Thorough Empirical Analysis of X-MAC over a Large Scale Internet of Things Testbed

Julien Beaudaux, Antoine Gallais, Julien Montavont, Thomas Noel, Damien Roth and Erkan Valentin

Abstract—The last decade saw the emergence of the Internet of Things paradigm, which aims to connect any object to the Internet. As in Internet of Things networks the radio component is usually the main source of energy depletion, specific medium access control protocols have been designed to reduce its usage. Among those, X-MAC has gained much interest in recent years, and has become one of the most popular MAC protocols in the Internet of Things community due to its many theoretical advantages. X-MAC is a preamble-sampling protocol based on the Low Power Listening mechanism, in which the nodes decide their schedule independently from their neighbors. So far, no study has ever identified malfunctions that appear in the protocol design. In addition, performance evaluations were only made through theoretical studies and simulations, or relatively limited scale experiments. In this article, we point out several omissions and black spots that appear in the original protocol design and that lead to encountering severe underachievements. We also provide explanations to these flaws and propose solutions that solve or at least reduce their impact. Finally, we have led a thorough experimental campaign on an indoor static grid composed of 240 nodes. From those studies, we quantify the performances of the original X-MAC design and the impact of our improvements.

Index Terms—Internet of Things; Medium Access Control; X-MAC; Testbeds; Experimental study

1 INTRODUCTION

Internet of Things (IoT) networks are composed of numerous sensor and actuator devices that monitor and interact with their environment [1]. Most IoT nodes are limited in terms of energy, memory and computation. In consequence, protocols design should remain as light and simple as possible and pay attention to energy consumption.

In IoT networks, wireless communications are the main factor of energy depletion [1]. As the wireless medium is a shared resource, its access and use must be controlled in an energy-efficient fashion. Over the years, numerous Medium Access Control (MAC) protocols have been designed to address the specificities of IoT networks [2]. Among these, X-MAC [3] has become one of the most popular propositions, thanks to its scalability and simplicity. Indeed, many contributions [4], [5] and real deployments [6] rely on this protocol.

Despite the popularity and broad usage of X-MAC, its original design presents several omissions and black spots that have not been pointed out yet. These flaws concern both the general design and optional improvements proposed by the original article, which may lead to severe underachievements. In this paper, we perform a thorough analysis of X-MAC: we identify and detail each malfunction in its design and propose alternatives or improvements to resolve them.

In addition, no in-depth empirical analysis of X-MAC performances has been achieved yet. Most of the former studies have been performed by either theoretical analysis, simulation or very limited scale experiments, which may lead to drastic simplifications [7]. Specifically, those experimental campaigns have not evaluated the scalability of X-MAC, due to testbeds of limited size.

In this article, we quantify, comment and explain the impact of realistic conditions and standard X-MAC configurations on network performances. We also evaluate our enhancements of X-MAC. We finally emphasize the gains in terms of message loss, channel utilization and overall latency in the network. This experimental campaign was led on the 240 node FIT IoT testbed. The X-MAC source code used in our experimental campaign is available at [8] for cross validation.

The remainder of this paper is organized as follows. Section 2 provides the required background on MAC in IoT networks. Section 3 describes the basics of X-MAC, as initially presented in the original paper [3]. In Section 4, we highlight some omissions or problematic situations that were not tackled in prior papers, and provide leads to solve or attenuate their effects. Section 5 introduces our empirical analysis of X-MAC and focuses on several problematic situations. Our investigations to clarify or solve such situations are presented in Section 6. Section 7 presents several prior studies on X-MAC. We conclude with final remarks and future investigations in Section 8.

2 MEDIUM ACCESS CONTROL IN IOT

Since the emergence of the IoT paradigm, numerous contributions have been developed to cope with its

• *Julien Beaudaux, Antoine Gallais, Julien Montavont, Thomas Noel, Damien Roth and Erkan Valentin are with the ICube Laboratory, University of Strasbourg, France. E-mail: {lastname}@unistra.fr*

This work was partly funded by the Alsace Region, France

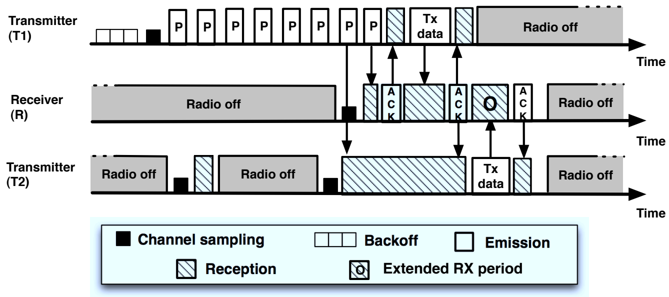


Fig. 1. X-MAC preamble-strobing and SMT mechanisms

specificities (e.g. limited energy) at the MAC layer [2]. In this section, we present major contributions at this level and detail why X-MAC represents a serious alternative to the dedicated standard.

In order to address the specificities of emerging IoT networks, the IEEE consortium developed the 802.15.4 MAC standard [9]. This standard is constituted of two modes : contention-based mode and the synchronized mode. In contention-based mode, the radio remains perpetually active, and as such no energy gain is obtained. In synchronized mode, all the nodes belonging to the same cluster alternate active and passive periods of the radio at the same rhythm. Each node is being attributed a specific slot, during which it will be the only one allowed to transmit. Establishing and maintaining synchronization is a very difficult task, hardly scalable, very sensitive to topology changes and requires an important overhead [10]. Thus, this mode remains adapted for limited-scale deployments only.

In parallel, the Low Power Listening (LPL) gained in popularity [11]. LPL aims at reducing energy consumption through duty-cycling, by putting the radio in sleep mode as often as possible and for long periods of time. Also, operating in a fully localized fashion alleviates the impact of scale changes. The X-MAC protocol belongs to this category, and can compare with the synchronized mechanisms in terms of performance (e.g. throughput, energy consumption), while remaining asynchronous [12].

