes up before the R_x, as soon as it receives a preamble, it goes back to sleep. The cost concerning this Case is as follows:

$$E^X_{Case_9,o} = t^X_p \cdot P_r + \frac{t^X_p + t^X_a}{2} \cdot P_l + (t_f - \frac{t^X_p + t^X_a}{2} - t^X_p) \cdot P_s \tag{20}$$

The overall energy cost is the sum of all costs weighted by the probability of the given case to happen:

$$E^X_o(1) = N_o \cdot \sum_{i=1}^{9} p_{Case_i} \cdot E^X_{Case_i,o} \tag{21}$$

LA-MAC ($B = 1$)

In LA-MAC, when the unique sender wakes up, it polls the channel for t_l seconds and then it transmits a series of preambles as in X-MAC. However, differently from X-MAC after *early* ACK reception, the sender goes back to sleep and waits for SCHEDULE message to be sent. Moreover, when the receiver captures one preamble, it clears it with an *early* ACK and completes its polling period in order to detect additional possible preambles to clear. Immediately after the end of polling period, the receiver processes the requests that has cleared and broadcasts a SCHEDULE message containing a local and temporal transmission organization. In LA-MAC, over-hearers go to sleep as soon as they receive any unicast message (preamble, *early* ACK or data) as well as a SCHEDULE (that is a broadcast message).

Because of the lack of synchronization, the expected number of preambles needed to *wake up* the receiver follows X-MAC expression with different sizes of preambles (t_p^L) and *early* ACKs (t_a^L):

$$\gamma^L = \left(\frac{t_l - t_a^L - t_p^L}{t_f}\right)^{-1}. \tag{22}$$

The cost of transmission activity concerning the current case ($E_t^L(1)$) is similar to the cost of X-MAC excepting for the cost of SCHEDULE message that must be added:

$$\begin{aligned} E_t^L(1) = & \quad (1-p)\cdot\gamma^L\cdot t_p^L\cdot P_t + p\cdot t_p^L\cdot P_t + t_a^L\cdot P_r + t_d\cdot P_t + t_g\cdot P_r \\ = & \quad ((1-p)\cdot\gamma^L + p)\cdot t_p^L\cdot P_t + (t_a^L + t_g)\cdot P_r + t_d\cdot P_t \end{aligned} \tag{23}$$

The cost of reception activity depends on the duration of preamble, *early* ACK, data and SCHEDULE messages:

$$E_r^L(1) = (t_p^L + t_d)\cdot P_r + (t_a^L + t_g)\cdot P_t \tag{24}$$

Differently from X-MAC, the receiver completes its polling period independently of the number of cleared preambles, so, its radio remains in polling mode for the duration of t_l seconds less the time spent in receiving a preamble and transmitting *early* ACK. The cost of polling activity is as follows:

$$E_l^L(1) = ((t_l + (1-p)\cdot(\gamma^L - 1)\cdot t_a^L) + (t_l - t_p^L - t_a^L))\cdot P_l \tag{25}$$

When the active nodes are not transmitting, receiving or polling the channel they can sleep. Cost of sleeping activity is as follows:

$$E_s^L(1) = (2\cdot t_f - (t_l + (1-p)\cdot\gamma^L\cdot t_p^L + p\cdot t_p^L + t_a^L + (1-p)\cdot(\gamma^L - 1)\cdot t_a^L + t_d + t_g) - (t_l + t_d + t_g))\cdot P_s \tag{26}$$

As in X-MAC as soon as over-hearers receive a message they go back to sleep. Therefore, their energy consumption depends on the probability that such nodes wake up while the channel is busy or clear. All possible combinations of wake-up schedules with relative probabilities are shown in the tree depicted in Fig. 7. Overall energy cost is the sum of all costs weighted by the probability of the given case to happen (cf. Eq. 38).

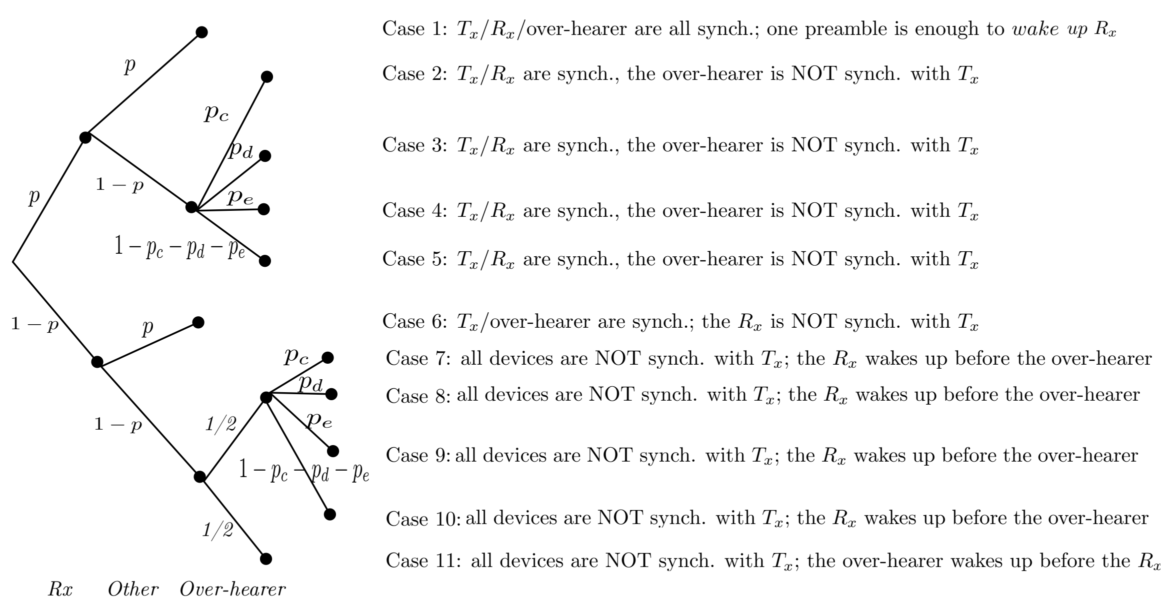

Figure 7: Scenario with global buffer size B=1, LA-MAC protocol. Tree containing all possible wake-up schedule combinations of T_x, R_x and over-hearers. Branches are independent, thus, the probability at leaf is the product of probabilities of the whole path from the root to the leaf.

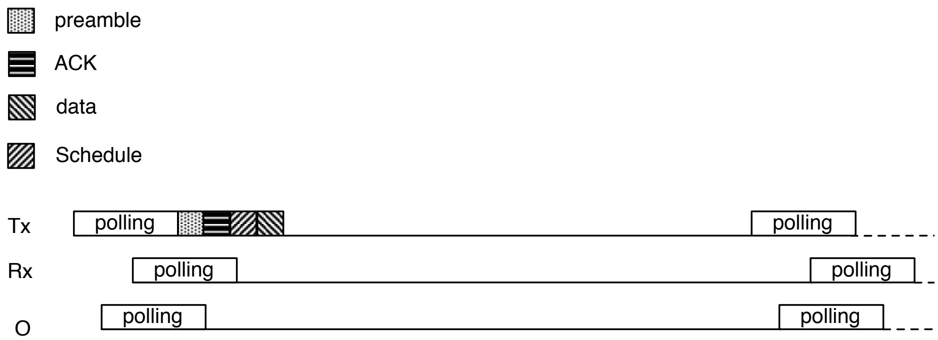

Figure 8: LA-MAC protocol, global buffer size $B = 1$. Overhearing situation for Case 1.

- Case 1: Sender, receiver, and over-hearer are quasi-synchronized. The over-hearer captures a preamble for the sink and goes back to sleep (cf. Sec. 8). The probability of this event to occur is $p \cdot p$. Energy cost concerning Case 1 is as follows:

$$E^L_{Case_1,o} = \frac{t_l}{2} \cdot P_l + t^L_p \cdot P_r + (t_f - \frac{t_l}{2} - t^L_p) \cdot P_s \tag{27}$$

- Cases 2, 3, 4 and 5: The receiver is synchronized with the sender whereas the over-hearer is not. Thus, when the over-hearer wakes up it may receive different messages: a preamble (Case 2), an *early* ACK (Case 3), a SCHEDULE (Case 4) or data (Case 5) as well as clear channel (Case 5 again) (see Fig. 9).

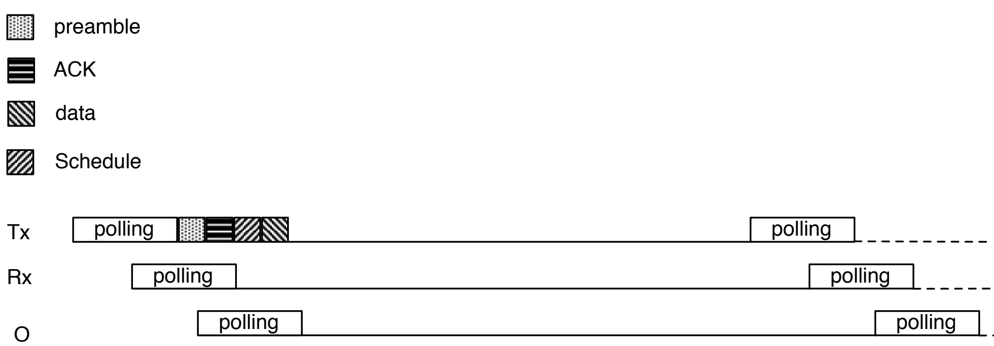

Figure 9: LA-MAC protocol, global buffer size $B = 1$. Overhearing situations for Cases 2, 3, 4 and 5.

  - Case 2: If an over-hearer wakes up during a preamble transmission, it stays in polling mode without receiving any message until it overhears the following *early* ACK. Afterwards it goes back to sleep. Probability of this event to occur is $p \cdot (1-p) \cdot p_c$, where $p_c = t^L_p / t_f$ represents the event that wake-up instant of the over-hearer happens immediately after the end of polling process of the sender. The energy cost concerning Case 2 is as follows:

$$E^L_{Case_2,o} = \frac{t^L_p}{2} \cdot P_l + t^L_a \cdot P_r + (t_f - \frac{t^L_p}{2} - t^L_a) \cdot P_s \tag{28}$$

  - Case 3: If the over-hearer wakes up during an *early* ACK transmission, it senses a silent period and overhears the following SCHEDULE message. Afterwards, it goes back to sleep. The probability of this event to occur is $p \cdot (1-p) \cdot p_d$, where $p_d = t^L_a / t_f$ includes the event that wake-up instant of the over-hearer happens after the transmission of a preamble. p_d neglects the time that elapses between the end of the *early* ACK and the end of channel polling process of the receiver. In other words, with p_d we assume that SCHEDULE message is sent immediately after the transmission of *early* ACK. The energy cost concerning Case 3 is as follows:

$$E^L_{Case_3,o} = \frac{t^L_a}{2} \cdot P_l + t_g \cdot P_r + (t_f - \frac{t^L_a}{2} - t_g) \cdot P_s \tag{29}$$

  - Case 4: If the over-hearer wakes up during the transmission of a SCHEDULE message, it does not sense any channel activity and remains in polling mode until it receives a data, then, it goes to sleep. The probability of this event to occur is $p \cdot (1-p) \cdot p_e$, with $p_e = t_g/t_f$ we assume that the wake-up instant of the over-hearer happens in average at the middle of SCHEDULE transmission. The energy cost concerning Case 4 is as follows:

$$E^L_{Case_4,o} = \frac{t_g}{2} \cdot P_l + t_d \cdot P_r + (t_f - \frac{t_g}{2} - t_d) \cdot P_s \tag{30}$$

- Case 5: The over-hearer either wakes up during data transmission or senses a free channel because both sender and receiver are already sleeping. Therefore, the over-hearer polls the channel for t_l seconds and goes back to sleep. The probability of this event to happen is $p \cdot (1-p) \cdot (1 - p_c - p_d - p_e)$. The energy cost concerning Case 5 is the following:

$$E^L_{Case_5,o} = t_l \cdot P_l + (t_f - t_l) \cdot P_s \tag{31}$$

- Case 6: Similarly to Case 1, if the over-hearer is quasi-synchronized with the sender, with probability $(1-p) \cdot p$ the energy cost is as follows:

$$E^L_{Case_6,o} = \frac{t_l}{2} \cdot P_l + t^L_p \cdot P_r + (t_f - \frac{t_l}{2} - t^L_p) \cdot P_s \tag{32}$$

- Cases 7, 8, 9, and 10: If neither the receiver nor the over-hearer are synchronized with sender, it may happen that the receiver wakes up before the over-hearer. We distinguish the situations of quasi-synchronization of the couple over-hearer-receiver and lack of synchronization as shown in Fig. 10.

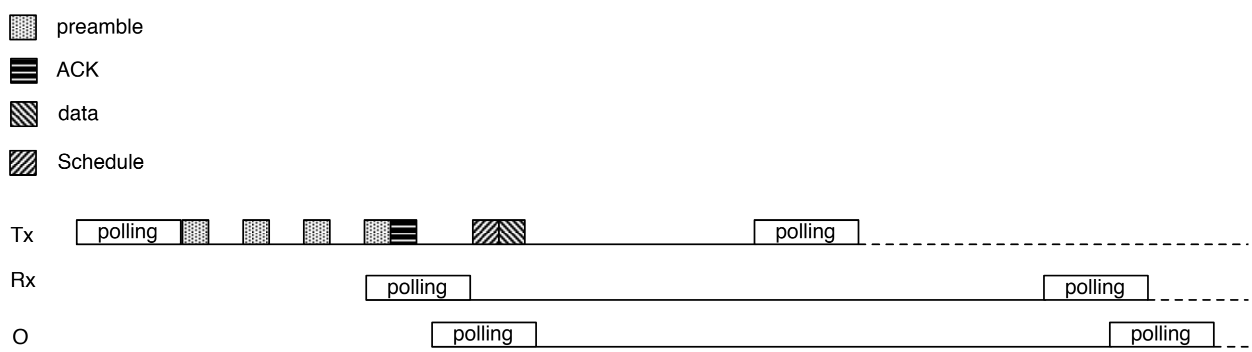

Figure 10: LA-MAC. Possible wake-up instants of over-hearers. Cases 7, 8, 9 and 10.

