Making Sensornet MAC Protocols Robust Against Interference

Carlo Alberto Boano[1], Thiemo Voigt[2], Nicolas Tsiftes[2], Luca Mottola[2],
Kay Römer[1], and Marco Antonio Zúñiga[3]

[1] Institut für Technische Informatik, Universität zu Lübeck, Lübeck, Germany
[2] Swedish Institute of Computer Science (SICS), Kista, Sweden
[3] Digital Enterprise Research Institute (DERI), Galway, Ireland

Abstract. Radio interference may lead to packet losses, thus negatively affecting the performance of sensornet applications. In this paper, we experimentally assess the impact of external interference on state-of-the-art sensornet MAC protocols. Our experiments illustrate that specific features of existing protocols, e.g., hand-shaking schemes preceding the actual data transmission, play a critical role in this setting. We leverage these results by identifying mechanisms to improve the robustness of existing MAC protocols under interference. These mechanisms include the use of multiple hand-shaking attempts coupled with packet trains and suitable congestion backoff schemes to better tolerate interference. We embed these mechanisms within an existing X-MAC implementation and show that they considerably improve the packet delivery rate while keeping the power consumption at a moderate level.

1 Introduction

The increasing number of wireless devices sharing the same unlicensed ISM bands affects both reliability and robustness of sensornet communications. Sensor networks that operate, for example, in the 2.4 GHz band must compete with the communications of WLAN, Bluetooth, WirelessUSB, and other 802.15.4 devices. They may also suffer the interference caused by appliances such as microwave ovens, video-capture devices, car alarms, or baby monitors. Such problems will increase when more of these devices will be deployed in the near future.

Interference may have a deteriorating effect on communication, as it leads to packet loss and lack of connectivity. This may result in worse performance and reduced energy efficiency of sensornets, causing major issues in a number of application domains, e.g. safety-critical applications in industry and health care.

Studying the impact of interference has been hard because of the lack of proper tools that enable an inexpensive generation of controlled interference. Recently, we demonstrated a method to generate customized and repeatable interference patterns using a common CC2420 radio transceiver in special mode [1]. Using that method, we experimentally study the impact of interference on several MAC protocols, such as Contiki's NULLMAC, X-MAC, LPP, and CoReDac; and TinyOS's LPL. Our goal is to find effective mechanisms that handle interference properly. We carry out our experiments in the 2.4 GHz ISM band, which is also the most crowded one.

In this paper, we investigate which mechanisms improve the robustness of communication in congested networks while remaining reasonably energy efficient. In our experiments we identify three methods that can increase the robustness of sensornet MAC protocols against interference. Since low-power MAC protocols allow nodes to turn off their radio most of the time, they require some kind of *handshaking*. For example, in X-MAC a receiver needs to hear a strobe and answer with a strobe acknowledgment [2]. In Low Power Probing (LPP), the opposite happens: a sender waits for a probe from the intended receiver before it can send the packet [3]. Our experiments show that protocols or parameter settings that enable potentially more handshakes in case some fail due to interference are more robust. Another method that we identify is to use *packet trains* that enable the sender to quickly send multiple packets that have been accumulated during an interference period. The third method is the selection of suitable *congestion backoff schemes* when using Clear Channel Assessment (CCA) and detecting a busy channel. Based on these findings, we include these mechanisms in an X-MAC version, and show its improved robustness to interference.

Our contributions are the following. First, to the best of our knowledge, we are the first to experimentally study how interference affects different MAC protocols. Second, we identify mechanisms that enable MAC protocols to sustain high packet delivery rates while using low-power consumption even in presence of interference. Third, we show experimentally that the choice of congestion backoff schemes is critical for communication performance and energy efficiency in congested networks. Fourth, we augment an existing X-MAC implementation with these mechanisms, and demonstrate substantial performance improvements.

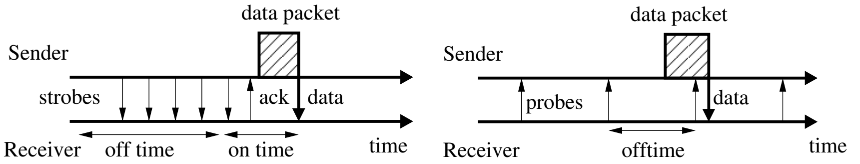
Our paper proceeds as follows. Section 2 provides an overview on the investigated MAC protocols. We describe the methodology and the setup of our experiments in Section 3. Thereafter, in Section 4 and 5, we present our experimental results and identify methods that handle interference properly. In Section 6 we design a new version of X-MAC that implements several of the identified methods and evaluate its performance. We review related work in Section 7 and present our conclusions in Section 8.