Many real IoT deployments and recent scientific contributions rely on this protocol [4], [12], [13]. Moreover, it was already pointed out that X-MAC is a serious alternative to the MAC layer defined in the 802.15.4 standard [6]. Its interest was also demonstrated in IPv6 IoT networks [14].

3 X-MAC PROTOCOL DESIGN

In recent years, X-MAC has gained much interest from the IoT community. Its low power consumption, offered throughput and simplicity make it particularly adapted for IoT constraints and deployments. In this section, we detail the basics of its design and point out important features on which we have focused our in-depth study.

3.1 X-MAC adaptation of the LPL mechanism

X-MAC is an asynchronous duty-cycled MAC protocol. It is mainly designed to optimize the use of the Low Power Listening mechanism (LPL). In LPL, each node turns into a low-power sleeping mode and regularly samples the medium during short intervals. This requires any transmitter to send a preamble before its data in order to awake sampling neighbors that then wait for data transmission. Note that all neighbors will turn into the receiving mode while the message might only target one of them. Such a situation leads to overhearing for all the other neighboring nodes. Actually, the target identifier (ID) being specified in the data only, any node hearing the preamble should wait for the data frame in order to determine whether or not it should be processed.

X-MAC was designed to limit such overhearing for non-recipient nodes. To do so, the preamble is split into small strobes. Each of these includes the target ID and thus can be individually acknowledged. This way, nodes detecting a strobe with a different ID from their own can switch their radio off, thus mitigating overhearing on non-recipient nodes. Furthermore, as the strobes stop as soon as the target node is ready to receive, this results in a reduction of the radio channel occupancy. Figure 1 illustrates the message transmission in X-MAC. It depicts the use of strobed preambles with embedded destination ID that leads to shortened activity periods for both transmitting, receiving and overhearing nodes (respectively referred to as $T1$, R and $T2$), and thus reduces channel utilization and energy consumption at the same time.

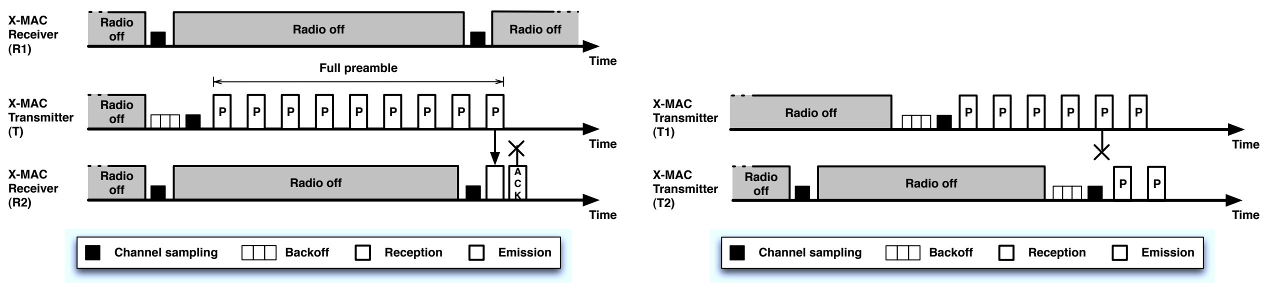
3.2 Simultaneous multiple transmitters

X-MAC includes an additional mechanism which improves channel utilization once multiple transmitters simultaneously schedule a transmission. For the sake of uniformity and simplicity, this mechanism is referred to as simultaneous multiple transmitters (SMT) in the present paper. SMT works as detailed in Figure 1.

At the end of a data transmission from a node $T1$, the receiving node R sends a data acknowledgment ($DACK$) and adds a short overhearing period (designated as **O** in the legend of Figure 1). By doing so, R can catch preambles from other neighbors, and directly deal with the corresponding message. In addition, let us assume that $T2$ has data destined to R. If $T2$ has overheard the preamble sent by $T1$ to R, then $T2$ delays its transmission until the $DACK$ emission. Thus, when R enters the extended overhearing period, $T2$ can directly send its data frame. In order to avoid the synchronization of multiple transmitters, a random backoff period is used before sending the data frame in the extended overhearing periods.

As a result, SMT allows nodes to skip the transmission of a preamble for certain messages. In addition, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



(a) Preamble acknowledgement

(b) Preamble interruption

Fig. 2. Black spots in the X-MAC design

receiver (R) and overhearing nodes (such as $T1$ depicted on Figure 1) save the reception of a preamble.

3.3 Targeted challenges

Overall, all of the mechanisms introduced by X-MAC (i.e. dividing preamble into strobes, including destination information in the strobes, SMT) show that X-MAC is especially designed to increase throughput and reduce excessive preambles, and thus wasted energy.

In addition, X-MAC has come with a set of interfaces that allow users and applications to access the primitive settings of the MAC layer. Several parameters can be adjusted (e.g. sleep duration, maximum number of retransmissions, number of preamble strobes) in order to comply with the specificities of the application. For instance, other protocols make use of this high configurability to improve end-to-end delay [5]. Yet, how to optimally configure the MAC layer remains an open problem. On the one hand, targeting low latency communication would require short duty-cycles that increase energy consumption due to frequent radio initialization operations. On the other, aiming at low energy consumption would tend to stretch the sleeping periods out along with the preamble length, thus increasing channel occupancy and reducing throughput accordingly.

We now investigate to what extent the X-MAC protocol meets these goals once facing a large-scale WSN deployment.

4 Foreseen black spots in X-MAC

Our thorough study of the X-MAC protocol has led us to discover artifacts and missing details in its design. Several specific situations may imply underachievements and malfunctions compared to the expected behavior. Moreover, some parts of its description were not sufficiently detailed or even omitted. To the best of our knowledge, neither abnormal behaviors nor omissions have ever been tackled in subsequent papers evaluating X-MAC. We aim at completing the X-MAC design in order to leave as few ambiguities as possible when facing a real situation with explicit deployment and application constraints.