In Cases 7 and 8, the over-hearer is quasi-synchronized with the receiver:

- Case 7: There is a probability to overhear a preamble. Such a probability is equal to $(1-p) \cdot (1-p) \cdot 1/2 \cdot p_c$. Consumption in this case is the same as in Case 2:

$$E^L_{Case_7,o} = E^L_{Case_2,o} \tag{33}$$

- Case 8: There is a probability to overhear an *early* ACK. Such a probability is equal to $(1-p) \cdot (1-p) \cdot 1/2 \cdot p_d$. Consumption in this case is the same as in Case 3:

$$E^L_{Case_8,o} = E^L_{Case_3,o} \tag{34}$$

If the over-hearer and the receiver are not synchronized among each other:

- Case 9: There is a probability to overhear a SCHEDULE. Such a probability is equal to $(1-p) \cdot (1-p) \cdot 1/2 \cdot p_e$. Consumption in this case is the same as in Case 4:

$$E^L_{Case_9,o} = E^L_{Case_4,o} \tag{35}$$

- Case 10: There is a probability to overhear a data message. Such a probability is equal to $(1-p) \cdot (1-p) \cdot 1/2 \cdot (1 - p_c - p_d - p_e)$. Consumption in this case is the same as in Case 5:

$$E^L_{Case_{10},o} = E^L_{Case_5,o} \tag{36}$$

- Case 11: Otherwise, if the over-hearer wakes up before the destination, it receives one preamble (whichever preamble among γ^L) and goes back to sleep. The cost in this case is as follows:

$$E^L_{Case_{11},o} = \frac{t^L_p + t^L_a}{2} \cdot P_l + t^L_p \cdot P_r + (t_f - \frac{t^L_p + t^L_a}{2} - t^L_p) \cdot P_s \tag{37}$$

The overall energy cost is the sum of all the elementary costs weighted by the probability of the given case to happen:

$$E_o^L(1) = N_o \cdot \sum_{i=1}^{11} p_{Case_i} \cdot E^L_{Case_i,o}. \tag{38}$$

C. Global Buffer with Two Messages (B=2)

If $B = 2$, there can be either one sender with two buffered messages, or two senders (T_{x1} and T_{x2}) with only one buffered message each. The number of over-hearers is $N_o = N - 1$ if there is just one sender, $N_o = N - 2$, otherwise. The probability that two messages are in different buffers is equal to $(N - 1)/N$, where N is the number of nodes excluding the sink.

B-MAC ($B = 2$)

Overall energy consumption for transmission and reception when $B \geq 1$ is linear with the global number of packets in the buffer, independently of how packets are distributed across the different local buffers, *i.e.,* independently of the number of senders. In fact, because of the long preamble to send ($t_p^b = t_f$), there can be only one sender per frame. Thus, the following relation exists: $E^B(B) = B \cdot E^B(1) = B \cdot (E_t^B(1) + E_r^B(1) + E_l^B(1) + E_s^B(1) + E_o^B(1))$.

Such a relation highlights the limitations of B-MAC protocol, since high-loaded traffic can hardly been addressed resulting in both high latency and energy consumption.

X-MAC ($B = 2$)

After the reception of a data message, the receiver remains in polling mode for an extra back-off time t_b during which it can possibly receive a second message. The energy consumed for the transmission of the first packet is the same as the energy defined in the previous section ($E_t^X(1)$); then, an additional cost for the transmission of the second message must be considered.

With probability $1/N$, both packets are in the same buffer; two different senders are involved, otherwise. Differently from B-MAC, the distribution of messages in the buffers impacts protocol behavior.

In case of multiple senders, the overall energy consumption depends on the way how wake-up instants of the active devices are scheduled with respect to each others. For example, assume that device A_1 with a message to send wakes up and receives a preamble from another sender A_2; thus, A_1 must remain in receiving mode and postpones its transmission until the intended receiver clears the preambles of A_2 with an *early* ACK, then A_1 sends its data message during the extra back-off time. In this example, the whole time that A_1 spends overhearing preambles from A_2 is wasted.

All the combinations of wake-up schedules that involve receiver, senders and over-hearers are summarized in the tree depicted in Fig. 11. Wake-up instants of different devices are all independent and we assume that each protocol frame begins at the wake-up instant of T_{x1}.

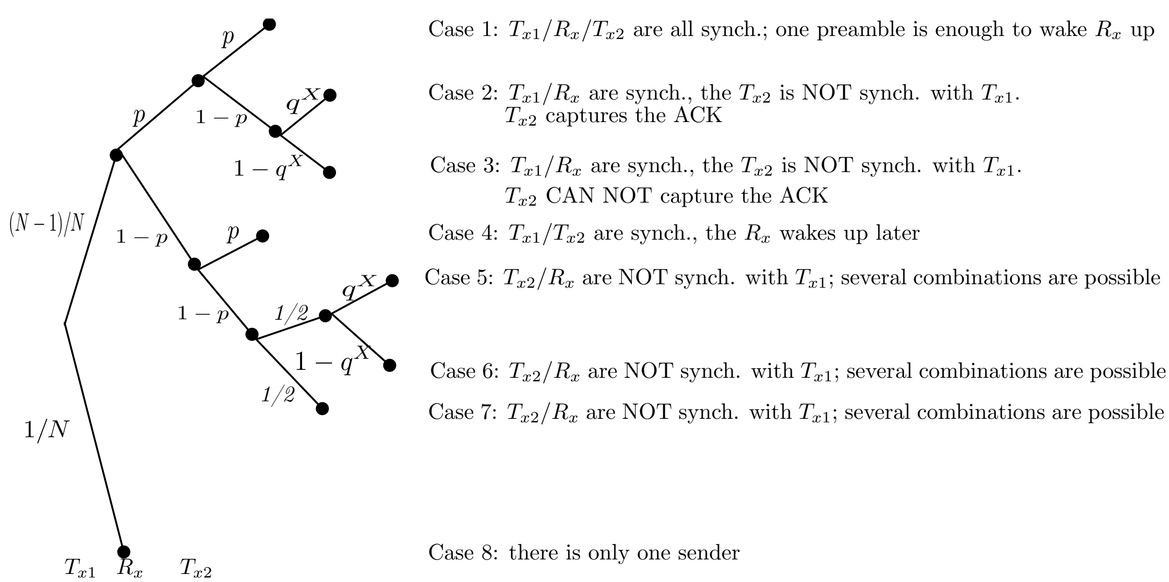

Figure 11: Scenario with global buffer size B=2, X-MAC protocol. Tree containing all possible wake-up schedule combinations of T_{x1}, T_{x2} and R_x. Branches are independent, thus, the probability at leaf is the product of probabilities of the whole path from the root to the leaf.

- Case 1: There are two senders and one receiver, all quasi-synchronized. The very first preamble sent by T_{x1} is cleared by the receiver that sends an *early* ACK; T_{x2} hears both the preamble and the *early* ACK. The probability of this scenario to occur is $p_{Case_1} = (N-1)/N \cdot p \cdot p$. The costs are as follows:

$$E^X_{Case_1,t}(2) = t^X_p \cdot P_t + t^X_a \cdot P_r + (t^X_p + t^X_a) \cdot P_r + 2 \cdot t_d \cdot P_t \tag{39}$$

$$E^X_{Case_1,r}(2) = (t^X_p + 2 \cdot t_d) \cdot P_r + t^X_a \cdot P_t \tag{40}$$

$$E^X_{Case_1,l}(2) = (t_l + \frac{t_l}{2} + \frac{t_l}{2}) \cdot P_l \tag{41}$$

$$E^X_{Case_1,s}(2) = (3 \cdot t_f - (t_l + t^X_p + t^X_a + t_d) - (\frac{t_l}{2} + t^X_p + t^X_a + t_d) - (\frac{t_l}{2} + t^X_p + t^X_a + 2 \cdot t_d)) \cdot P_s \tag{42}$$

Depending on wake-up instants of the over-hearers, several situations may happen. If an over-hearer is quasi-synchronized with one of the three active devices (the receiver or one of the two senders), it senses a busy channel (cf. Fig. 12). In this case, each over-hearer that polls the channel for some time may overhear a preamble, an *early* ACK or a data. For simplicity, we consider the worst case, *i.e.,* we assume that the over-hearer polls the channel for an average duration equal to half of a polling period and then it overhears a data (*i.e.,* the longest message that can be overheard). The probability to wake up during a busy period is $p^X_{case1,B=2} = (t^X_p + t^X_a + 2 \cdot t_d)/t_f$.
Otherwise, if the over-hearer wakes up while channel is free, it polls the channel for t_l seconds and then goes back to sleep. The energy cost is as follows:

$$\begin{aligned} E^X_{Case_1,o}(2) = \; & N_o \cdot (p^X_{case1,B=2} \cdot (\tfrac{t_l}{2} \cdot P_l + t_d \cdot P_r + (t_f - \tfrac{t_l}{2} - t_d) \cdot P_s)) + \\ + \; & N_o \cdot ((1 - p^X_{case1,B=2}) \cdot (t_l \cdot P_l + (t_f - t_l) \cdot P_s)) \end{aligned} \tag{43}$$

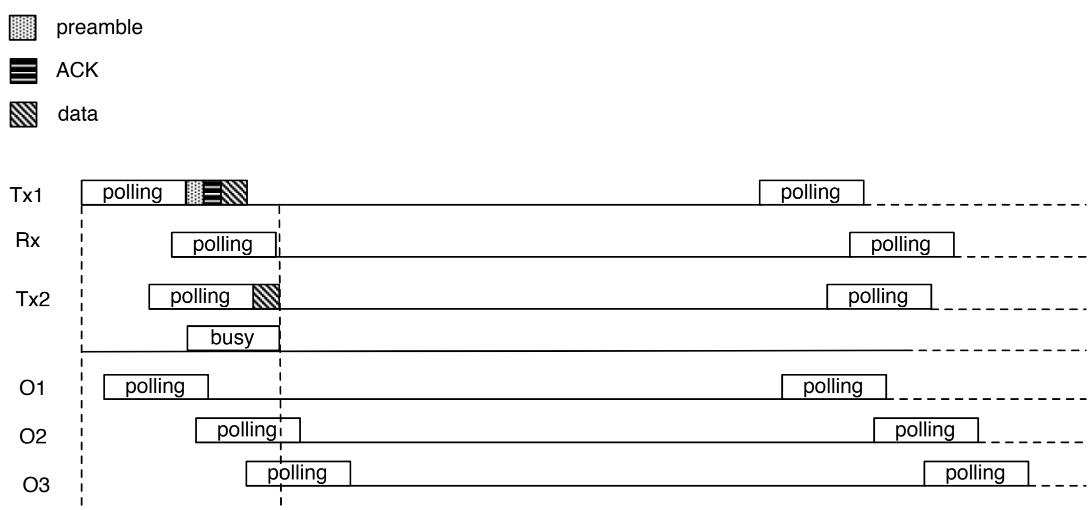

Figure 12: X-MAC protocol, global buffer size $B = 2$. Overhearing situations for Case 1.

- Case 2: The first sender and receiver are quasi-synchronized, whereas T_{x2} is not synchronized with T_{x1} (cf. Fig. 13). The only possibility for the second sender to send data in the current frame is to poll the channel and capture the *early* ACK of the receiver. This event happens with probability $q^X = (t_l - t^X_a)/t_f$. The probability of this scenario is $p_{Case_2} = (N-1)/N \cdot p \cdot (1-p) \cdot q^X$.
Energy consumption concerning Case 2 is about the same as Case 1, with different event probability. Energy consumption of different activities becomes:

$$E^X_{Case_2,t}(2) = E^X_{Case_1,t}(2) - t^X_p \cdot P_r \tag{44}$$

$$E^X_{Case_2,r}(2) = E^X_{Case_1,r}(2) \tag{45}$$

$$E^X_{Case_2,l}(2) = E^X_{Case_1,l}(2) - \frac{t_l - t^X_p}{2} \cdot P_l \tag{46}$$

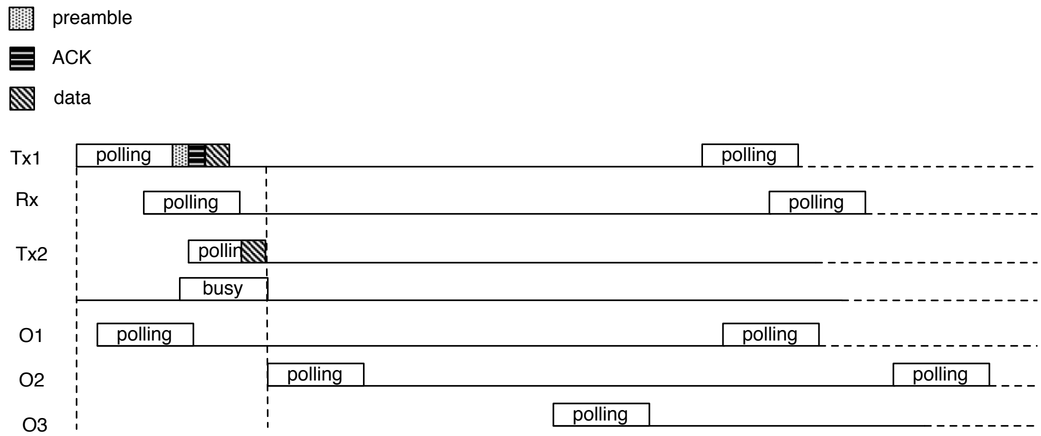

Figure 13: X-MAC protocol, global buffer size $B = 2$. Overhearing situations for Case 2.