2 Background

Medium access control for wireless sensor networks has been a very active research area for the past couple of years, and the literature provides an amazing number of different implementations and incremental improvements. In our work, we exploit the four MAC layers available in Contiki (NULLMAC, X-MAC, LPP, CoReDac) and Tiny OS' LPL. Section 2.1 briefly describes these protocols, and Section 2.2 explains the role of CCA in sensornet MAC protocols.

2.1 Overview of used MAC protocols

NULLMAC. NULLMAC is a minimalistic MAC protocol that simply forwards traffic between the network layer and the radio driver. As such, it does not provide any power-saving mechanism, and keeps the radio always on. This allows



(a) X-MAC (b) LPP

Fig. 1. In X-MAC (left), the sender strobes until the receiver is awake and can receive a packet. In LPP (right), the receivers send probes to announce they are awake and ready to receive packets.

for the maximum throughput achievable, while consuming the highest amount of energy. When used with CCA and back-off timers, NULLMAC behaves as a traditional CSMA-CA protocol. Because of these characteristics, we use NULLMAC as a baseline to compare the performance of other protocols, and to verify the correctness of our setup.

X-MAC. X-MAC is a power-saving MAC protocol [2] in which senders use a sequence of short preambles (strobes) to wake up receivers. Nodes turn off the radio for most of the time to reduce idle listening. They wake up shortly at regular intervals to listen for strobes. When a receiving node wakes up and receives a strobe destined to it, it replies with an acknowledgment indicating that it is awake. After receiving the ACK, the sender transmits the data packet, as shown in in Figure 1(a).

The X-MAC implementation in Contiki has several parameters of significance to our experiments. *Ontime* determines the maximum time that a receiver listens for strobes, whereas *offtime* specifies the time to sleep between waking up to listen for strobes. *Strobe_time* denotes the duration a sender transmits strobes until it receives a strobe acknowledgment from the receiver. In the default Contiki X-MAC implementation, strobe_time = offtime + (20 × ontime).

Low-Power Probing (LPP). LPP is a power-saving MAC protocol where receivers periodically send small packets, so called probes, to announce that they are awake and ready to receive a data packet [3]. After sending a probe, the receiver keeps its radio on for a short time to listen for data packets. A node willing to send a packet turns on its radio waiting for a probe from a neighbor it wants to send to. On the reception of a probe from a potential receiver, the node sends an acknowledgment before the data packet, as shown in Figure 1(b).

The LPP implementation in Contiki contains two important parameters. *Ontime* determines how long a receiver keeps the radio on after the transmission of a probe, *offtime* is the time between probes. We use $\frac{1}{2}$ and $\frac{1}{64}$ seconds for *offtime* and *ontime* respectively. Another parameter is the time to keep an unsent packet: Contiki LPP's default value is 4×(ontime + offtime). If LPP receives a packet from the network layer when the packet queue is full, LPP discards the new packet. The queue length is configurable, and the default size is 8 packets.

Low-Power Listening (LPL). We consider a Low-Power Listening (LPL) layer that implements an asynchronous wake-up scheme for CC2420 radios [4]. Nodes periodically wake up to detect transmissions. To do so, they rely on CCA rather than attempting to pick up a full packet. Unlike X-MAC, senders repeatedly transmit the entire packet for twice the duration of the wake-up period. In case of unicast transmissions, the intended receiver may acknowledge the transmission to notify the sender on correct packet delivery so that the sender can stop transmitting earlier. To implement this functionality, packet transmissions are interleaved with periods of silence in order to allow ACK transmissions. The only LPL parameter tunable by the users is the wake-up period.

CoReDac. CoReDac is a TDMA-based convergecast protocol [5] that builds a collection tree that guarantees collision-free radio traffic. From D-MAC [6] CoReDac borrows the idea of staggered communication. To avoid collisions among packets from their children, CoReDac parents split their reception slots into subslots, and assign one to each child. Packet acknowledgments are pivotal in CoReDac because they piggyback the assignment information, and they are used for synchronizing the TDMA-schedules. A node that misses an acknowledgment must keep its radio on until it hears a new one.