4.1 Preamble acknowledgement

In X-MAC, when a node is willing to send a data to one of its neighbors, it first emits a streak of preamble strobes (c.f. Section 3.1), and eventually starts the data frame transmission upon reception of the preamble acknowledgement (*PACK*). However, if the *PACK* is not received, it is never detailed whether or not the data frame should be sent after the preamble. This situation can either occur when the receiver does not catch the preamble (and thus does not prepare itself for data reception) or when the PACK is lost.

Two possible interpretations arise. First, if the preamble transmission ends without any *PACK* reception, the data is transmitted anyway. This solution is risky, since the sender does not know if the receiver is ready to receive data. In consequence, this solution may increase the packet loss. Second, if the preamble ends without having been acknowledged, we could deduce that the receiver is not ready and thus schedule a new transmission. This solution may lead to increased channel occupancy and delay, as a complete preamble was sent for nothing.

This scenario is detailed in Figure 2(a). When reviewing the X-MAC design, these different solutions lead to non deterministic behaviors. Indeed, if the preamble is not perceived by the target (as for $R1$), the second solution should prevail. On the other hand, if the preamble is perceived but the *PACK* is lost (as for $R2$), the first solution should be chosen.

4.2 Preamble interruption

According to the original X-MAC design (see Section 3.1), the preamble strobes should be sent until the $PACK$ is received or until the preamble maximum length (i.e. maximum number of strobes) is reached. However, in [3], there is no detail concerning the situation where the source node perceives noise in the radio channel while waiting for a $PACK$. This situation can occur when two nodes send their preambles at the same time but only one of them can hear the other sender (i.e. the link between these two nodes is not bidirectional). Figure 2(b) illustrates such situation.

One interpretation is that the remaining strobes are still emitted, provided that the received information is

Black spots	Interpretations	Potential consequences
Preamble ends without being acknowledged	transmit data	$\nearrow$ packet loss
	schedule retransmission	$\nearrow$ delay and channel occupancy
Noise during the preamble emission	continue the preamble	$\nearrow$ packet loss
	schedule retransmission	$\nearrow$ delay and channel occupancy
No evaluation of SMT		$\nearrow$ throughput and $\searrow$ latency

TABLE 1
Foreseen black spots and available alternatives

not a *PACK*. As this specific situation is not handled in the original paper, we assume this to be the default behavior of the initial X-MAC protocol. This behavior may lead to increase packet loss, as the medium may be occupied by one or more nodes. Another interpretation would be that, during the preamble period, if the transmitter receives any other kind of information than the $PACK$ it is waiting for, it can consider the preamble as unacknowledged. Thus, the source node immediately interrupts its transmission and schedules a further one. Typically, this is how X-MAC is provided by Contiki OS. We advocate that such interruption should not be considered as a loss at the MAC layer (e.g. it is not considered in the maximum number of retransmission counter). Indeed, the preamble interruption mechanism is designed to avoid the loss of the pending data. However, interrupting the preamble may increase delays and channel occupancy due to a new transmission of the preamble after a random backoff.

4.3 Potential benefits of SMT mechanism

According to the original design of X-MAC, the SMT mechanism (see Section 3.2) aims at increasing throughput while reducing energy consumption by attenuating overhearing. Note that the real impact of this mechanism has never been evaluated, neither in the original paper nor in subsequent ones.

However, using SMT requires any receiving node to stay awake in order to be able to potentially receive new messages (i.e. data frames without preamble or preamble strobes). In addition, senders willing to benefit from SMT have to listen the ongoing transmission to detect the extra RX period. The cost of this optimization is an extended overhearing period for both SMT transmitters and receivers, while the benefit depends on the proportion of senders that take advantage of this mechanism. Consequently, assertions such as SMT *"allows for lower latency and higher throughput"* are questionable.

To analyze the real impact of the mechanisms presented here, we conducted a comparative evaluation of X-MAC with different scenarios. Our evaluation focuses on performances, and specifically on loss-rate and access time. Each black spot is detailed in Table 1, together with the available interpretations and potential consequences.

5 EXPERIMENTATION SETUP

Setting up a complete WSN deployment is a very complex task [15]. In this section, we review the various choices made during the implementation on the FIT IoT testbed and the ensuing large-scale X-MAC experiments.

5.1 Platform specifications

Our thorough empirical analysis of X-MAC is conducted over the large scale FIT IoT testbed. FIT is composed of 1024 nodes equally distributed over four sites. In our experiments, we have used the platform installed in Strasbourg, France. This testbed is structured as a 3D grid of 240 nodes. The X-MAC source code used in our experimental campaign is available at [8] for cross validation.

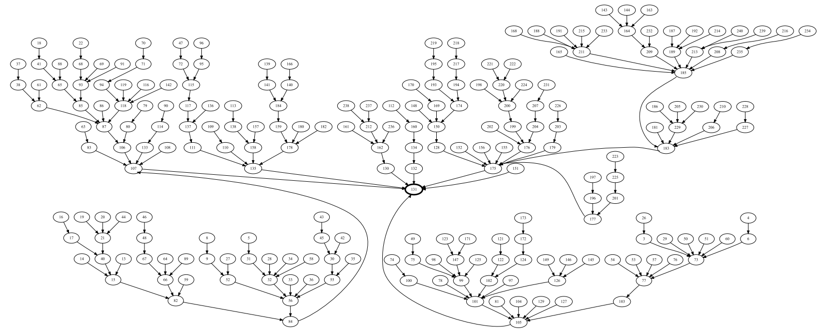
Each node is equipped with a Texas Instruments CC1101 radio chipset and a MSP430 micro-controller. One of the most important parameter of the radio component is the Clear Channel Assessment (CCA) threshold, which indicates if the radio channel is free or busy. In fact, while the CCA is enabled, packets will only be transmitted if the channel is free. Obviously, changing parameters (e.g. sampling duration, strobe length, backoff duration) affects the radio behavior and results of the experiment (e.g. time required to send a message, listening sensitivity). For our experiments, we choose to retrieve values provided by the radio configuration software SmartRF, in order to ensure the proper communication between the nodes. This software returns coherent hardware parameter values complying with criteria provided as parameters (e.g. sleep period length, data packet size).