$$E^X_{Case_2,s}(2) = E^X_{Case_1,s}(2) + \frac{t_l + t^X_p}{2} \cdot P_l \tag{47}$$

We assume that the probability of busy channel is the same as in Case 1. So, overhearing consumption is unchanged:

$$E^X_{Case_2,o}(2) = E^X_{Case_1,o}(2) \tag{48}$$

- Case 3: With probability $1 - q^X$, the second sender wakes up too late and cannot capture the *early* ACK. If this happens, it goes back to sleep and it transmits its data during the next frame. The energy cost is the sum of the transmission cost of the first packet in the current frame and the second packet in the following frame. The cost of second frame is the same as $E^X(1)$. This scenario happens with probability $p_{Case_3} = (N-1)/N \cdot p \cdot (1-p) \cdot (1-q^X)$. The energy costs in this case are the following:

$$E^X_{Case_3,t}(2) = t^X_p \cdot P_t + t^X_a \cdot P_r + t_d \cdot P_t + E^X_t(1) \tag{49}$$

$$E^X_{Case_3,r}(2) = t^X_p \cdot P_r + t^X_a \cdot P_t + t_d \cdot P_r + E^X_r(1) \tag{50}$$

$$E^X_{Case_3,l}(2) = (t_l + t_l + \frac{t_l}{2}) \cdot P_l + E^X_l(1) \tag{51}$$

$$E^X_{Case_3,s}(2) = (3 \cdot t_f - (t_l + t^X_p + t^X_a + t_d) - t_l - (\frac{t_l}{2} + t^X_p + t^X_a + t_d)) \cdot P_s + E^X_s(1) \tag{52}$$

In the second frame, the local buffer of the first sender is empty, thus, it can be counted as an over-hearer. Therefore, number of over-hearers does not change from first to second frame. The energy cost per over-hearer is the same as in the case of a single message to send $(B = 1)$, that is:

$$E^X_{Case_3,o}(2) = (N_o + (N_o + 1)) \cdot \frac{E^X_o(1)}{N_o + 1} \tag{53}$$

- Case 4: The first and second senders are quasi-synchronized whereas the receiver wakes up later. If this happens, the first sender sends a series of preambles until the receiver wakes up and sends an *early* ACK; second sender hears the entire series of preambles and then sends its data during the extra back-off time (cf. Fig. 14). Between short preambles, both senders poll channel waiting for an *early* ACK from receiver. The probability of this scenario to happen is $p_{Case_4} = (N-1)/N \cdot (1-p) \cdot p$. The energy costs in this case are as follows:

$$E^X_{Case_4,t}(2) = \gamma^X \cdot t^X_p \cdot (P_t + P_r) + 2 \cdot t^X_a \cdot P_r + 2 \cdot t_d \cdot P_t \tag{54}$$

$$E^X_{Case_4,r}(2) = (t^X_p + 2 \cdot t_d) \cdot P_r + t^X_a \cdot P_t \tag{55}$$

$$E^X_{Case_4,l}(2) = (t_l + \frac{t_l}{2} + 2 \cdot (\gamma^X - 1) \cdot t^X_a + \frac{t^X_p + t^X_a}{2}) \cdot P_l \tag{56}$$

$$\begin{aligned} E^X_{Case_4,s}(2) = \quad & (3 \cdot t_f - (t_l + \gamma^X \cdot (t^X_p + t^X_a) + t_d) - (\tfrac{t_l}{2} + \gamma^X \cdot (t^X_p + t^X_a) + t_d) \\ - \quad & (\tfrac{t^X_p + t^X_a}{2} + t^X_p + t^X_a + 2 \cdot t_d)) \cdot P_s \end{aligned} \tag{57}$$

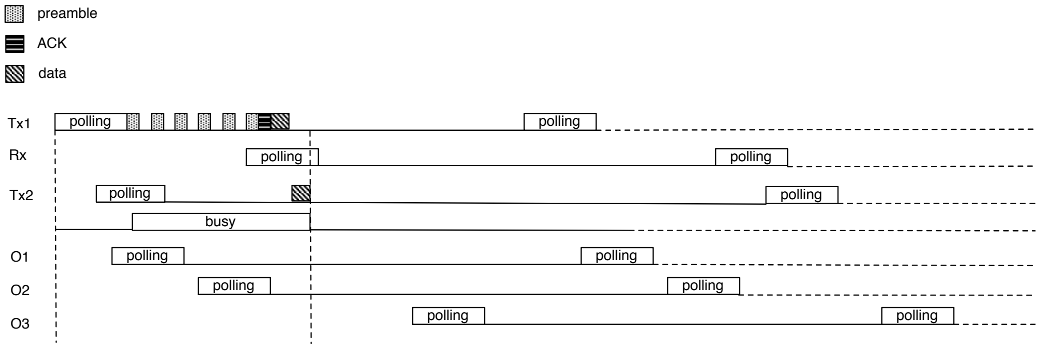

Figure 14: X-MAC protocol, global buffer size $B = 2$. Overhearing situations for Case 4.

Because the receiver wakes up after both senders, the probability that an over-hearer wakes up during a transmission of a preamble is higher than in previous cases. If this happens, the over-hearer stays in polling mode for a very short time, it overhears a message (most likely a preamble) and then it goes back to sleep. For simplicity we assume that the over-hearer polls the channel for a duration equal to half of $(t_p^X + t_a^X)$ and then overhears an entire preamble. The probability of busy channel is thus $p^X_{case4,B=2} = (\gamma^X \cdot (t_p^X + t_a^X) + 2 \cdot t_d)/t_f$. The overhearing cost in this case is the following:

$$\begin{aligned} E^X_{Case_4,o}(2) = &\quad N_o \cdot (p^X_{case4,B=2} \cdot (\tfrac{t_p^X + t_a^X}{2} \cdot P_l + t_p^X \cdot P_r + (t_f - \tfrac{t_p^X + t_a^X}{2} - t_p^X) \cdot P_s) \\ + &\quad (1 - p^X_{case4,B=2}) \cdot (t_l \cdot P_l + (t_f - t_l) \cdot P_s)) \end{aligned} \tag{58}$$

- Cases 5, 6, and 7: The second sender and receiver are not synchronized with first sender; behavior of the protocol depends on which device among T_{x2} and R_x wakes up first.
  - Case 5: The receiver wakes up first as illustrated in Fig. 15. Similarly to Case 2, the only possibility for the second transmitter to send data in the current frame is to poll the channel and capture the *early* ACK of the receiver. This event happens with probability $q^X = (t_l - t_a^X)/t_f$. However, there is also the possibility for T_{x2} to capture a preamble sent by T_{x1}. Such eventuality can happen with probability $u^X = (t_p^X + t_a^X)/(2 \cdot t_p^X + t_a^X)$. This scenario happens with probability $p_{Case_5} = (N-1)/N \cdot (1-p) \cdot (1-p) \cdot \frac{1}{2} \cdot q^X$. The energy costs become:

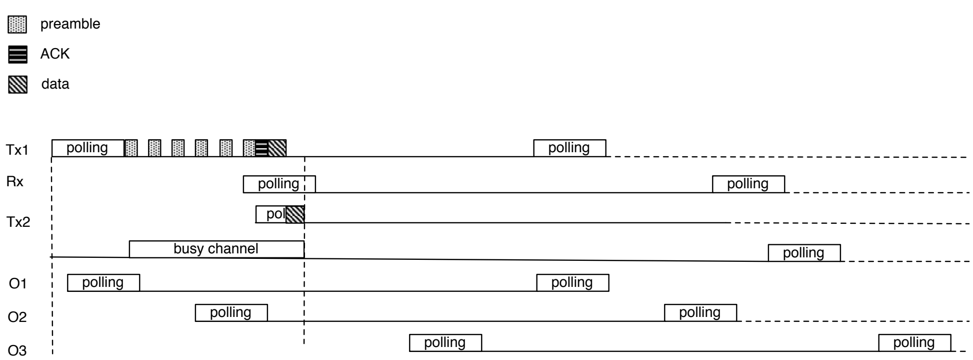

Figure 15: X-MAC protocol, global buffer size $B = 2$. Overhearing situations for Case 5.

$$E^X_{Case_5,t}(2) = (\gamma^X \cdot t_p^X + t_d) \cdot P_t + t_a^X \cdot P_r + (u^X \cdot t_p^X + t_a^X) \cdot P_r + t_d \cdot P_t \tag{59}$$

$$E^X_{Case_5,r}(2) = (t_p^X + 2 \cdot t_d) \cdot P_r + t_a^X \cdot P_t \tag{60}$$

$$E^X_{Case_5,l}(2) = (t_l + (\gamma^X - 1) \cdot t_a^X + \frac{t_p^X + t_a^X}{2} + u^X \cdot \frac{t_p^X + t_a^X}{2} + (1 - u^X) \cdot \frac{t_p^X}{2}) \cdot P_l \tag{61}$$

$$\begin{aligned} E^X_{Case_5,s}(2) = &\quad (3 \cdot t_f - (t_l + \gamma^X \cdot (t_p^X + t_a^X) + t_d) \\ - &\quad (u^X \cdot \tfrac{t_p^X + t_a^X}{2} + (1 - u^X) \cdot \tfrac{t_p^X}{2} + u^X \cdot t_p^X + t_a^X + t_d) \\ - &\quad (\tfrac{t_p^X + t_a^X}{2} + t_p^X + t_a^X + 2 \cdot t_d)) \cdot P_s \end{aligned} \tag{62}$$

As in Case 4, the over-hearer senses a busy channel because of the transmission of preambles; so when it wakes up we assume that if spends half of $(t_p^X + t_a^X)$ time in polling mode before overhearing an entire preamble. The probability of busy channel when the over-hearer wakes up is $p_{case5}^X = p_{case4}^X$. The overhearing costs are as follows:

$$E_{Case_5,o}^X(2) = E_{Case_4,o}^X(2) \tag{63}$$

– Case 6: The receiver wakes up first. Similarly to Case 3, with probability $(1-q^X)$, T_{x2} wakes up too late and cannot capture the *early* ACK from the receiver. Thus, it goes back to sleep and transmits its data during the next frame. This scenario happens with probability $p_{Case_6} = (N-1)/N \cdot (1-p) \cdot (1-p) \cdot \frac{1}{2} \cdot (1-q^X)$. The energy consumption is as follows:

$$E_{Case_6,t}^X(2) = \gamma^X \cdot t_p^X \cdot P_t + t_a^X \cdot P_r + t_d \cdot P_t + E_t^X(1) \tag{64}$$

$$E_{Case_6,r}^X(2) = (t_p^X + t_d) \cdot P_r + t_a^X \cdot P_t + E_r^X(1) \tag{65}$$

$$E_{Case_6,l}^X(2) = (t_l + (\gamma - 1) \cdot t_a^X) \cdot P_l + t_l \cdot P_l + \frac{t_p^X + t_a^X}{2} \cdot P_l + E_l^X(1) \tag{66}$$

$$E_{Case_3,s}^X(2) = (3 \cdot t_f - (t_l + \gamma^X \cdot (t_p^X + t_a^X) + t_d) + t_l + (\frac{t_p^X + t_a^X}{2} + t_p^X + t_a^X + t_d)) \cdot P_s + E_s^X(1) \tag{67}$$

$$E_{Case_6,o}^X(2) = E_{Case_3,o}^X(2) = 2 \cdot E_o^X(1) \tag{68}$$

– Case 7: The second transmitter wakes up first, it over-hears a part of the series of preambles until the receiver wakes up and sends an *early* ACK.
On the average, when T_{x2} wakes up, it polls the channel for a duration that is equal to the half of the gap between two successive short preambles: $(t_p^X + t_a^X)/2$. After that, it over-hears an average number of $\lfloor \gamma^X/2 \rfloor$ short preambles before the receiver wakes up and stops the series of preambles by sending an *early* ACK. The probability of this scenario is $p_{Case_7} = (N-1)/N \cdot (1-p) \cdot (1-p) \cdot \frac{1}{2}$. The energy costs become:

$$E_{Case_7,t}^X(2) = (\gamma^X \cdot t_p^X + t_d) \cdot P_t + t_a^X \cdot P_r + (\lfloor \frac{\gamma^X}{2} \rfloor \cdot t_p^X + t_a^X) \cdot P_r + t_d \cdot P_t \tag{69}$$

$$E_{Case_7,r}^X(2) = (t_p^X + t_d) \cdot P_r + t_a^X \cdot P_t + t_d \cdot P_r \tag{70}$$

$$E_{Case_7,l}^X(2) = (t_l + (\gamma^X - 1) \cdot t_a^X) \cdot P_l + ((\lfloor \frac{\gamma^X}{2} \rfloor - 1) \cdot t_a^X + \frac{t_p^X + t_a^X}{2}) \cdot P_l + \frac{t_p^X + t_a^X}{2} \cdot P_l \tag{71}$$

$$\begin{aligned} E_{Case_7,s}^X(2) = & \quad (3 \cdot t_f - (t_l + \gamma^X \cdot (t_p^X + t_a^X) + t_d) - (\tfrac{t_p^X + t_a^X}{2} + \lfloor \tfrac{\gamma^X}{2} \rfloor \cdot (t_p^X + t_a^X) + t_d) \\ - & \quad (\tfrac{t_p^X + t_a^X}{2} + t_p^X + t_a^X + 2 \cdot t_d)) \cdot P_s \end{aligned} \tag{72}$$

From the over-hearers point of view, this case is equivalent to Cases 4 and 5. The consumption is:

$$E_{Case_7,o}^X(2) = E_{Case_4,o}^X(2) \tag{73}$$

- Case 8: There is only one sender and it has two messages in its buffer. This last scenario happens with a probability equal to $p_{Case_8} = 1/N$. The costs are:

$$E_{Case_8,t}^X(2) = E_t^X(1) + t_d \cdot P_t \tag{74}$$

$$E_{Case_8,r}^X(2) = E_r^X(1) + t_d \cdot P_r \tag{75}$$

$$E_{Case_8,l}^X(2) = E_l^X(1) - t_d \cdot P_l \tag{76}$$

$$E_{Case_8,s}^X(2) = E_s^X(1) - t_d \cdot P_s \tag{77}$$

When the sender is unique, energy consumption of the over-hearers can be assumed about the same as the one in case of a global buffer with one packet to send ($B = 1$). We have:

$$E_{Case_8,o}^X(2) = E_o^X(1) \tag{78}$$

The overall energy cost is the sum of all costs of each scenario, weighted by the probability of the scenario to happen (as showed in Fig. 11):

$$E^X(2) = \sum_{i=1}^{8} p_{Case_i} \cdot E_{Case_i}^X(2) \tag{79}$$

LA-MAC ($B = 2$)

Energy consumption $E^L(2)$ depends on the number of senders as well as on how wake-up instants occur. All different combinations of wake-up instants with their probabilities are given in the tree illustrated in Fig. 16. With the probability equal to $(N-1)/N$, there are two senders, a single sender otherwise. Cases 1-7 refer to situations in which two senders are involved, whereas Case 8 refers to a scenario with one sender. We introduce now some probabilities that are used in the remainder of this section. As previously defined, let $p = t_l/t_f$ be the probability of quasi-synchronization between two devices. The probability that T_{x2} polls the channel and over-hears the *early* ACK from R_x is $q^L = (t_l - t_a^L)/t_f$. The probability that R_x receives a preamble from T_{x2} before the end of its polling period is $w^L = (t_l - 2 \cdot t_p^L - t_a^L)/t_f$.