2.2 Clear Channel Assessment

Clear Channel Assessment (CCA) is a mechanism used to determine if a wireless channel is currently free. In wireless MAC protocols, CCA is used to implement Carrier Sense Multiple Access: each node first listens to the medium to detect ongoing transmissions, and transmits the packet(s) only if the channel is free, thus reducing the chance of collisions. CCA is typically implemented by comparing the Received Signal Strength (RSS) obtained from the radio against a threshold. The channel is assumed to be clear if the RSS does not exceed the given threshold. As false negatives result in collisions and false positives cause increased latency, the choice of the threshold is critical [7]. When using CCA to perform CSMA, backoff schemes play an important role. There are two types of backoff: congestion backoff and contention backoff. The former controls the waiting time between consecutive assessments if the channel is not clear. The second controls the waiting time before a retransmission after a collision is detected.

3 Methodology

In our experiments, we use a set of MAC protocols from both the Contiki and TinyOS operating systems. To set a protocol's parameters, we look at the configurations used in popular, low-rate data collection applications [8, 9] that employed similar MAC protocols. These parameters are in general not set to perform optimally under interference.

3.1 Generating Controllable Interference

In our experiments we use a method proposed by Boano et al. [1] to generate customized, controllable, and repeatable interference patterns using common sensornet devices. This method enables the generation of precisely adjustable levels of interference on a specific channel, by exploiting the special test modes of the radio chip.

3.2 Performance Measurements

We use Contiki's software-based power profiler [10] to measure power consumption. For the experiments concerning TinyOS, we have implemented the same mechanism in TinyOS. For computing the power consumption, we assume a current of 20 mA for the radio in receive mode, and a voltage of 3 V, as measured by Dunkels et al. [10]. In all our experiments, the power consumed by the radio in receive mode (*RX power*) is much higher than the one used for transmitting (*TX power*). Because of its strobe mechanism, X-MAC has the highest TX power among the MAC protocols that we examine. At 60% interference, the TX power is around 1 mW, whereas the RX power is almost 20 mW. For LPP instead, the TX power is usually between 0.1 and 0.2 mW only. The power values represent the average power during the full experiment. Since the RX power is at least an order of magnitude larger than the TX power in our experiments, we display only the RX power in our graphs.

3.3 Experimental Setup and Interference Model

In our experiments we put three nodes near each other: a sender, a receiver, and an interferer. The latter interferes using the CC2420's maximum output power level (31), while the sender and the receiver use TX power level 7. The placement of the nodes and their power levels ensure that an active interferer blocks any ongoing communication between the sender and the receiver.

Interference may result from other packet radios (Wi-Fi, Bluetooth, and other sensor networks) operating in the same frequency band, and from other electromagnetic sources such as motors or microwave ovens. Unfortunately, at the time of writing, there are no accepted interference models – an important research issue by itself that is beyond the scope of this paper. Hence, we resort to two simple models here. The *bursty interferer* models continuous blocks of interference with uniformly distributed duration and spacing. This type of interference may be caused, for example, by Wi-Fi or Bluetooth transmissions. The *semi-periodic interferer* also models continuous blocks of interference, but the duration of the periods and their spacing have smaller variance. This type of interference may be caused, for example, by a sensornet performing periodic data collection.

Bursty Interference. In order to describe the transmission and interference patterns, let us define the following random variables:

- S: Bernoulli random variable with parameter 0.5;
- R: Uniformly distributed over [0, 100];
- $Q(x)$: Uniformly distributed over [0, x].

Interference follows continuous off/on periods, and is dictated by a simple two-state discrete Markov process, as depicted in Figure 2. C denotes the clear

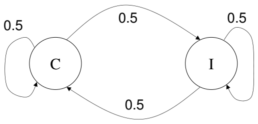


Fig. 2. The interference model used in our experiments.

channel state, and I denotes the interference state. The transitions between the two states is specified by S. At each step of the Markov process, we obtain a time period, $R \times Q(x)$, that determines the duration of the next state. For example, assuming that we move to state I and that we obtain values $R = 40$ and $Q = 20$, the next period will be an interference period of length $40 \times 20 \times 0.3$ ms=240 ms (0.3 ms is a constant factor). $Q(x)$ is used to scale the burstiness of the interference. A higher value represents longer interference slots, such as the ones caused by bursts of Bluetooth or Wi-Fi traffic, whereas a lower value represents shorter transmissions. In the experiments we will select a configuration with long interference slots ($x = 50$) that we call *long bursts*, and a configuration with shorter slots ($x = 8$) that we call *short bursts*.