5.2 Application, routing and MAC details

At the beginning of an experiment, up to 25 nodes are randomly selected as data sources, resulting in an average of 56 additional nodes participating in the multi-hop communication. These sources implement a time-driven application model and transmit data every 8 seconds. We chose a 7 byte data size, which corresponds to the general information used by monitoring applications (e.g. node ID, position, time, sensed value) [16]. The sink is located at the center of the testbed.

The network uses a gradient-based routing protocol [17] which generates a virtual routing tree rooted

(a) A view of the testbed physical topology



(b) Routing topology

Fig. 3. FIT : a very large-scale open IoT testbed - Site of Strasbourg, France

at the sink. This routing protocol was chosen for its performance under realistic conditions [15]. Moreover, its simplicity and low overhead properties allowed us to focus on MAC protocol performance by affecting the results as little as possible. By doing so, a rank and a next hop toward the sink are attributed to each node of the network. Figure 3(b) illustrates a virtual instance of the network organization obtained after the gradient construction.

We also implemented X-MAC in order to strictly comply with the original version detailed in [3]. The Low Power Listening (LPL) process is managed by the wake on radio mechanism included in the CC1101 chipset. With the CC1101 chipset, turning the radio into the sleep mode consumes about 500 nA while it often reaches 17 mA in both the receiving and transmitting modes. In this case, the sleep mode divides up to 34000 times the energy consumption. Here, the sleep period duration is fixed to 100 ms and the listening time is fixed to 15 ms. This 13% duty-cycle divides energy consumption by a factor of more than 7. The total number of preamble strobes is fixed to 15, so that the preamble length is slightly above LPL period. Such configuration ensures a good trade-off between loss-rate and duty-cycle in such conditions, as observed in several related works [5]. An overview of our experimental parameters is given in Table 2.

5.3 Selecting high-quality radio links

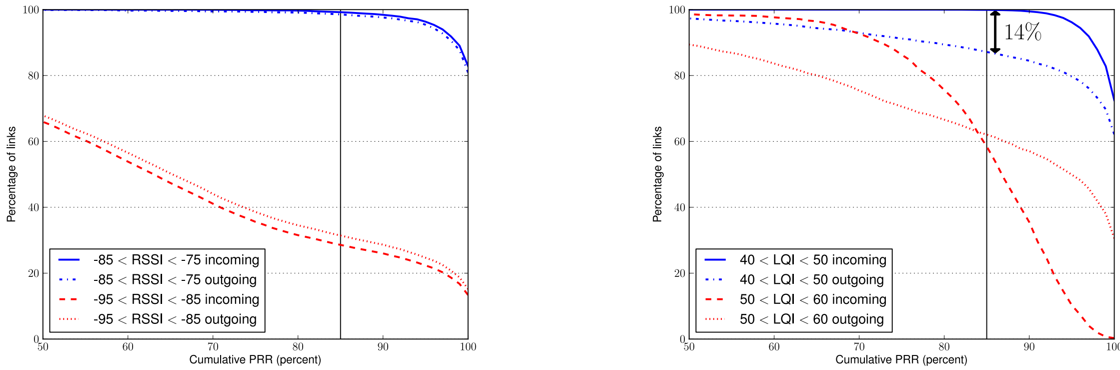
X-MAC requires bidirectional links so that two nodes can exchange information (e.g. preamble, acknowledgements, data). However, wireless links are lossy, unstable, unreliable and sensitive to environmental conditions [18]. In the real-world, there is no guarantee that the links used for the network communications will be bidirectional. During our experimental campaign, we aimed at removing one hypothesis at a time. Thus, we first decided to take into account only high-quality bidirectional links. In order to guarantee this assumption, we evaluated several metrics likely to identify such links.

The quality of a link can be reflected by its Packet Reception Rate (PRR). Yet, this value can only be estimated by averaging the success or failure of several transmissions. In addition, the PRR can not provide any information on the bidirectionality of a link. In consequence, we evaluated alternative metrics provided by the radio chipset to identify high-quality bidirectional links without the need for preliminary traffic. High-quality links refer to links with a PRR greater than 85%. Most radio chipsets provide two indicators that could be used for this purpose, the Received Signal Strength Indication (RSSI, expressed in dBm) and the Link Quality Indicator (LQI). Although the RSSI value is identically calculated for all radio chipsets, LQI differs for each of them. The RSSI value is an estimate of the signal power level in the radio channel. For CC1101 radio chipsets, the LQI estimates how easily a received signal can be demodulated.

Experimental parameters	**Value**
Deployed nodes	240 indoor static nodes
Testbed organization	10 $m \times 8$ $m \times 3$ m regular grid
Nodes spacement	One meter
Number of sources	5, 10, 15, 20, 25
Duration	10 minutes, repeated 10 times
Routing model	Gradient-based routing
Application model	Constant bit-rate : 1 message / 8s
Data packet size	7 bytes (+6 byte MAC header)
X-MAC parameters	**Value**
Maximum retries	5
Queue length	8 messages
Total preamble length	146 ms maximum (15 strobes)
Sleep period length	100 ms
Listening time	15 ms
SMT extended RX period	20 ms
Initial Backoff	From 30 ms to 56 ms
Interruption Backoff	From 10 ms to 56 ms
SMT Backoff	From 0 ms to 15 ms
Hardware parameters	**Value**
Antenna model	Omnidirectional CC1101 interface
Frequency	868MHz
Modulation	76.8 kBauds GFSK
Emission power	-30 dBm (range: ± 2.5 m)
Battery	880 mAh, 3.7 V

TABLE 2
Experiment parameters

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



(a) Correlation between RSSI value and link PRR

(b) Correlation between LQI value and link PRR

Fig. 4. Interest of RSSI and LQI to evaluate link quality and bidirectionality

Former contributions [19], [20] already proposed mechanisms to estimate the quality of radio links. However, most of these are hardly applicable as they require preliminary traffic or do not take into account the link bidirectionality. Therefore, we here provide an evaluation of the correlation between both RSSI or LQI and PRR on our testbed. For this, we have conducted experiments in which each node broadcasts a hundred messages in turn, so that we can determinate average RSSI, LQI and PRR for each existing link. The results are given in Figure 4. Each curve represents the set of links for which the RSSI or LQI is within a given range. For each set, we display the percentage of links for which the PRR is greater than a certain value (given on the Y-axis). The vertical line represents the 85% PRR threshold from which we consider a link as high-quality.