If none of the previous situations happen, R_x is not able to send an *early* ACK to T_{x2}. In this case, the address of T_{x2} is not included in the SCHEDULE message and it must wait until next frame to send data.

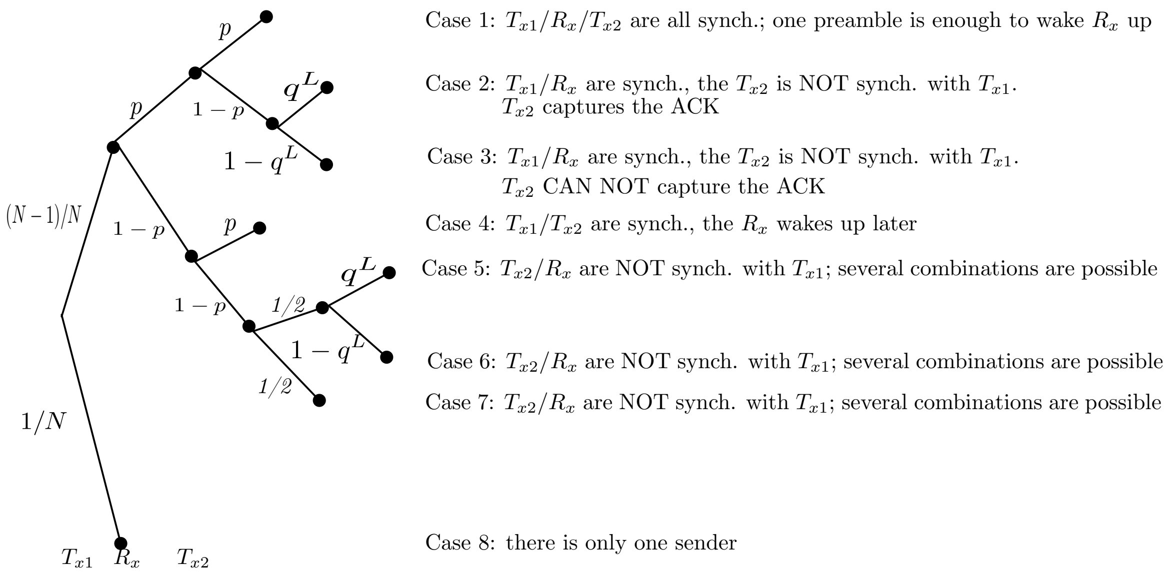

Figure 16: Scenario with global buffer size B=2, LA-MAC protocol. Tree containing all possible wake-up schedule combinations of T_{x1}, T_{x2} and R_x. Branches are independent, thus, the probability at leaf is the product of probabilities of the whole path from the root to the leaf.

- Case 1: The three active devices are all quasi-synchronized. The first preamble is instantly cleared by the receiver; T_{x2} hears both the preamble and the *early* ACK. This event happens with probability $p_{Case_1} = (N-1)/N \cdot p \cdot p$.
  Depending whether the second transmitter succeeds or not in sending in time a preamble (before the end of polling period of the receiver, *i.e.,* with probability w^L), one or two frames are needed for sending two data messages. The energy costs are as follows:

$$\begin{aligned} E^L_{Case_1,t}(2) = \;& (t_p^L + t_d) \cdot P_t + (t_a^L + t_g) \cdot P_r \\ +\;& w^L \cdot (t_p^L \cdot (P_r + P_t) + 2 \cdot t_a^L \cdot P_r + t_g \cdot P_r + t_d \cdot P_t) \\ +\;& (1 - w^L) \cdot (t_p^L \cdot P_r + t_a^L \cdot P_r + E_t^L(1)) \end{aligned} \tag{80}$$

$$E^L_{Case_1,r}(2) = (t_p^L + t_d) \cdot P_r + (t_a^L + t_g) \cdot P_t + w_L \cdot (t_p^L \cdot P_r + t_a^L \cdot P_t + t_d \cdot P_r) + (1 - w^L) \cdot E_r^L(1) \tag{81}$$

$$E^L_{Case_1,l}(2) = (2 \cdot t_l - t_p^L - t_a^l) \cdot P_l + w^L \cdot (-(t_p^L + t_a^L) + \frac{t_l}{2}) \cdot P_l + (1 - w_l) \cdot (\frac{t_l}{2} \cdot P_l + E_l^L(1)) \tag{82}$$

$$\begin{aligned} E^L_{Case_1,s}(2) = \;& (2 \cdot t_f - (t_l + t_p^L + t_a^L + t_g + t_d) - (t_l + t_g + t_d)) \cdot P_s \\ +\;& w^L \cdot (-t_d + t_f - (\tfrac{t_l}{2} + 2 \cdot (t_p^L + t_a^L) + t_g + t_d)) \cdot P_s \\ +\;& (1 - w^L) \cdot ((t_f - (\tfrac{t_l}{2} + t_p^L + t_a^L)) \cdot P_s + E_s^L(1)) \end{aligned} \tag{83}$$

  As far as over-hearers are concerned, several situations may happen depending on their instants of wake-up. If an over-hearer is quasi-synchronized with one of the three active devices (T_{x1}, T_{x2} or R_x), it senses a busy channel (cf. Fig. 17). When an over-hearer wakes up, it polls the channel for some time and then it can overhear a message (that can be a preamble, an *early* ACK, a SCHEDULE or a data). We consider the worst case, *i.e.,* we assume that the over-hearer polls the channel for an average time equal to half the duration of t_l and then it overhears a data (the longest message that can be sent). The probability to wake up during a busy period is $p^L_{case1.1,B=2} = (2 \cdot (t_p^L + t_a^L + t_d) + t_g)/t_f$ if two data are

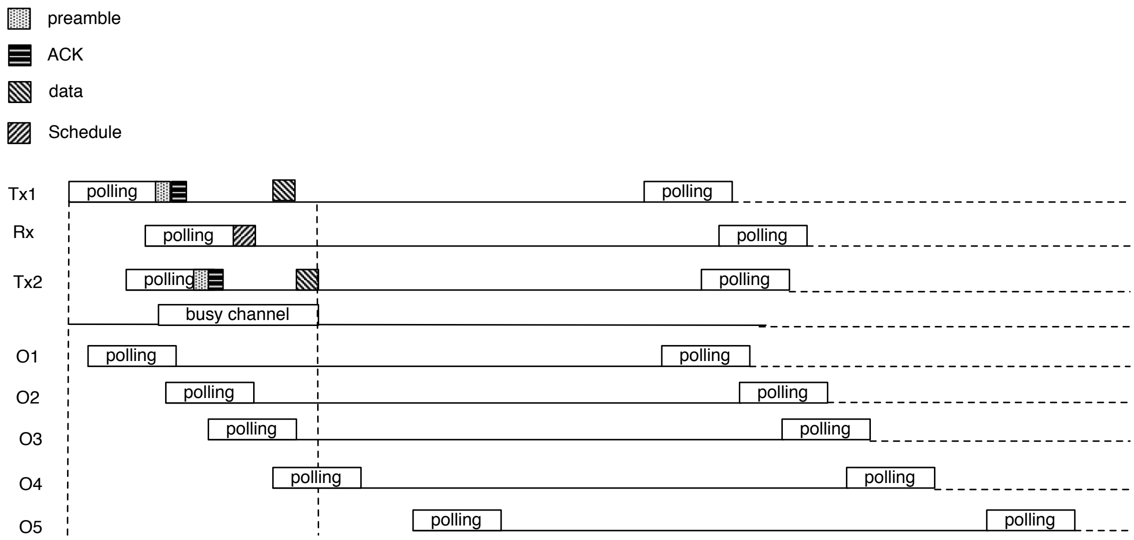

Figure 17: LA-MAC protocol, global buffer size $B = 2$. Overhearing situations for Case 1.

sent within the same frame, $p^L_{case1.2,B=2} = (t^L_p + t^L_a + t_d + t_g)/t_f$, otherwise. If the over-hearer wakes up while channel is free, it polls the channel for t_l seconds, then it goes to sleep. The overhearing cost is the following:

$$\begin{aligned} E^L_{Case_1,o}(2) = \; & N_o \cdot w^L \cdot p^L_{case1.1,B=2} \cdot (\tfrac{t_l}{2} \cdot P_l + t_d \cdot P_r + (t_f - \tfrac{t_l}{2} - t_d) \cdot P_s) \\ + \; & N_o \cdot w^L \cdot (1 - p^L_{case1.1,B=2}) \cdot (t_l \cdot P_l + (t_f - t_l) \cdot P_s) \\ + \; & N_o \cdot (1 - w^L) \cdot p^L_{case1.2,B=2} \cdot (\tfrac{t_l}{2} \cdot P_l + t_d \cdot P_r + (t_f - \tfrac{t_l}{2} - t_d) \cdot P_s) \\ + \; & N_o \cdot (1 - w^L) \cdot (1 - p^L_{case1.2,B=2}) \cdot (t_l \cdot P_l + (t_f - t_l) \cdot P_s) \\ + \; & (1 - w^L) \cdot E^L_o(1) \end{aligned} \tag{84}$$

- Case 2: The first transmitter and the receiver are quasi-synchronized whereas T_{x2} is not synchronized with T_{x1}. R_x first clears a preamble of T_{x1}, and then waits in polling mode for another possible preamble to come until the end of its polling period. Immediately after the end of polling period, the receiver broadcasts a SCHEDULE message. The only possibility for T_{x2} to be included in the SCHEDULE of the current frame is to send a preamble before that the receiver stops polling the channel and that R_x sends it an *early* ACK; this event happens with probability $q^L = (t_l - t^L_a)/t_f$.
If T_{x2} sends a preamble too late, it may happen that there is not enough remaining time for the receiver to receive a preamble and send an *early* ACK (probability of this event is w^L) before the end of its polling period. Case 2 happens with probability $p_{Case_2} = (N-1)/N \cdot p \cdot (1-p) \cdot q^L$. The energy costs in the current case are as follows:

$$\begin{aligned} E^L_{Case_2,t}(2) = \; & (t^L_p + t_d) \cdot P_t + (t^L_a + t_g) \cdot P_r \\ + \; & w^L \cdot ((t^L_p + t_d) \cdot P_t + (2 \cdot t^L_a + t_g) \cdot P_r) \\ + \; & (1 - w^L) \cdot (t^L_a \cdot P_r + E^L_t(1)) \end{aligned} \tag{85}$$

$$E^L_{Case_2,r}(2) = (t^L_p + t_d) \cdot P_r + (t^L_a + t_g) \cdot P_t + w^L \cdot ((t^L_p + t_d) \cdot P_r + t^L_a \cdot P_t) + (1 - w^L) \cdot E^L_r(1) \tag{86}$$

$$E^L_{Case_2,l}(2) = (2 \cdot t_l - t^L_p - t^L_a) \cdot P_l + w^L \cdot (-(t^L_p + t^L_a) + \frac{t^L_p}{2}) \cdot P_l + (1 - w^L) \cdot (\frac{t_l}{2} \cdot P_l + E^L_l(1)) \tag{87}$$

$$\begin{aligned} E^L_{Case_2,s}(2) = \; & (2 \cdot t_f - (t_l + t^L_p + t^L_a + t_g + t_d) - (t_l + t_g + t_d)) \cdot P_s \\ + \; & w_l \cdot (-t_d + t_f - (\tfrac{t^L_p}{2} + t^L_p + 2 \cdot t^L_a + t_g + t_d)) \cdot P_s \\ + \; & (1 - w^L) \cdot ((t_f - (\tfrac{t_l}{2} + t^L_a)) \cdot P_s + E^L_s(1)) \end{aligned} \tag{88}$$