Semi-Periodic Interference. The semi-periodic interferer is a 2-stage process. As described above, we have a clear channel C and an interference I states. The process stays in state I for a time that is uniformly distributed between $\frac{9}{16}$ seconds and $\frac{15}{16}$ seconds. After the transition to state C, it stays in this state for a time that is uniformly distributed between $\frac{3}{4} \times \text{clear\_time}$ and $\frac{5}{4} \times \text{clear\_time}$, where *clear_time* is a parameter that determines the rate of interference.

4 Experimental Evaluation: the Performance of MAC Protocols under Interference

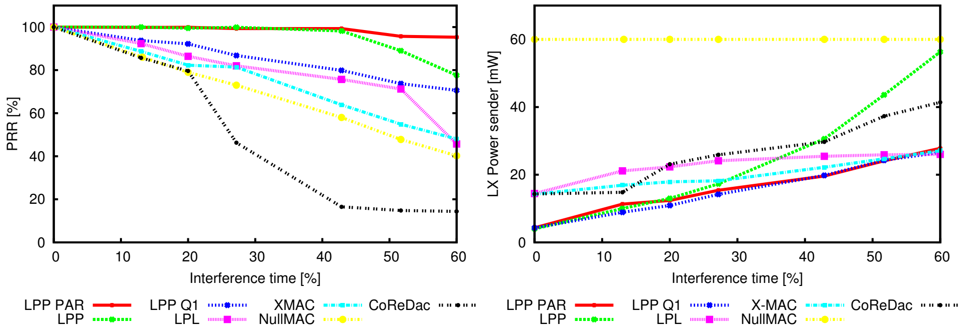
In this section we report on the performance of several MAC protocols under the different interference patterns described in the previous section.

4.1 Semi-periodic Interference

In our experiments, the sender transmits unicast packets with a payload of 22 bytes to the receiver in a time uniformly distributed between 0.75 s and 1.25 s. We collect the measurements until several thousands packets have been transmitted. We use a semi-periodic interference pattern as described in Section 3.

Figure 3 shows the results of our experiments with different MAC protocols tested against varying interference rates. As expected, the PRR in NULLMAC decreases linearly with the interference rate, following the rule 100% minus the interference rate, which is the probability that a packet is not interfered (Figure 3(a)). The RX power consumption when using NULLMAC is 60 mW independently on the interference pattern, since NULLMAC keeps the radio always on (Figure 3(b)). This confirms the validity of our setup, described in Section 3.

Figure 3(a) shows that all variants of LPP have fairly high packet reception rates compared to the other protocols we consider. Among LPP-based solutions,



(a) Packet Reception Rate (b) RX Power Consumption (sender side)

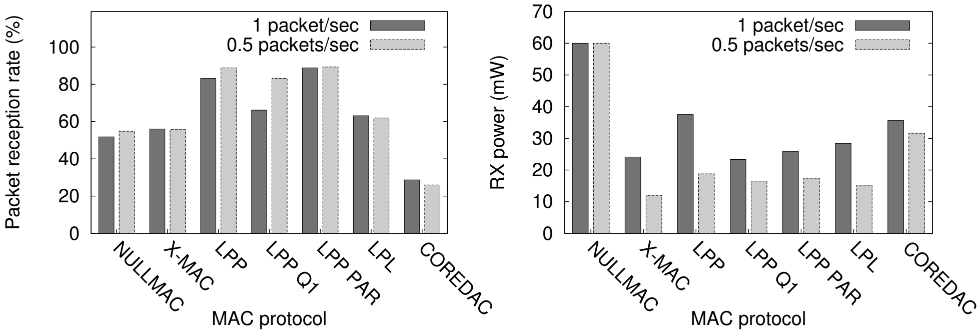
Fig. 3. MAC protocols performance under semi-periodic interference.

the best performance is obtained with LPP-PAR, where the receiver transmits a new probe immediately after a packet reception. By doing so, the sender can drain its queue when the interference clears and sustain a high PRR also under high interference by deferring transmissions until interference is over. LPP-PAR outperforms both the standard LPP version, and the so called LPP-Q1, that does not have a queue: a new packet from the upper is discarded in case the previous one has not been transmitted by the MAC layer. At an interference rate of 42%, LPP-Q1 still achieves a PRR of about 80%, showing that even only two probe attempts provide more opportunities to deliver a packet than other solutions.