Figures 4(a) compare the RSSI value perceived by the receiver with the PRR of the incoming link ($Transmitter \rightarrow Receiver$) and the PRR of the outgoing link ($Receiver \rightarrow Transmitter$). Figures 4(b) makes the same comparison with LQI values. By this mean, we aim at finding a correlation between the RSSI or LQI values with the bidirectionality of links. The results indicate that both RSSI and LQI can help us identify high-quality incoming links. About 100% of links with an RSSI value greater than $-85\ dBm$ or an LQI value lower than 50 have a PRR equal or greater than 85% (the X-axis is cumulative). In parallel, the results shows that a RSSI value greater than $-85\ dBm$ indicates a high-quality outgoing link. Above this RSSI value, more than 98% of the associated outgoing links have a PRR greater than 85%. High-quality bidirectional links can thus be taken into consideration by processing packets with a RSSI value above $-85\ dBm$ only. By contrast, 14% of the message with a LQI value below 50 are associated with an outgoing link with a PRR lower than 85%. As a result, the LQI value can help identify high-quality links but, contrary to the RSSI, fails at identifying bidirectional ones.

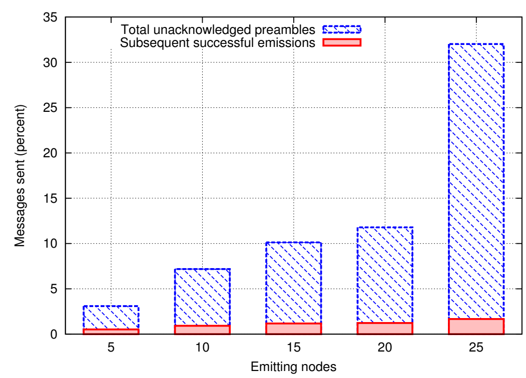


Fig. 5. Percentage of successful emissions consecutive to unreceived PACK

6 Experimentation Results and Analysis

The experiments presented in Section 5.3 allowed us to only consider high-quality bidirectional links in our testbed, in order to provide a fair analysis of X-MAC (that requires bidirectional links to operate). In this section, we investigate thoroughly each black spot detailed in Section 4 in order to measure and quantify their respective impacts on X-MAC performances. When necessary, we implemented and evaluated an alternative version to solve subsequent underachievement. Each set of experimentations uses identical routing topologies. The results presented in this section are an average of the overall data collected on the set of experimentations. The 95% confidence interval indicates the reliability of our measurements.

6.1 Preamble acknowledgement

As explained in Section 4.1, the X-MAC specifications do not mention which behavior to adopt at the end of the preamble emission when no $PACK$ is received. Sending the data anyway can be risky, since the non-reception of $PACKs$ can mean that the destination node is not ready to receive (i.e. it was deaf to the preamble). As a result, the data frame is likely to be lost. On the contrary, considering the preamble as failed would lead to the retransmission of an additional preamble for the same data frame, while the destination node might be ready to receive (e.g. if the $PACK$ was lost). Both scenarios are illustrated in Figure 2(a). We experimented both available solutions, by either sending data anyway and waiting for the associated $DACK$, or considering the preamble as failed and scheduling another transmission.

For each set of experiments, we quantified the percentage of unacknowledged preambles. As depicted in Figure 5, between 3% and 32% of emitted preambles are not acknowledged. Among these, at most 17% of subsequently emitted data are acknowledged by a $DACK$ (i.e. the $PACK$ was sent but lost). Otherwise, the preamble was not caught by the target node. Consequently, sending the data while the preamble is not acknowledged only favors 2% of overall emitted messages.

In the light of these results, sending the data packet while the preamble was not acknowledged is a risky choice as the probability of losing the data packet is very high. Thus we recommend considering the preamble as failed whenever it is not acknowledged. This behavior will be used during further experiments.

6.2 Preamble interruption

Then, we experimented two methods for managing the perception of noise (e.g. data intended to another node) during preamble emission. This case is detailed in Section 4.2 and illustrated in Figure 2(b). First, the source continues to transmit the remaining strobes. Otherwise, the preamble is interrupted and rescheduled for a further transmission.

We first quantified the packet loss rates occurring at the MAC layer for each solution. As shown in Figure 6, carrying on sending preamble strobes increases the packet loss rate by 33% to 50%. This behavior affects both preamble and data transmissions, as both loss-rates increase symmetrically. Also, as detailed in Table 3, less than 14% of the preambles are disturbed by noise during their transmission, and are thus responsible for this loss-rate increase. Such situation mainly occurs when two nodes are using the medium at the same time, while they are only able to hear each other sporadically (i.e. not having a high-quality bidirectional link). As a result, continuing the preamble emission may interfere with their transmission, leading to collisions and losses.

We also predicted that interrupting the preamble emission in such situation could increase access-time and channel occupancy, as it requires the whole preamble to be rescheduled. In order to measure this impact, we quantified the total number of strobes emitted before a packet is successfully sent to its destination or dropped in Figure 7. The dashed curve represents the transmissions in which preambles were not interrupted whenever noise was detected during the transmission, while the solid line only represents the transmissions in which