We assume that the probability of busy channel is the same as in Case 1. So, consumption is assumed to be the same, that is:

$$E^L_{Case_2,o}(2) = E^L_{Case_1,o}(2) \tag{89}$$

- Case 3: With probability $(1 - q^L)$, T_{x2} wakes up too late and cannot capture the acknowledge sent by the receiver to T_{x1}. In this case, the second sender goes back to sleep and transmits its data during the next frame. Nevertheless, depending on its exact wake-up instant, T_{x2} can spend more or less time in each radio mode. Let us define the remaining time $t_{remain} = (t_f - t_l/2 - t^L_p - t^L_a)$ as being the part of the receiver frame during which the second sender can wake up. Let us also define a variable that behaves like a test of positivity: $test = max(t_l/2 - t^L_p - t^L_a, 0)$; such $test$ variable states that "T_{x2} wakes up in the time that follows the transmission of *early* ACK by R_x". Case 3 happens with probability $p_{Case_3} = (N-1)/N \cdot p \cdot (1-p) \cdot (1-q^L)$. The energy costs are the following:

$$E^L_{Case_3,t}(2) = (t^L_p + t_d) \cdot P_t + (t^L_a + t_g) \cdot P_r + E^L_t(1) \tag{90}$$

$$E^L_{Case_3,r}(2) = (t^L_p + t_d) \cdot P_r + (t^L_a + t_g) \cdot P_t + E^L_r(1) + \frac{test}{t_{remain}} \cdot t_g \cdot P_r + \frac{t_g}{t_{remain}} \cdot t_d \cdot P_r \tag{91}$$

$$\begin{aligned} E^L_{Case_3,l}(2) = & \quad (2 \cdot t_l - t^L_p - t^L_a) \cdot P_l + E^L_l(1) \\ + & \quad \tfrac{test}{t_{remain}} \cdot \tfrac{test}{2} \cdot P_l + \tfrac{t_g}{t_{remain}} \cdot \tfrac{t_g}{2} \cdot P_l + (1 - \tfrac{test+t_g}{t_{remain}}) \cdot t_l \cdot P_l \end{aligned} \tag{92}$$

$$\begin{aligned} E^L_{Case_3,s}(2) = & \quad (2 \cdot t_f - (t_l + t^L_p + t^L_a + t_g + t_d) - (t_l + t_g + t_d)) \cdot P_s + E^L_s(1) \\ + & \quad \tfrac{test}{t_{remain}} \cdot (t_f - t_g) \cdot P_s + \tfrac{t_g}{t_{remain}} \cdot (t_f - t_d) \cdot P_s + (1 - \tfrac{test+t_g}{t_{remain}}) \cdot (t_f - t_l) \cdot P_s \end{aligned} \tag{93}$$

Since there are two frames for sending two data messages, the energy spent by over-hearers is about the same as the one detailed in previous section ($B = 1$), that is:

$$E^L_{Case_3,o}(2) = \frac{N_o + N_o + 1}{N_o - 1} \cdot E^L_o(1) \tag{94}$$

- Case 4: The first and second senders are quasi-synchronized, whereas the receiver wakes up later (see Fig. 18). In this case, T_{x1} sends a series of preambles until the receiver wakes up and sends the *early* ACK. Even if the second sender overhears some preambles, it must remain awake until *early* ACK is sent. This scenario happens with probability $p_{Case_4} = (N-1)/N \cdot (1-p) \cdot p$ and the resulting energy costs are as follows:

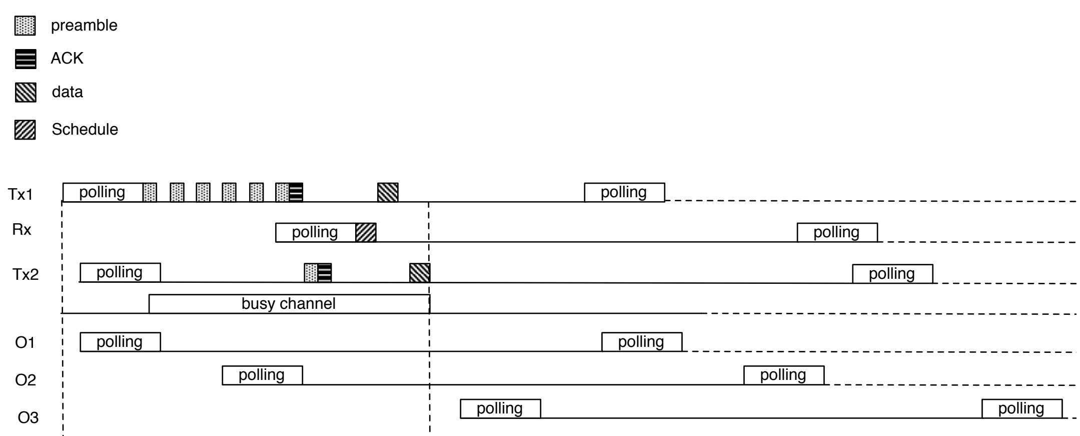

Figure 18: LA-MAC protocol, global buffer size $B = 2$. Overhearing situations for Case 4.

$$E^L_{Case_4,t}(2) = (\gamma^L \cdot t^L_p + t_d) \cdot P_t + (t^L_a + t_g) \cdot P_r + (t^L_p + t_d) \cdot P_t + (\gamma^L \cdot t^L_p + 2 \cdot t^L_a + t_g) \cdot P_r \tag{95}$$

$$E^L_{Case_4,r}(2) = (t^L_p + t_d) \cdot P_r + (t^L_a + t_g) \cdot P_t + (t^L_p + t_d) \cdot P_r + t^L_a \cdot P_t \tag{96}$$

$$E^L_{Case_4,l}(2) = (t_l + (\gamma^L - 1) \cdot t^L_a + t_l - t^L_p - t^L_a) \cdot P_l + (-(t^L_p + t^L_a) + \frac{t_l}{2} + (\gamma^L - 1) \cdot t^L_a) \cdot P_l \tag{97}$$

$$\begin{aligned} E^L_{Case_4,s}(2) = & \quad (2 \cdot t_f - (t_l + \gamma^L \cdot (t^L_p + t^L_a) + t_g + t_d) - (t_l + t_g + t_d)) \cdot P_s \\ + & \quad (-t_d + t_f - \tfrac{t_l}{2} - (\gamma^L + 1) \cdot (t^L_p + t^L_a) - t_g - t_d) \cdot P_s \end{aligned} \tag{98}$$

If the receiver wakes up after the couple of senders, the probability that an over-hearer wakes up during a transmission of a preamble is high. If this happens, the over-hearer stays in polling mode for a very short time, overhears a message (most likely a preamble) and then goes back to sleep. We consider the *pessimistic* case where the over-hearer polls the channel for a duration equal to $\frac{t_l}{2}$ and then overhears the longest possible type of message, *i.e.,* a data. The probability of busy channel when the over-hearer wakes up is $p^L_{case4} = ((\gamma^L + 1) \cdot (t^L_p + t^L_a) + t_g + 2 \cdot t_d)/t_f$. The overhearing cost is as follows:

$$E^L_{Case_4,o}(2) = N_o \cdot (p^L_{case4} \cdot (\frac{t_l}{2} \cdot P_l + t_d \cdot P_r + (t_f - \frac{t_l}{2} - d) \cdot P_s) + (1 - p^L_{case4}) \cdot (t_l \cdot P_l + (t_f - t_l) \cdot P_s)) \tag{99}$$

- Cases 5, 6, and 7: According to these three cases, T_{x2} and R_x are not synchronized with T_{x1}; the behavior of the protocol depends on which device wakes up first among the second transmitter and the receiver.
  - Case 5: R_x wakes up first; similarly to Case 2, the only possibility for the second transmitter to send data in the current frame is to poll the channel and capture the *early* ACK of the receiver (cf. Fig. 19). This event happens with probability $q^L = (t_l - t_a^L)/t_f$. As previously explained, the energy spent for the transmission of the second data message depends on the probability of the receiver to capture in time the preamble sent by the second sender. This fifth scenario has occurring probability given by $p_{Case_5} = (N-1)/N \cdot (1-p) \cdot (1-p) \cdot 1/2 \cdot q^L$. The energy cost concerning this case is as follows:

$$\begin{aligned} E^L_{Case_5,t}(2) = &\ (\gamma^L \cdot t_p^L + t_d) \cdot P_t + (t_a^L + t_g) \cdot P_r \\ + &\ w^L \cdot ((t_p^L + t_d) \cdot P_t + (2 \cdot t_a^L + t_g) \cdot P_r) \\ + &\ (1 - w^L) \cdot (t_a^L \cdot P_r + E_t^L(1)) \end{aligned} \tag{100}$$

$$E^L_{Case_5,r}(2) = (t_p^L + t_d) \cdot P_r + (t_a^L + t_g) \cdot P_t + w^L \cdot ((t_p^L + t_d) \cdot P_r + t_a^L \cdot P_t) + (1 - w^L) \cdot E_r^L(1) \tag{101}$$

$$\begin{aligned} E^L_{Case_5,l}(2) = &\ (t_l + (\gamma^L + 1) \cdot t_a^L + t_l - (t_p^L + t_a^L)) \cdot P_l \\ + &\ w^L \cdot (-(t_p^L + t_a^L) + \tfrac{t_p^L}{2}) \cdot P_l \\ + &\ (1 - w^L) \cdot (\tfrac{t_l}{2} \cdot P_l + E_l^L(1)) \end{aligned} \tag{102}$$

$$\begin{aligned} E^L_{Case_5,s}(2) = &\ (2 \cdot t_f - (t_l + \gamma^L \cdot (t_p^L + t_a^L) + t_g + t_d) - (t_l + t_g + t_d)) \cdot P_s \\ + &\ w_l \cdot (-t_d + t_f - (\tfrac{t_p^L}{2} + t_p^L + 2 \cdot t_a^L + t_g + t_d)) \cdot P_s \\ + &\ (1 - w^L) \cdot ((t_f - (\tfrac{t_l}{2} + t_a^L)) \cdot P_s + E_s^L(1)) \end{aligned} \tag{103}$$

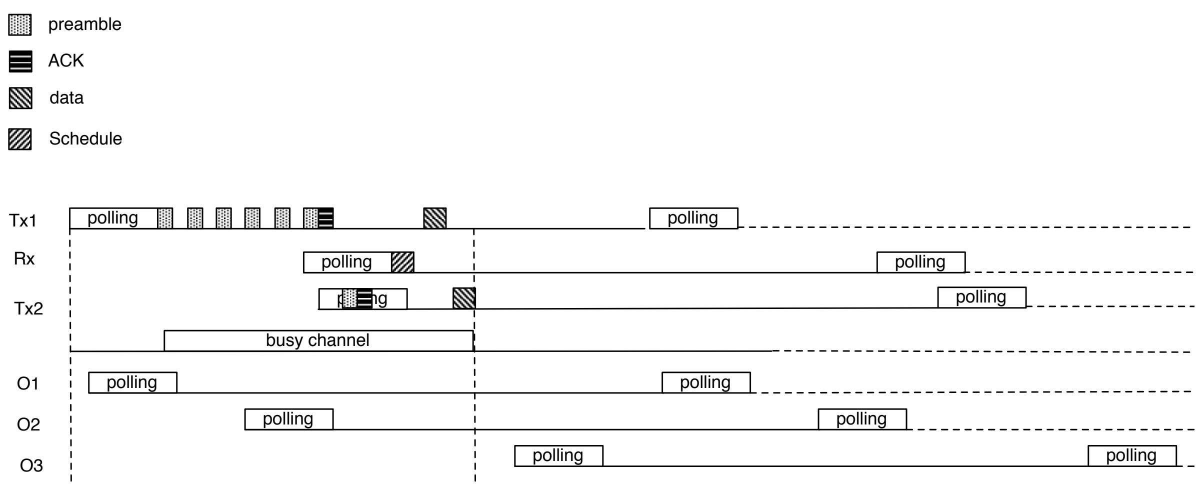

Figure 19: LA-MAC protocol, global buffer size $B = 2$. Overhearing situations for Case 5.

As in the previous scenario, the over-hearer senses a very busy channel because of the transmission of preambles; it wakes up, stays half of t_l in polling mode and than overhears a data. The overhearing cost concerning this case is the following:

$$E^L_{Case_5,o}(2) = E^L_{Case_4,o}(2) \tag{104}$$

  - Case 6: The receiver wakes up before T_{x2}, similarly to Case 3. With probability $(1 - q^L)$, the second sender wakes up too late and cannot capture the *early* acknowledge. In this case, it goes back to sleep and transmits its data during the next frame.
    The first sender needs to send a series of preambles to wake up the receiver. The probability of this scenario to happen is given by $p_{Case_6} = (N-1)/N \cdot (1-p) \cdot (1-p) \cdot 1/2 \cdot (1 - q^L)$.
    We now provide the expressions for t_{remain} and $test$ variables. We have:

$$\begin{aligned} t_{remain} = &\ t_f - \tfrac{t_p^L + t_a^L}{2} - t_p^L - t_a^L \\ test = &\ max(\tfrac{t_p^L + t_a^L}{2} - t_p^L - t_a^L, 0) \end{aligned} \tag{105}$$

    The energy costs of Case 6 are the following:

$$E^L_{Case_6,t}(2) = (\gamma^L \cdot t_p^L + t_d) \cdot P_t + (t_a^L + t_g) \cdot P_r + E_t^L(1) \tag{106}$$

$$E^L_{Case_6,r}(2) = (t^L_p + t_d) \cdot P_r + (t^L_a + t_g) \cdot P_t + E^L_r(1) + \frac{test}{t_{remain}} \cdot t_g \cdot P_r + \frac{t_g}{t_{remain}} \cdot t_d \cdot P_r \tag{107}$$

$$\begin{aligned} E^L_{Case_6,l}(2) = &\quad (t_l + (\gamma^L - 1) \cdot t^L_a + t_l - t^L_p - t^L_a) \cdot P_l + E^L_l(1) \\ + &\quad \tfrac{test}{t_{remain}} \cdot \tfrac{test}{2} \cdot P_l + \tfrac{t_g}{t_{remain}} \cdot \tfrac{t_g}{2} \cdot P_l + (1 - \tfrac{test+t_g}{t_{remain}}) \cdot t_l \cdot P_l \end{aligned} \tag{108}$$

$$\begin{aligned} E^L_{Case_6,s}(2) = &\quad (2 \cdot t_f - (t_l + \gamma^L \cdot (t^L_p + t^L_a) + t_g + t_d) - (t_l + t_g + t_d)) \cdot P_s + E^L_s(1) \\ + &\quad \tfrac{test}{t_{remain}} \cdot (t_f - t_g) \cdot P_s + \tfrac{t_g}{t_{remain}} \cdot (t_f - t_d) \cdot P_s + (1 - \tfrac{test+t_g}{t_{remain}}) \cdot (t_f - t_l) \cdot P_s \end{aligned} \tag{109}$$