Figure 3(b) shows that the power consumption of LPP-Q1 is lower than the standard LPP one. The reason comes from the lower PRR shown by LPP-Q1: with fewer packets to be transmitted, the radio is turned off more often. This difference becomes very apparent at an interference rate of 60%, where LPP has its radio turned on almost all the time since there is almost always a packet in the queue waiting to be transmitted. In contrast with the default LPP, LPP-PAR can quickly drain its queue during interference-free periods and hence turn off quickly its radio, saving a substantial amount of power.

X-MAC's packet reception rate is similar but slightly higher than NULL-MAC's (Figure 3(a)), since in X-MAC the sender's strobe_time is a little longer than the receiver's off_time. Hence, the receiver has in average more than one chance to hear a strobe. Furthermore, under a semi-periodic interference pattern, it is unlikely that interference comes into effect during the exchange of strobe, acknowledgment, and data packet, which take very little time. Therefore, if the strobe succeeds, the entire operation most likely successfully completes. The same reasoning also applies for CoReDac when the interference rate is 20% or lower. At higher interference, however, CoReDac looses synchronization and its performance drastically degrades.

With regard to LPL, we observe two modes of operations along the PRR axis in Figure 3(a). When the interference rate is lower than 60%, the CCA



(a) Packet Reception Rate (b) RX Power Consumption (sender side)

Fig. 4. MAC protocols performance under bursty interference.

mechanism is reasonably effective at detecting the presence of interference, and packet losses occur mostly because of data corruption during the transmission. Indeed, we verify that an increasing number of packets are received but do not pass the integrity checks. The increasing power consumption shown for LPL in Figure 3(b) is simply an effect of the decreasing PRR: the fewer packets are received, the less likely is the sender to receive the acknowledgment and stop the transmissions earlier. On the other hand, at 60% interference it is often the case that the CCA mechanism never finds the channel free. After a maximum number of reattempts, the packet is dropped on the sender side, causing a drastic decrease in PRR. However, without even transmitting the packet, not much energy is spent on the sender side. This is confirmed in Figure 3(b), where the power consumption at 60% interference is still comparable to other settings.

Our results suggest that more handshakes opportunities improve the PRR in interfered networks. When comparing different LPP versions with each other, we can see that we can achieve a low power consumption and a high PRR using LPP-PAR, thanks to its queue drain when a period of interference has ended.

Impact of Queue Size on Performance. Our experiments clearly show that the queue size may drastically change the performance of a MAC protocol under interference. We investigated the impact of the queue size both on power consumption and packet reception rate by running LPP with different queue sizes under 60% semi-periodic interference. Our results show that a queue size of four packets guarantees good performance.

4.2 Bursty Interference

We carry out the same set of experiments in presence of bursty interference ($x = 50$, see Section 3), and different transmission rates, in order to investigate how performance changes depending on the network load. Figure 4 illustrates the results.

For most MAC protocols the PRR does not change depending on the transmission rate (Figure 4(a)). In most cases, indeed, the interference rate is what ultimately determines the observed PRR. An exception is LPP-Q1, where the PRR increases by almost 10% when the application transmits packets less frequently. The reason is that with higher transmission rates, a packet cannot be sent before the application hands the next packet to the MAC layer, and thus the latter packet is discarded. This can either happen with long periods of interference, or when periods of interference overlap with the instants in which the receiver sends probes.

5 The Impact of Clear Channel Assessment and Congestion Backoff under Interference

While many contention-based MAC protocols implement CSMA, one could also start transmitting a packet without carrying out CCA. The latter approach saves the CCA overhead of listening to the channel and switching the radio between send and receive modes, which may take hundreds of microseconds [11]. Few retransmissions consume a negligible amount of power compared to a continuous use of CCA. An increased probability of collisions may be negligible in low data rate applications, but not in settings with high interference.