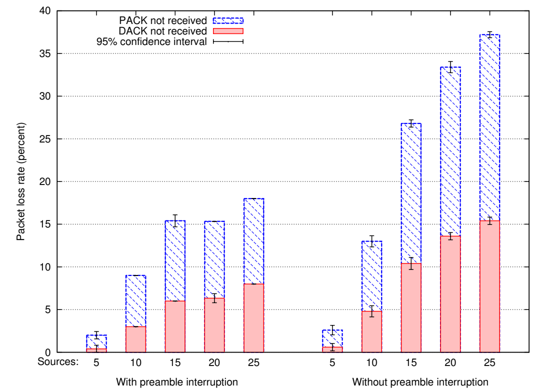


Fig. 6. Impact of preamble interruption on loss rates

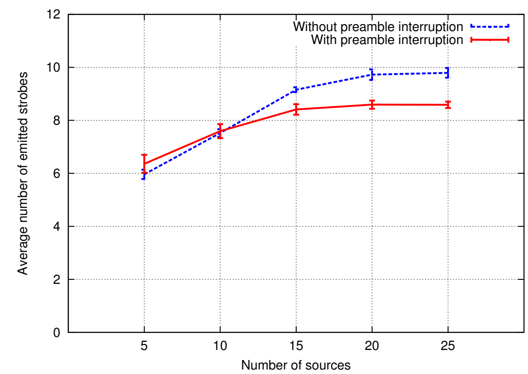


Fig. 7. Impact of interruption on preamble length

preambles were interrupted. We can notice that the average number of strobes required to transmit a data is quite similar whatever the preamble is interrupted or not. Indeed, as exposed in Table 3, the vast majority of interrupted preambles are stopped after 4 strobes in average. We also observed in our experimentations that both access-time remained identical, whatever the adopted solution. Without preamble interruption, most of the transmissions for which the preamble is disturbed by noise usually fail and thus lead to a retransmission. Also, the delay induced by a further preamble retransmission (from $10\ ms$ to $56\ ms$) is too low to significantly impact the access-time.

Finally, we measured the end-to-end delay and throughput of each interpretation, respectively in Figures 8 and 9. In our setup, the considered applicative model implies a small gap between two sequences of message transmission (when all the nodes send their message following a random backoff). During this period, the network is not facing any contention. While taken into consideration, this gap strongly affects the overall throughput. In order to exclude this external phenomenon and focus on the MAC performance, we also provided the network throughput for contention periods only (i.e. when a sequence of message transmission is ongoing). The provided results validate that interrupting the preamble does not impact negatively the network performance, and can even improve it,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

Source nodes	5	10	15	20	25
Proportion of interrupted preambles	1.38% [±0.21]	4.33% [±0.56]	9.08% [±0.40]	12.48% [±0.45]	13.04% [±0.53]
Strobes emitted before interruption	4.78 [±1.72]	4.66 [±0.64]	4.80 [±0.32]	4.51 [±0.23]	4.52 [±0.20]

TABLE 3
Interrupted preambles

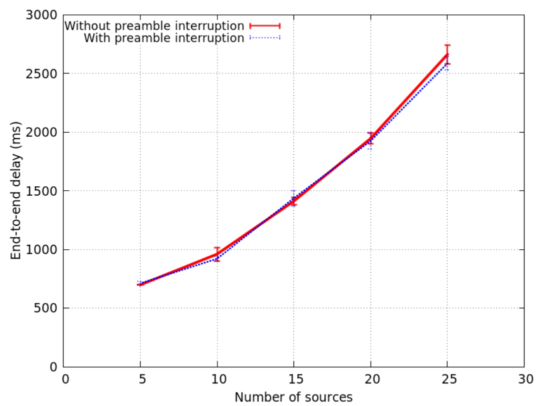


Fig. 8. Impact of preamble interruption on end-to-end delay

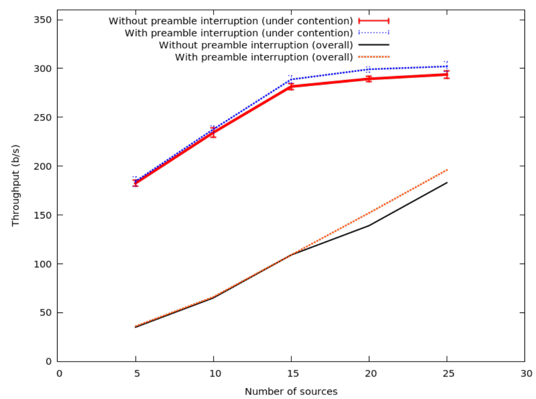


Fig. 9. Impact of preamble interruption on throughput

especially with an increased number of source nodes (i.e. at most 11% improvement for end-to-end delay and 3% for throughput).

In the light of the results, we can conclude that interrupting the preamble emission reduces the number of losses at the MAC layer, without impacting the time to send a data packet, the channel occupancy nor the delay. Consequently, we state that the preamble emission should be interrupted when managing the perception of noise. We used this behavior by default during the next experimentations.

6.3 Interest of SMT mechanism

As underlined in Section 3.2, the SMT mechanism was originally designed to reduce excessive preambles. However, such assertions were never verified nor quantified. As SMT concerns a non-negligible part of the data frames sent (between 0.18% and 6.6% depending on the number of sources), we evaluated its impact on the network performances.

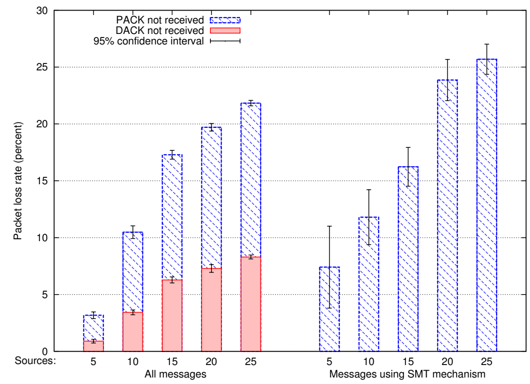


Fig. 10. Impact of SMT mechanism on loss-rates

First, we separately evaluated the average loss-rate for classical messages (not using the SMT mechanism) and for those which profit from this mechanism (which do not require a preamble to send data). The results are shown in Figure 10. We can notice that the messages transmitted with SMT have a loss-rate equal or higher than classical messages. This effect may be due to the fact that all transmitters using SMT are synchronized as they wait for the extended overhearing period to send their data frame. Multiple transmitters may try to use SMT at the same time and collisions might occur between data packets during the extended overhearing period, leading to packet loss. We observe that the backoff used before sending the data frame is insufficient to regulate access to the medium. Also, the access-time remains identical when using SMT or not. Indeed, the proportion of nodes impacted is too small to have a significative impact on this value.