From the over-hearers point of view, this scenario is comparable to the one of Case 3, resulting in the following cost equal to:

$$E^L_{Case_6,o}(2) = E^L_{Case_3,o}(2) \tag{110}$$

- Case 7: T_{x2} wakes up before R_x, so, it is ready to send a preamble immediately after the transmission of the *early* ACK destined to T_{x1}. The second transmitter hears a part of the strobed preamble of the first transmitter: in average, it hears $\lfloor \gamma^L/2 \rfloor$ preambles. This scenario has an occurring probability equal to $p_{Case_7} = (N-1)/N \cdot (1-p) \cdot (1-p) \cdot 1/2$. The energy costs are as follows:

$$E^L_{Case_7,t}(2) = (\gamma^L \cdot t^L_p + t_d) \cdot P_t + (t^L_a + t_g) \cdot P_r + (\lfloor \frac{\gamma^L}{2} \rfloor \cdot t^L_p + 2 \cdot t^L_a t_g) \cdot P_r + (t^L_p + t_d) \cdot P_t \tag{111}$$

$$E^L_{Case_7,r}(2) = (t^L_p + t_d) \cdot P_r + (t^L_a + t_g) \cdot P_t + (t^L_p + t_d) \cdot P_r + t^L_a \cdot P_t \tag{112}$$

$$\begin{aligned} E^L_{Case_7,l}(2) = &\quad (t_l + (\gamma^L - 1) \cdot t^L_a + t_l - t^L_p - t^L_a) \cdot P_l \\ + &\quad (-(t^L_p + t^L_a) + \tfrac{t^L_p + t^L_a}{2} + (\lfloor \tfrac{\gamma^L}{2} \rfloor - 1) \cdot t^L_a) \cdot P_l \end{aligned} \tag{113}$$

$$\begin{aligned} E^L_{Case_7,s}(2) = &\quad (2 \cdot t_f - (t_l + \gamma^L \cdot (t^L_p + t^L_a) + t_g + t_d) - (t_l + t_g + t_d)) \cdot P_s \\ + &\quad (-t_d + t_f - \tfrac{t^L_p + t^L_a}{2} - (\lfloor \tfrac{\gamma^L}{2} \rfloor + 1) \cdot (t^L_p + t^L_a) - t_g - t_d) \cdot P_s \end{aligned} \tag{114}$$

From the over-hearers point of view, this case is equivalent to Case 4. The cost is as follows:

$$E^L_{Case_7,o}(2) = E^L_{Case_4,o}(2) \tag{115}$$

- Case 8: There is only one sender that sends two messages in a row. This last scenario happens with probability $p_{Case_8} = 1/N$. The resulting costs are as follows:

$$E^L_{Case_8,t}(2) = E^L_t(1) + t_d \cdot P_t \tag{116}$$

$$E^L_{Case_8,r}(2) = E^L_r(1) + t_d \cdot P_r \tag{117}$$

$$E^L_{Case_8,l}(2) = E^L_l(1) \tag{118}$$

$$E^L_{Case_8,s}(2) = E^L_s(1) - 2 \cdot t_d \cdot P_s \tag{119}$$

When the sender is unique, overhearing consumption can be assumed the same as the case of $B = 1$. The cost in this case becomes:

$$E^L_{Case_8,o}(2) = E^L_o(1) \tag{120}$$

The overall energy cost is the sum of all energy consumption of each case weighted by the probability of the case to happen (as showed on the Fig. 16):

$$E^L(2) = \sum_{i=1}^{8} p_{Case_i} \cdot E^L_{Case_i} \tag{121}$$

D. Global Buffer with More Than Two Messages (B¿2)

We now derive the generic expression of energy consumption for larger values of B. Following the same approach of the previous cases would be complex and tedious because of the large number of possible wake-up combinations, thus, to provide the generalized expression we follow a different approach based on the maximum number of packets that can be sent during a single frame.

B-MAC ($B > 2$)

With B-MAC protocol, if the global buffer state is larger than 1, energy consumption linearly increases with the number of messages in the global buffer independently of how packets are locally distributed, *i.e.,* independently of the number of senders (cf. Sec. IV-C).

X-MAC ($B > 2$)

With X-MAC protocol, only two messages can be delivered per each frame. After the transmission of the first data, other devices with buffered messages to send compete among each other (using the extra back-off time) to directly transmit data (without sending any preamble). Nevertheless, the extra back-off time allows the transmission of only one additional data per frame. If buffer size B is larger than 2, at least two frames are needed to empty it. In the following expressions, we assume that no collisions of preambles and messages occur so that the provided expression is rather *optimistic*. Without collision of preambles, it results that frames are always *efficiently filled*, that is, devices always use the minimal number of frames to send B messages.

The computation of $E^X(B)$ uses a modulo operator: if *B* is even we have to compute the number of *full frames*, *i.e.,* frames during which two messages are sent; otherwise, if *B* is odd, the cost of an extra frame for the remaining data must be added. It follows the expression:

$$\begin{aligned} remain(B) = & \quad rem(B, 2) \\ nb_{full\ frames}(B) = & \quad \frac{B - remain(B)}{2} \\ E^X(B) = & \quad nb_{full\ frames}(B) \cdot E^X(2) + remain(B) \cdot E^X(1) \end{aligned} \tag{122}$$

Consequently, the evolution of $E^X(B)$ with the increasing values of B is a step function that raises each two messages in the buffer, as depicted in Fig. 20.

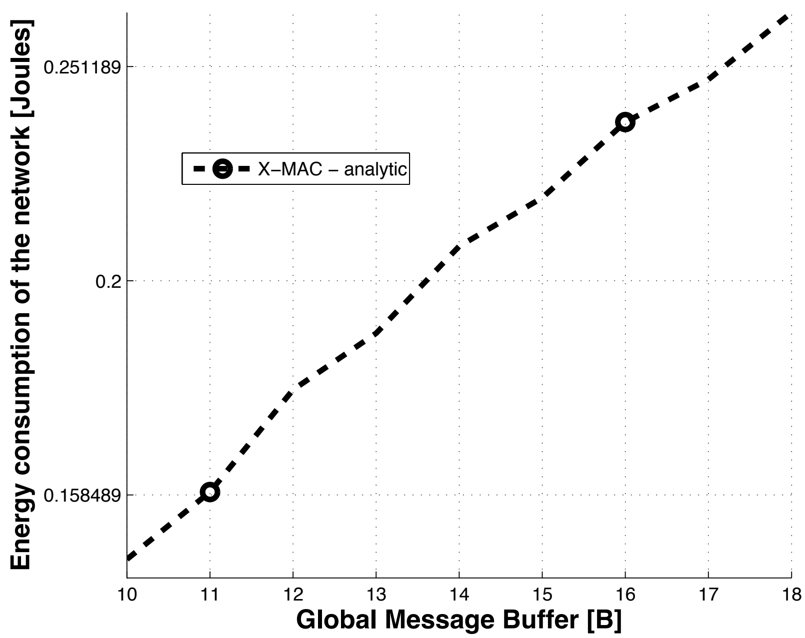

Figure 20: Energy analysis for small values of global buffer size. We focus on the model for X-MAC that shows a step trend each two messages in the buffer.

LA-MAC ($B > 2$)

With LA-MAC protocol, several senders can be scheduled per each frame. As for X-MAC, we assume that there are no collisions and that frames are *efficiently filled*, *i.e.,* devices use the minimal number of frames to send *B* messages.

The limit of data that a frame can contain is fixed by either the duration of a polling period and the duration of a frame, that is the interval between two consecutive wake-ups. Fig. 21 shows the organization of an *efficiently filled* frame: after polling period and SCHEDULE transmission, the whole time until next wake-up instant can be used to transmit messages.

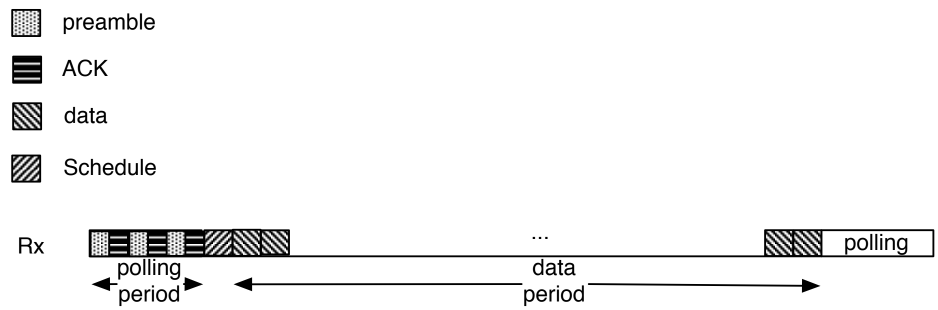

Figure 21: LA-MAC protocol, frame *efficiently filled* with data.

Provided that the polling period has limited duration, the number of preambles that can be cleared during a single polling period is limited as well. For this reason, the way how messages are distributed across nodes influences performance.

Assume that there are 10 messages to send ($B = 10$) and the frame duration is large enough to transmit all messages in a singe frame. If all messages are backlogged in the same buffer, there is only one sender that wakes up the receiver with preambles, receives the SCHEDULE and then transmits all messages to empty its queue. However, if there are 10 senders with one message each, they all try to wake up the receiver; depending on the collisions that may occur and the limited duration of polling periods only a part of them are cleared by the receiver with an *early* ACK. If this happens, only some senders can transmit during the current frame. All senders that do not receive an *early* ACK go to sleep until the next wake up instant.

In the following expressions, we first assume that each transmitter has a maximum one message in its buffer, then we remove this assumption to provide the final expression.

Such an assumption is rather *pessimistic* for two reasons: first, overhead is high because of the cost of sending a series of preamble is high compared to the benefit of sending a single data and second, if there are *B* messages in the global buffer, this implies that there will be *B* contending users that want to send preamble resulting in high traffic congestion.

Analytic expressions that follow assume that energy consumed by all transmitters excepting the first one, is the same. The first sender in fact, is the one who *wakes up* the receiver by sending a series of preambles; thus, it consumes more than other transmitters that overhear preambles and compete for channel access. With the assumption that each transmitter has only one message to send, we can derive the energy cost of the first transmitter (that is the cost for transmitting the first message) from the expression $E^L(2)$ (cf. Eq. 121). In the expression, what is not the consumption of first sender is called *elementary energy consumption* and we assume that this is the cost for each additional transmitter excepting the first one. Let this amount be E^L_{tx2}, we have:

$$E^L_{tx1} = E^L(2) - E^L_{tx2} \tag{123}$$

The overall cost of transmission activity depends on the buffer size and elementary energy consumption. We note that E^L_{tx1} and E^L_{tx2} already include the energy cost for the over-hearers.

We now define the maximum number of preambles that can be acknowledged within a single polling process as $nb^{max}_{preambles}$. The maximum number of data messages that can be transmitted within a frame is : nb^{max}_{data}. Because of the assumption that each node can only transmit one message per frame it holds that the maximum number of data that can be delivered within a single frame is limited by the number of preambles that can be sent in the polling period; such value is $nb^{max}_{data\ per\ frame}$. We have:

$$\begin{aligned} nb^{max}_{preambles} &= \lfloor \tfrac{t_l}{t^L_p + t^L_a} \rfloor \\ nb^{max}_{data} &= \lfloor \tfrac{t_f - t_l - t_g}{t_d} \rfloor \\ nb^{max}_{data\ per\ frame} &= min(nb^{max}_{preambles}, nb^{max}_{data}) \end{aligned} \tag{124}$$

To compute the number of necessary full frames as well as the number of data in the last and incomplete frame, we use a modulo operator:

$$\begin{aligned} remain(B) &= rem(B, nb^{max}_{data\ per\ frame}) \\ nb_{full\ frames}(B) &= \tfrac{B - remain(B)}{nb^{max}_{data\ per\ frame}} \end{aligned} \tag{125}$$

The overall energy cost is composed of the sum of E^L_{tx1}, a fixed part corresponding to the transmission of the first data, and E^L_{tx2}, additional variable part depending on *B* and elementary energy consumption. It holds:

$$E^L_{pessimistic}(B) = nb_{full\ frames}(B) \cdot (E^L_{tx1} + (nb^{max}_{data\ per\ frame} - 1) \cdot E^L_{tx2}) + E_{last\ frame}(B) \tag{126}$$

where B is used to compute $nb_{full\,frames}$ and $remain$; besides, also last incomplete frame must be considered:

$$E_{last\,frame}(B) = \begin{cases} remain(B) \cdot E^L_{tx1} & \text{if } (remain(B) \leq 1) \\ E^L_{tx1} + (remain(B) - 1) \cdot E^L_{tx2} & \text{otherwise} \end{cases} \tag{127}$$

Provided that each transmitter has only one message to send, two situations are possible: either there are few messages in the global buffer so that a portion of polling period of R_x is unused or the number of messages in the global buffer is larger than the maximum number of preambles allowed in a single polling period. We explicit both cases:

- If $(nb^{max}_{data} < nb^{max}_{preambles})$, it means that the receiver spends a part of its polling period without receiving any preamble. For this reason, in this case, we set $nb^{max}_{data\,per\,frame} = nb^{max}_{data}$ and we assume

$$E^L(B) = E^L_{pessimistic}(B) \tag{128}$$

- Otherwise, the receiver spends the entire polling period in receiving preambles and sending *early* ACKs. Thus, $nb^{max}_{preambles}$ senders will send one message each. If there are more than $nb^{max}_{preambles}$ messages in the buffer, the senders will need several frames to deliver all of them, thus jeopardizing LA-MAC performance.

We now release the assumption that each sender has only one message to send to derive *optimistic* energy consumption for LA-MAC.

Since $nb^{max}_{data} \geq nb^{max}_{preambles}$, some transmitters will send more than one data message each. We do not need to know how these data messages are distributed across all the different senders.

As previously mentioned, this energy is formed by the part E^L_{tx1} for the transmission of the first sender and by several times E^L_{tx2}. The total number of messages that are sent in a single frame is nb^{max}_{data}. For each data message, sender and receiver spend t_d seconds respectively in sending and receiving, instead of sleeping. We have:

$$\begin{array}{l} \text{Number of data to send in the last frame:} \\ \quad remain(B) = rem(B, nb^{max}_{data}) \\ \text{Number of complete frames:} \\ \quad nb_{full\,frames}(B) = \frac{B - remain(B)}{nb^{max}_{data}} \end{array} \tag{129}$$

$$\begin{aligned} E^L_{full\,frame} = & \quad E^L_{tx1} + (nb^{max}_{preambles} - 1) \cdot E^L_{tx2} \\ + & \quad (nb^{max}_{data} - nb^{max}_{preambles} + 1) \cdot t_d \cdot (P_t + P_r - 2 \cdot P_s) \end{aligned} \tag{130}$$

If the buffer size is larger than the maximum number of messages that can be sent in a single frame, is needed an additional frame. The last frame may be not completely filled, either because there are not enough senders to fill the entire polling period, or because there are less than nb^{max}_{data} to send. We have:

$$\begin{array}{l} \text{Number of data to send in the last frame:} \\ \quad remain(B) = rem(B, nb^{max}_{data}) \end{array} \tag{131}$$

Thus, energy consumption for the last frame is as follows:

$$\begin{array}{lll} \text{IF} & \multicolumn{2}{l}{(remain(B) \leq nb^{max}_{preambles}) AND (remain(B) \neq 0)} \\ & \text{IF} & (remain(B) = 1) \\ & & E^L_{last\,frame}(B) = E^L(1) \\ & \text{ELSE} & \\ & & E^L_{last\,frame}(B) = E^L_{tx1} + (remain(B) - 1) \cdot E^L_{tx2} \\ \text{ELSE} & & \\ & \multicolumn{2}{l}{E^L_{last\,frame}(B) = E^L_{full\,frame} - (nb^{max}_{data} - remain(B)) \cdot t_d \cdot (P_t + P_r - 2 \cdot P_s)} \end{array} \tag{132}$$

Finally, we can derive the overall energy consumption:

$$E^L_{optimistic}(B) = nb_{full\,frames}(B) \cdot E^L_{full\,frame} + E^L_{last\,frame}(B) \tag{133}$$

Equation 133 is *optimistic* for several reasons. First, all frames are *efficiently filled* (cf. Fig. 21). The equation assumes that the first $nb^{max}_{preambles}$ that are cleared by the receiver contain a global transmission request so that frames are filled. In the real world however, there is a probability that this not happens: nodes that win the contention and transmit data may have transmission requests of few messages so that frames are not *efficiently filled.* Second, preambles may collide so that even though there are more than $nb^{max}_{preambles}$ senders with backlogged messages, the number of preambles that are cleared in a given polling period is smaller than $nb^{max}_{preambles}$. In this case, some senders must go to sleep and wait for the next wake-up instant. Both *pessimistic* and the *optimistic* expressions are plotted in Fig. 22. The curves illustrated in the figure are obtained assuming that $nb^{max}_{preambles}$ is equal to 5 and nb^{max}_{data} to 29. Such values are used in the numerical validation that is presented in the following section (cf. Fig.V). As expected, the *pessimistic* curve shows a step trend each 5 messages, because no more than 5 messages can be sent per each frame. In this case, only 5 messages over a maximum of 29 are sent in each frame.

Also the *optimistic* curve shows a step trend, however, in this case the step size is larger, because the optimistic model assumes that frames are always efficiently filled, *i.e.,* there is an increment of consumed energy each 29 messages in the buffer.

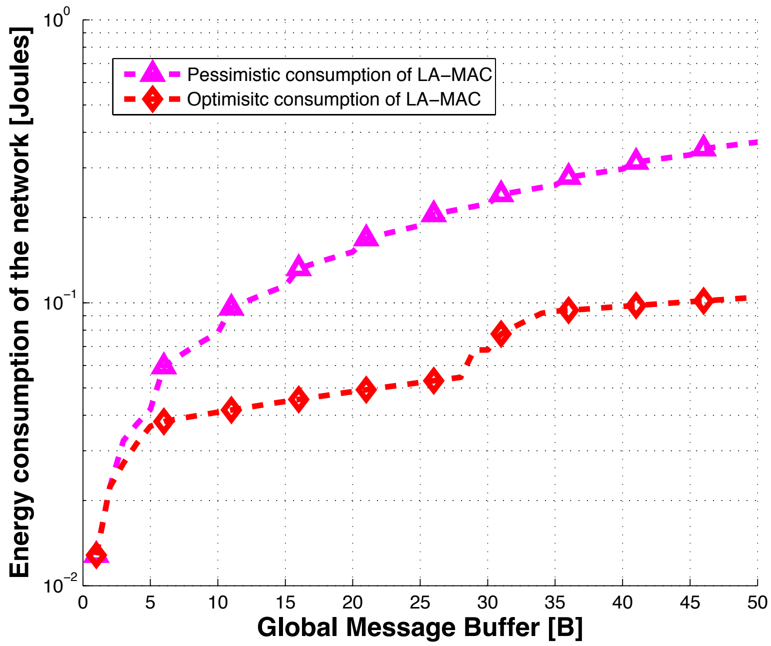

Figure 22: Comparison between optimistic and pessimistic energy consumption of LA-MAC vs. the global buffer size.

V. Numerical Validation

We have implemented the analyzed MAC protocols in the OMNeT++ simulator [20] for numerical evaluation. Each numerical value is the average of 100 runs and we show the corresponding confidence intervals at 95% confidence level. We assume that devices use the CC1100 [21] radio stack with bitrate of 20Kbps. The values of power consumption for different radio states are specific to the CC1100 transceiver considering a 3V battery. Each numerical value is averaged over 1000 independent simulation runs and figures show the corresponding confidence intervals at 95% confidence level. We assume a scenario with $N = 9$ senders and one receiver. Periodical wake-up period is the same for all protocols: $t_f = t_l + t_s = 250\ ms$. Also the polling duration is the same for all protocols: $t_l = 25\ ms$, thus the duty cycle with no messages to send is $10\,\%$. We provide numerical and analytic results for buffer size $B \in [1, 50]$.

In Fig. 23, we show the comparison between the proposed energy consumption analysis and numerical simulations for different values of the global buffer size. We assume that at the beginning of each simulation all messages to send are already buffered, so that the first sender starts its channel polling at $t = 0$ and other devices wake up later as assumed in the analytic analysis. The simulation stop condition is the delivery of last message in the buffer. Fig. 23 highlights the validity of the analytic expressions for energy consumption—we can see that the curves reflect the main trends. The simulation results exceed the analytic data because the simulation reflects the detailed behavior of the protocols, which cannot be captured in simple expressions. As expected, B-MAC is the most energy consuming protocol: as the buffer size increases, the transmission of a long preamble locally saturates the network resulting in high energy consumption and latency (cf. Fig. 25). In X-MAC, short preambles mitigate the effect of the increasing local traffic load, thus both latency and energy consumption are reduced with respect to B-MAC. Even if X-MAC is more energy efficient than B-MAC, Fig. 24 shows that even for small buffer sizes, the delivery ratio for this protocol is lower than 100 % most likely because packets that are sent after the back-off collide at the receiver. Energy consumption of LA-MAC lies in between the pessimistic and the optimistic curves when global buffer size is higher than 16. When traffic load is light, we observe that energy consumption of LA-MAC slightly exceeds the pessimistic curve. The reason for this is that even though the maximum number of preambles that can be cleared in a polling period is 5 (with current protocol parameters), the probability to clear exactly 5 preambles is low when the number of senders is low. In fact, to clear the maximum of preambles it must happen that one the senders transmits a preamble immediately after the beginning of the polling period of the receiver so that the time between the begin of channel polling and the reception of the first preamble is minimized. When traffic load is light, the number of senders is limited and each node has only few messages

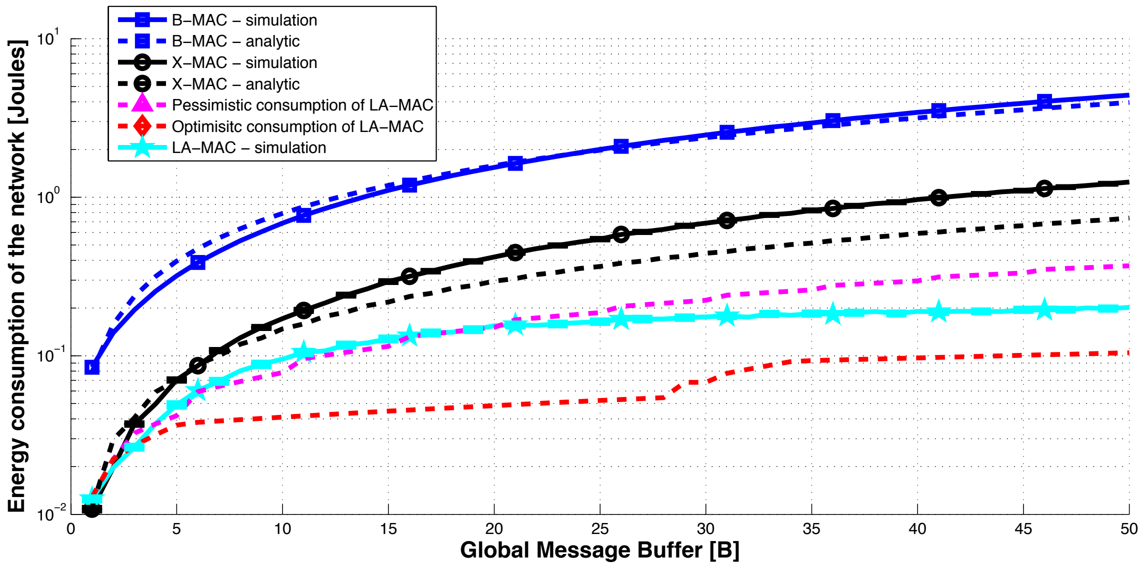

Figure 23: Energy analysis and OMNeT++ simulations versus the global buffer size.

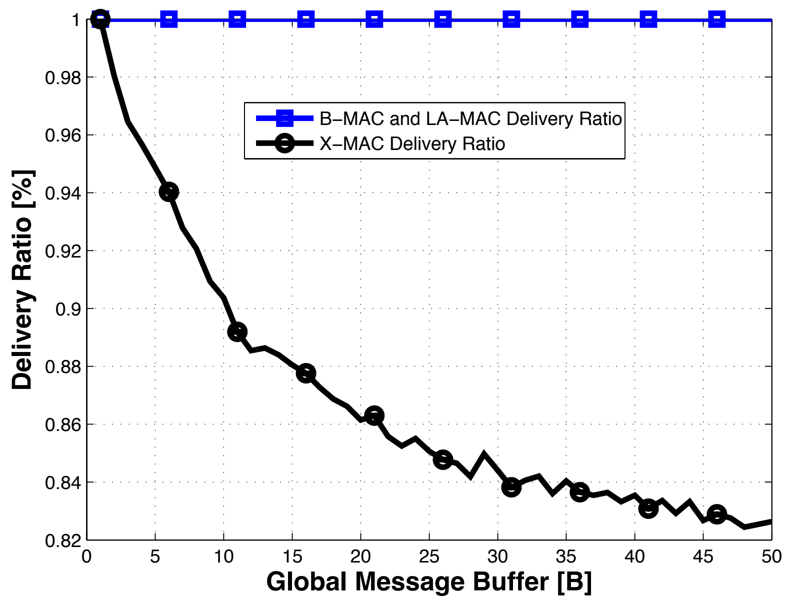

Figure 24: Delivery ratio vs. the global message buffer. In X-MAC, most collisions happen when messages are sent after the back-off time.

to send. Therefore, the probability that there si one of them that sends a preamble immediately after the start of polling process of the receiver is low, resulting in energy consumption similar to the pessimistic case.

In the simulation, all messages in the buffer are distributed among different buffers in a uniform way, so that all cases are possible. Thus, as traffic load increases, the number of senders increases as well so that the probability of having efficiently filled frame becomes higher and energy consumption lies in between the pessimistic and the optimistic curves.