In order to evaluate the general impact of the SMT mechanism, we also measured the end-to-end delay and throughput for each scenario (i.e. using the mechanism or not). As for the results relative to preamble interruption, we evaluated the throughput both for the overall experiment and only when a sequence of message transmission is ongoing. The results provided in Figure 11 demonstrate that X-MAC *"allows for lower latency"*, as stated in the original paper. Indeed, with a large number of sources, the use of the SMT mechanism (and thus the avoidance of any preamble transmission for some messages) leads to reduced delay (i.e. at most 11% with 25 sources). However, the equivalent assertion regarding throughput (i.e. X-MAC *"allows for higher throughput"*) do not hold. Indeed, as depicted in Figure 12, the SMT

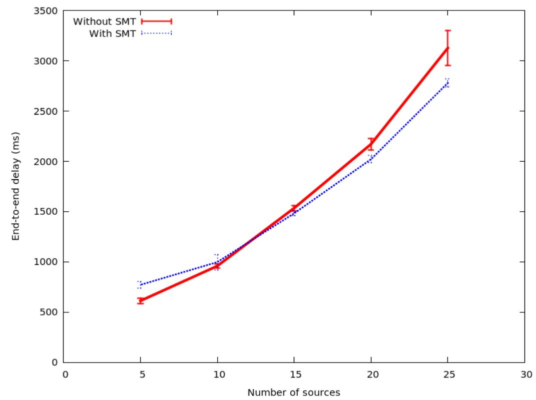


Fig. 11. Impact of SMT mechanism on end-to-end delay

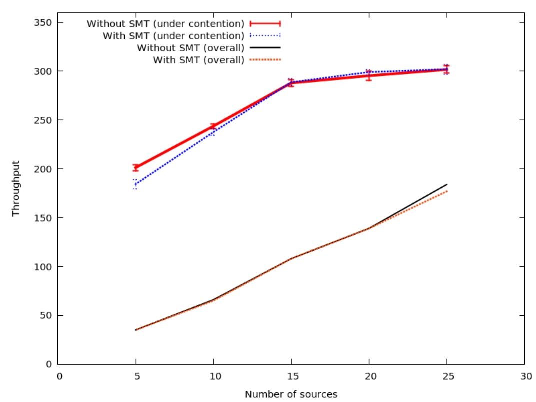


Fig. 12. Impact of SMT mechanism on throughput

mechanism does not have any significative impact on throughput, and can even deteriorate it when considering a limited number of sources.

To conclude SMT unfavorably impacts the global network performances, as it increases the loss-rate with no significative improvement in terms of throughput. In addition, SMT introduces extra overhearing periods which may have significant impact on energy consumption. However, as stated in [3], it helps reduce channel occupation, as it allows up to 6.6% of transmissions to be performed without sending any preamble. It also helps reduce the overall delay by up to 11%. All things considered, we thus do not recommend to implement SMT, as it may deteriorate the loss-rate and throughput in the network with limited gains in term of delay.

6.4 Experimentation conclusions

In the present experimentations, we investigated different solutions to handle the foreseen black spots in X-MAC. We determined which behavior to adopt while $PACK$ is not received, evaluated two variants of preamble emission, and finally quantified the impact of SMT.

In the light of the results obtained, some conclusions can be drawn. First, as anticipated, our campaign confirmed that sending data frames while the preamble was not acknowledged significantly increases packet loss. Most of the time, an unacknowledged preamble is mainly caused by a receiver not ready to receive. We have then shown that interrupting the preamble whenever unexpected information is sampled increases X-MAC performances. Indeed, the preamble interruption reduces overall packet loss rate as well as the preamble length. Finally, we demonstrated that SMT does not meet expectations and even degrades the overall packet loss rate, with no significative impact on access-time.

7 Related Work

The last few years have seen the emergence of real-world deployments of IoT networks. Only few have used the dedicated IEEE 802.15.4 standard, due to its lack of scalability and energy gains. Many protocols were developed to address those problematics, and X-MAC emerged as one of the most popular and efficient solution.

X-MAC was firstly introduced in [3]. This paper provides an exhaustive description of the protocol principles with series of experimentations conducted over a testbed composed of 10 nodes. Although the results provided highlight the benefits of X-MAC compared to the basic LPL mechanism, they only offer a glimpse of X-MAC performances in a real deployment setup.

In [12], an analytical model of X-MAC is proposed and validated through a comparative simulation campaign. This study demonstrates that X-MAC can compete with state-of-the-art synchronized MAC protocols such as S-MAC in terms of throughput, delay and energy consumption. It also points out that X-MAC is less affected by scale changes than synchronized protocols. Preamble sampling protocols have the key advantage that senders and receivers can be completely decoupled in their duty cycles. X-MAC is therefore a serious candidate for managing medium access control in IoT networks.

However, our present evaluation shown that X-MAC can induce high loss-rate (20% with 25 sources). Such observation can also be made in [6] which compares a Contiki OS implementation of X-MAC with the contention-based IEEE 802.15.4 MAC layer. This study, conducted over a 8 node testbed, demonstrates that X-MAC realizes important energy gains but significantly increases delays and loss-rate when compared to MAC layers that do not focus on energy preservation. Furthermore, it appears that most of the existing MAC protocols for IoT networks neglect network performances in favor of energy preservation. The next challenge in the context of wireless sensor and actuator networks is to develop MAC protocols that offer the throughput and reliability necessary for today's and future applications, together with paying attention to the energy consumption.

One interesting way to address this issue is to provide on-the-fly adaptation of MAC parameters. For instance, the pTunes framework [4] monitors network parameters (e.g. delay, loss-rate, energy consumption) in order to find a suitable MAC configuration (i.e. duty-cycle and preamble length values). This framework was experimented in conjunction with X-MAC over a 44 nodes testbed, and helped reduce the average loss-rate below 10%. This gain came at no cost in term of energy consumption.