LA-MAC is the most energy saving protocol and it also outperforms other protocols in terms of latency and the delivery ratio. We observe that when the instantaneous buffer size is lower than 2 messages, the cost of the SCHEDULE message is paid in terms of a higher latency with respect to X-MAC (cf. Fig. 25); however, for larger buffer sizes the cost of the SCHEDULE transmission is compensated by a high number of delivered messages. In Fig. 26, we show the percentage of the time during which devices spend in each radio mode versus the global buffer size. Thanks to the efficient message scheduling of LA-MAC, devices sleep most of the time independently of the buffer size and all messages are delivered. Resulting duty cycle (percent of simulation time spent in one of the active modes) is shown in Fig. 27. The figure shows that the trend of the duty cycle of LA-MAC differs from the one of B-MAC and X-MAC. The duty cycle trend of B-MAC and X-MAC shows

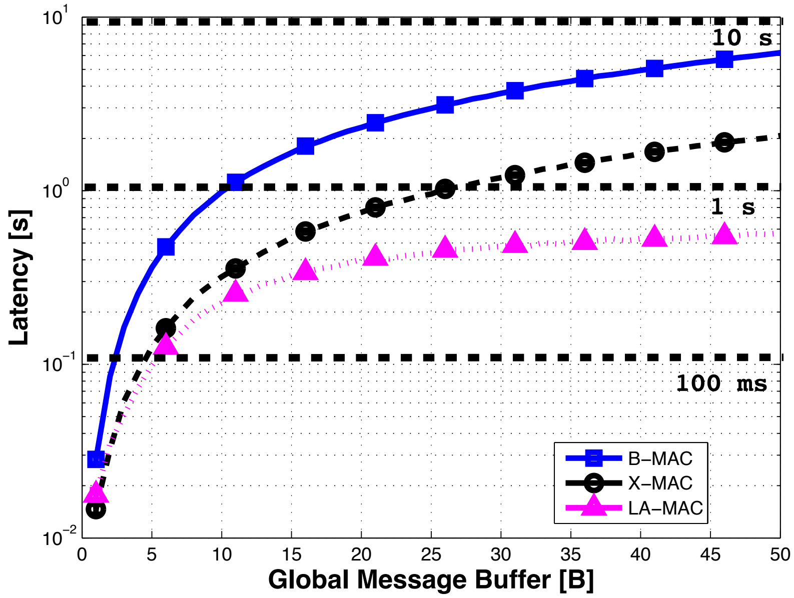

Figure 25: Average latency vs. the global message buffer.

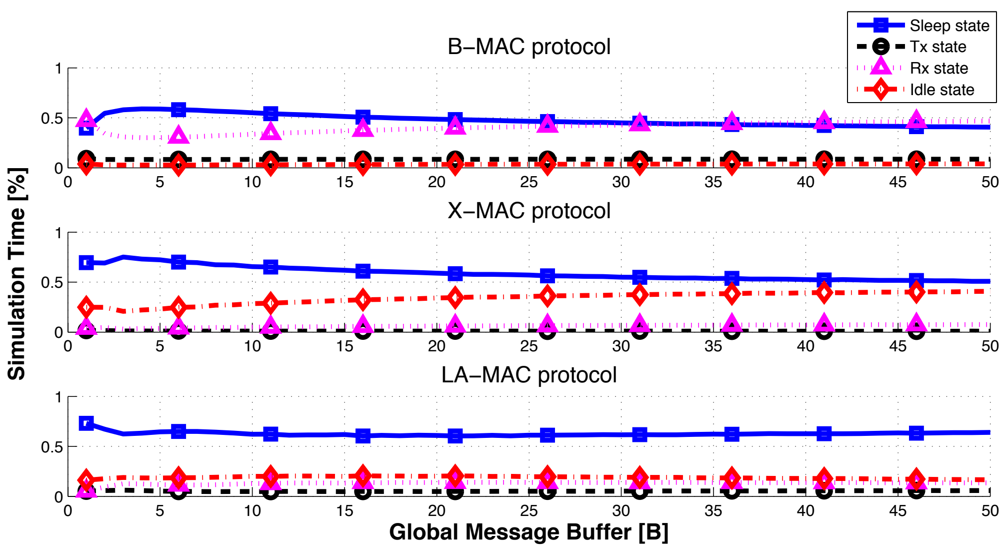

Figure 26: Percentage of the time spent in each radio mode vs. the global message buffer.

two phases: it first decreases until a value around $B = 3$ and then it increases with traffic load. With LA-MAC, the duty cycle shows a different behavior. It increases with traffic load until a value around $B = 15$, where it reaches its maximum value, and then it decreases.

In B-MAC and X-MAC, the reason for the decreasing phase comes from how the simulation environment is defined. When there is only one or few messages to send, the simulation ends in a short time, that is, as soon as the first sender has finished its transmission. If the simulations is short, the energy consumption of the active couple governs the duty cycle of the entire network. For example, consider the case with B=1. With B-MAC, the sender spends almost all the simulation time in transmission mode (excepting the time that it spends in polling mode before transmitting the preamble) (cf. Tab.Ia). As consequence, the other nodes *i.e.,* the receiver and the over-hearers spend most of the time in receiving mode because the probability of busy channel when they wake-up is high and they cannot go back to sleep until the end of data transmission. With X-MAC, simulations are shorter with respect to B-MAC resulting in lower duty cycle; however, the duty cycle shows the same trend.

The simulation duration increases with the value of B. In the second phase of duty cycle, that is when B is larger than 3, we observe that both X-MAC and B-MAC not only result in increasing energy consumption because simulations last more time, but also result in increasing duty cycle. With B-MAC, the duty increases because of the large amount of time that the receiver and over-hearers spend in reception mode. With X-MAC, the duty cycle increases because the number of packets that can be sent in a single frame is limited to two, resulting in high congestion when traffic load becomes heavy.

With LA-MAC, when there is only one message to send, the average simulation duration and duty cycle are in between the duration of X-MAC and B-MAC because of the use of SCHEDULE message. When B increases, the duty cycle increases as well until the maximum of 39.6% that is reached when $B = 15$ (cf. Tab. Ic). For values of B lower than 15, the duty cycle of LA-MAC is higher than the one of X-MAC because LA-MAC frames are not efficiently filled, then, the order of the curves is inverted. We observe that even though LA-MAC frames are not efficiently filled, the resulting delivery ratio and latency outperform the values of X-MAC.

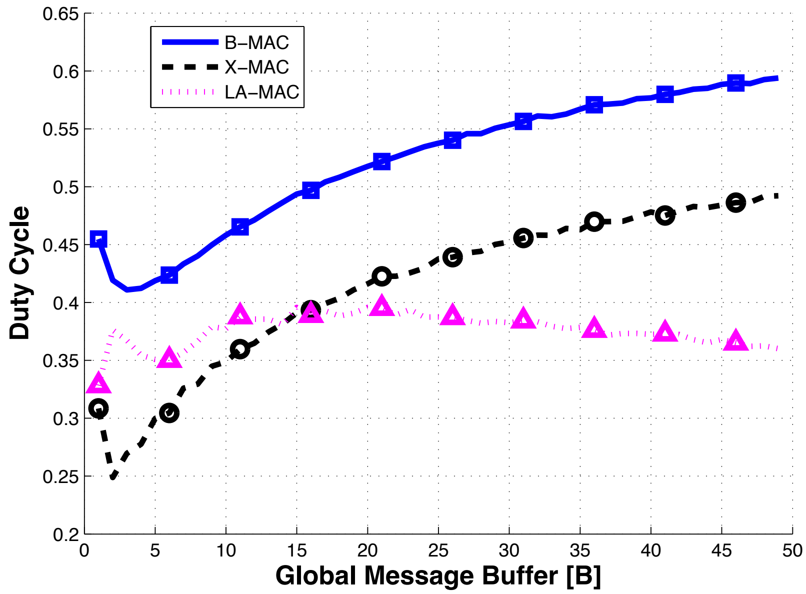

Figure 27: Duty cycle vs. the global message buffer.

Mode / B size	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 message	0.5452	0.0839	0.3422	0.0286	0.4548
3 messages	0.5890	0.0829	0.3013	0.0268	0.4110
5 messages	0.5812	0.0832	0.3080	0.0275	0.4188
15 messages	0.5064	0.0849	0.3761	0.0327	0.4936
30 messages	0.4465	0.0857	0.4313	0.0365	0.5535
50 messages	0.4061	0.0862	0.4686	0.0391	0.5939

(a) B-MAC

Mode / B size	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 message	0.6915	0.0157	0.0455	0.2472	0.3085
3 messages	0.7304	0.0115	0.0365	0.2216	0.2696
5 messages	0.7003	0.0111	0.0416	0.2470	0.2997
15 messages	0.6090	0.0097	0.0574	0.3239	0.3910
30 messages	0.5477	0.0094	0.0680	0.3749	0.4523
50 messages	0.5078	0.0093	0.0753	0.4075	0.4922

(b) X-MAC

Mode / B size	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 message	0.6717	0.0578	0.0931	0.1774	0.3283
3 messages	0.6328	0.0575	0.1244	0.1853	0.3672
5 messages	0.6491	0.0505	0.1144	0.1860	0.3509
15 messages	0.6043	0.0498	0.1396	0.2063	0.3957
30 messages	0.6175	0.0529	0.1388	0.1908	0.3825
50 messages	0.6399	0.0578	0.1355	0.1668	0.3601

(c) LA-MAC

Table I: Numerical details of time spent in each radio mode versus the traffic load per different values of B.

VI. Conclusions

In the present paper, we have analyzed the energy consumption of preamble sampling MAC protocols by means of simple probabilistic modeling. Analytic results are then validated by simulations. We compare the classical MAC protocols (B-MAC and X-MAC) with LA-MAC. Our analysis highlights the energy savings achievable with LA-MAC with respect to B-MAC and X-MAC. It also shows that LA-MAC provides the best performance in the considered case of high density networks under traffic congestion. The proposed analytic model is very flexible and can be used by MAC designers as an approach to understand the energy consumption of PS protocol in different congestion situations.

Acknowledgments

A part of this work has been performed in the framework of the ICT project ICT-5-258512 EXALTED partly funded by the European Commission. The authors Corbellini, Abgrall and Calvanese Strinati would like to acknowledge the contributions of their colleagues from the EXALTED project, although the expressed views are those of the authors and do not necessarily represent the project.

References

[1] "ICT-258512 EXALTED project." [Online]. Available: http://www.ict-exalted.eu/.

[2] K. Langendoen, "Energy-Efficient Medium Access Control," *Book chapter : Medium Access Control in Wireless Networks, H. Wu and Y. Pan (editors), Nova Science Publishers*, 2008.

[3] Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," *in Proc. of Annual IEEE International Conference on Computer Communications, INFOCOM*, vol. 3, pp. 1567–1576, 2002.

[4] T. van Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," *in Proc of. ACM International Conference on Embedded Networked Sensor Systems, SenSys*, pp. 171–180, 2003. [Online]. Available: http://portal.acm.org/citation.cfm?id=958512

[5] L. van Hoesel, P. Havinga, "A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks: Reducing Preamble Transmissions and Transceiver State Switches," *in Proc. of IEEE INSS*, Tokyo, Japan 2004.

[6] G. Lu, B. Krishnamachari, and C. Raghavendra, "An Adaptive Energy-efficient and Low-latency MAC for Data Gathering in Wireless Sensor Networks," *in Proc. of 18th IEEE IPDPS*, p. 224, 2004.

[7] V. Rajendran, J. J. Garcia-Luna-Aceves, and K. Obraczka, "Energy-Efficient, Application-Aware Medium Access for Sensor Networks," *in Proc. of IEEE International Conference on Mobile Ad-hoc and Sensor Systems, MASS*, pp. 8–pp, 2005.

[8] A. El-Hoiydi, "Aloha with Preamble Sampling for Sporadic Traffic in Ad Hoc Wireless Sensor Networks," *in Proc. of the Int. Conf. on Communications*, vol. 5, pp. 3418–3423, New York, NY, USA, April 2002.

[9] E-Y. Lin, J. Rabaey, A. Wolisz, "Power-Efficient Rendez-vous Schemes for Dense Wireless Sensor Networks," *in Proc. of the Int. Conf. on Communications*, Paris, France, June 2004.

[10] J. Polastre, J. Hill and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," *in Proc of. ACM International Conference on Embedded Networked Sensor Systems, SenSys*, 2004.

[11] M. Avvenuti, P. Corsini, P. Masci, and A. Vecchio, "Increasing the Efficiency of Preamble Sampling Protocols for Wireless Sensor Networks," *in Proc. of the First Mobile Computing and Wireless Communication International Conference, MCWC*, pp. 117–122, 2006.

[12] S. Mahlknecht and M. Boeck, "CSMA-MPS: A Minimum Preamble Sampling MAC Protocol for Low Power Wireless Sensor Networks," *in Proc. of IEEE Workshop on Factory Communication Systems*, pp. 73–80, Vienna, Austria, September 2004.

[13] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: a Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks," *in Proc of. ACM International Conference on Embedded Networked Sensor Systems, SenSys*, pp. 307–320, 2006. [Online]. Available: http://portal.acm.org/citation.cfm?id=1182807.1182838

[14] R. Kuntz, A. Gallais, and T. Noel, "Auto-adaptive MAC for energy-efficient burst transmissions in wireless sensor networks," *in Proc. of IEEE Wireless Communications and Networking Conference, WCNC*, pp. 233 –238, march 2011.

[15] G. Corbellini, E. Calvanese Strinati, E. Ben Hamida, and A. Duda, "DA-MAC: Density Aware MAC for Dynamic Wireless Sensor Networks," *in Proc. of 22nd IEEE Personal Indoor Mobile Radio Communications (PIMRC'11 - LPAN)*, pp. 920–924, September 2011.

[16] C. Enz, A. El-Hoiydi, J. Decotignie, V. Peiris., "WiseNET: An Ultra-Low-Power Wireless Sensor Network Solution," *IEEE Computer Society Press*, vol. 37, no. 8, pp. 62–70, August 2004.

[17] W. Ye, F. Silva and J. Heidemann, "Ultra-Low Duty Cycle MAC with Scheduled Channel Polling," *in Proc of. ACM International Conference on Embedded Networked Sensor Systems, SenSys*, pp. 321–334, Boulder, CO, USA, November 2006.

[18] G. Corbellini, E. Calvanese Strinati, and A. Duda, "LA-MAC: Low-Latency Asynchronous MAC for Wireless Sensor Networks," in *submitted for publication*.

[19] C. Wan, S. Eisenman, A. Campbell, and J. Crowcroft, "Siphon: Overload Traffic Management Using Multi-radio Virtual Sinks in Sensor Networks," *in Proc of. ACM International Conference on Embedded Networked Sensor Systems, SenSys*, pp. 116–129, 2005.

[20] "OMNeT++ Discrete Event Simulator," http://www.omnetpp.org.

[21] "Texas Instruments, CC1100 datasheet," http://focus.ti.com/docs/prod/folders/print/cc1100.html.