As overviewed, X-MAC was only experienced over small-scale testbed consisting of several dozens of nodes. No lessons regarding the protocol design were learned empirically. Several studies evaluate X-MAC performances, but none of those underlines flaws in its design, rather avoiding it by interpreting its functioning.

8 Conclusions and Future Work

In this paper, we have performed a complete analysis of X-MAC. We first highlighted several insufficiently described and omitted parts in the design, and detailed their potential repercussions. We then provided some insight and proposed alternatives to obtain the best performances of X-MAC. Thanks to a precise evaluation of X-MAC conducted over a IoT testbed consisting of 240 nodes, we precisely determined which behavior to adopt in order to obtain the best performances.

Three conclusions can be drawn from this thorough empirical analysis. First, sending the data packet while the preamble was not acknowledged is a risky choice as the probability of losing the data packet is very high. Thus, we recommend to consider the preamble as failed whenever it is not acknowledged. This allows the sender to schedule a new transmission of the pending data. Second, we advocate that the preamble emission should be interrupted when managing the perception of noise. Such interruption reduces the network loss-rate and overall preamble length. Finally, we demonstrated that SMT did not perform as well as expected in the X-MAC original design. Indeed, this mechanism reduces channel occupancy at the cost of increased loss-rate in the network.

To the best of our knowledge, this work stands as the first to highlight omissions and black spots in the original X-MAC design and to provide it with significant improvements. As we observed throughout our literature review, it represents the first large-scale experimentation of LPL-based MAC protocols. This way, we were able to provide real understanding on their scalability and performances in a realistic situation. Although X-MAC stands as one of the best solutions for managing MAC in IoT, we demonstrated that it could generate high loss-rate. More generally, most of the existing MAC protocols for IoT networks neglect network performances in favor of energy preservation. Further investigations remain to be led in order to develop MAC protocols that offer the throughput and reliability necessary for current and future applications, together with paying attention to the energy consumption. Considering X-MAC, a dynamic configuration of MAC parameters (i.e. preamble length and duty-cycle) could help offer the exact required performances at all time, without damaging overall energy gains.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] A. Bachir, M. Dohler, T. Watteyne, and K. Leung, "MAC Essentials for Wireless Sensor Networks," *IEEE Communications Surveys Tutorials*, vol. 12, no. 2, pp. 222 –248, 2010.

[3] E. A. M. Buettner, G.V. Yee and R. Han, "X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks," in *Proceedings of the 4th ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, 2006, pp. 307–320.

[4] M. Zimmerling, F. Ferrari, L. Mottola, T. Voigt, and L. Thiele, "pTunes: Runtime Parameter Adaptation for Low-power MAC Protocols," in *Proceedings of the 11th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2012, pp. 173–184.

[5] R. Kuntz, A. Gallais, and T. Noël, "From Versatility to Auto-Adaptation of the Medium Access Control in Wireless Sensor Network," *Elsevier Journal of Parallel and Distributed Computing*, pp. 1236–1248, 2010.

[6] P. Suarez, C. Renmarker, A. Dunkels, and T. Voigt, "Increasing ZigBee Network Lifetime with X-MAC," in *Proceedings of the ACM Workshop on Real-World Wireless Sensor Networks (REALWSN)*, 2008, pp. 26–30.

[7] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental evaluation of wireless simulation assumptions," in *ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, 2004, pp. 78–82.

[8] (2013). [Online]. Available: https://planet.u-strasbg.fr/redmine/projects/xmaccode

[9] IEEE Computer Society, "IEEE Standard 802.15.4-2009: Wireless medium access control and physical layer specifications for low-rate wireless personal area networks," April 2009.

[10] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *IEEE Network*, vol. 18, no. 4, pp. 45–50, 2004.

[11] C. Cano, B. Bellalta, A. Sfairopoulou, and M. Oliver, "Low energy operation in wsns: A survey of preamble sampling mac protocols," *Computer Networks*, vol. 55, no. 15, pp. 3351—3363, 2011.

[12] O. Yang and W. Heinzelman, "Modeling and performance analysis for duty-cycled mac protocols with applications to s-mac and x-mac," in *IEEE Transactions on Mobile Computing (TMC)*, vol. 11, no. 6, 2012, pp. 905–921.

[13] J. Beaudaux, A. Gallais, and T. Noel, "Heterogeneous MAC duty-cycling for energy-efficient Internet of Things deployments," *Springer Networking Science Journal*, 2013.

[14] A. Dunkels, J. Eriksson, N. Finne, F. Osterlind, N. Tsiftes, J. Abeille, and M. Durvy, "Low-power IPv6 for the Internet of Things," in *International Conference on Networked Sensing Systems (INSS)*, 2012.

[15] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "SensorScope: Application-Specific Sensor Network for Environmental Monitoring," *ACM Transactions on Sensor Networks*, vol. 6, pp. 17:1–17:32, 2010.

[16] H. Kdouh, H. Farhat, G. Zaharia, C. Brousseau, G. Grunfelder, and G. Zein, "Performance analysis of a hierarchical shipboard wireless sensor network," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2012, pp. 781–786.

[17] T. Watteyne, K. Pister, D. Barthel, M. Dohler, and I. Auge-Blum, "Implementation of Gradient Routing in Wireless Sensor Networks," in *Proceedings of the IEEE Global Telecommunications Conference (Globecom)*, 2009, pp. 1–6.

[18] C. A. Boano, J. Brown, Z. He, U. Roedig, and T. Voigt, "Low-Power Radio Communication in Industrial Outdoor Deployments: The Impact of Weather Conditions and ATEX-Compliance," in *Sensor Applications, Experimentation, and Logistics*, 2010, vol. 29, pp. 159–176.

[19] T. Liu and A. Cerpa, "Foresee (4C): Wireless link prediction using link features," in *Proceedings of the 10th ACM/IEEE International Symposium on Information Processing in Sensor Networks (IPSN)*, 2011, pp. 294 –305.

[20] ——, "TALENT: Temporal Adaptive Link Estimator with No Training," in *Proceeding of the 10th ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, 2012.