A second aspect that affects the performance of CSMA-based MAC protocols such as B-MAC [12], WiseMAC [13], and BoX-MAC [14] is the backoff algorithm that adapts the scheduling of CCA executions to wireless channel conditions. B-MAC, for example, uses by default a small random congestion and contention backoff time, but does also support user-defined backoff schemes. BoX-MAC uses a randomized long congestion backoff period in the order of a few hundred milliseconds.

In this section we identify (1) the scenarios where adopting CCA improves or decreases the performance of MAC protocols under interference, and (2) if the choice of the congestion backoff scheme plays a pivotal role under interference. We investigate these issues in terms of energy efficiency and latency.

5.1 Experimental Setup

In our first experiment, we compare a scenario in which CCA is not used (and packets are sent without a carrier sense) with one in which a node sleeps after detecting a busy channel for a congestion backoff time B_C. We explore different types of backoff algorithms, in particular null (no waiting time), constant (waiting time uniformly drawn from a fixed backoff window), linear (backoff window increases by a constant amount after failed CCA), quadratic (backoff window squared after failed CCA), and cubic (backoff window cubed after failed CCA).

We select an initial backoff time randomly short and we eventually increase it according to the backoff algorithm. We further study a variant where the backoff is truncated after $R = 8$ CCA attempts. We use the CC2420's default CCA threshold.

Our experimental setup is described in Section 3. The transmitter sends N packets towards the receiver at different transmission rates. Each packet has to be acknowledged within $\frac{1}{64}$ seconds.

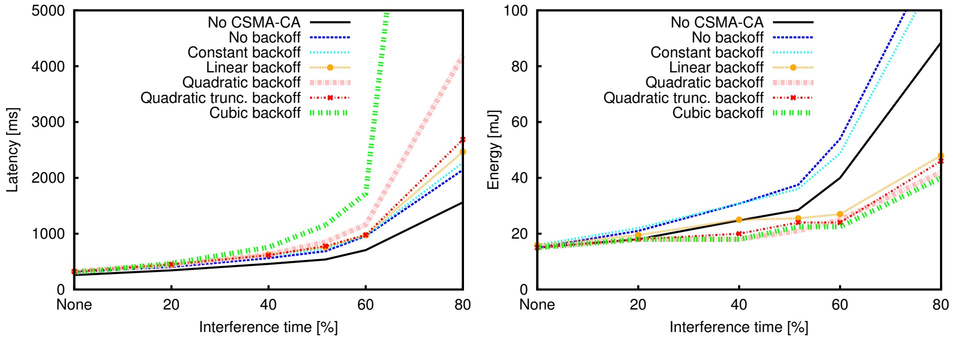
We further investigate two different strategies for scheduling retransmissions. With the first approach, queued packets are retransmitted immediately after timeout. With the second approach, the sender turns off its radio after a timeout occurs, and the queued packets are retransmitted according to the original packet transmission rate (e.g. after 0.5 seconds if we transmit 2 packets per second). We measure the latency required to transmit the sequence of N packets and the total amount of energy consumed by the radio of the sender. The latter is appropriate because interference mainly affects the sender, assuming that a receiver can distinguish valid data from interference and go back to sleep in case of the latter. The sender node runs NULLMAC with or without CSMA, and its radio is turned off after the reception of an ACK (or after the timeout fires), and turned on again for the next transmission. Since we are only interested in the energy consumption of the sender, the receiver keeps the radio on all the time. To isolate the effect of CCA from that of other MAC mechanisms, we avoid mechanisms such as LPL and the associated use of long preambles.

5.2 Experimental Results

In the first set of experiments, we evaluate the communication performance when transmitting $N = 50$ packets at the highest available rate, and compare transmissions with and without CSMA. We average the results after sending several thousand packets. Figure 5 shows the results. As expected, the more aggressive the backoff strategy is, the lower is the energy required to complete the transmission. The latency increases proportionally with the backoff delays, however, indicating a tradeoff between energy consumption and latency. The energy consumption is, however, significantly reduced when not using CSMA, but using aggressive backoffs such as quadratic and cubic algorithms on a channel that is interfered more than 20% of the time. We can also see that truncating the backoff window yields a good balance between energy and latency.

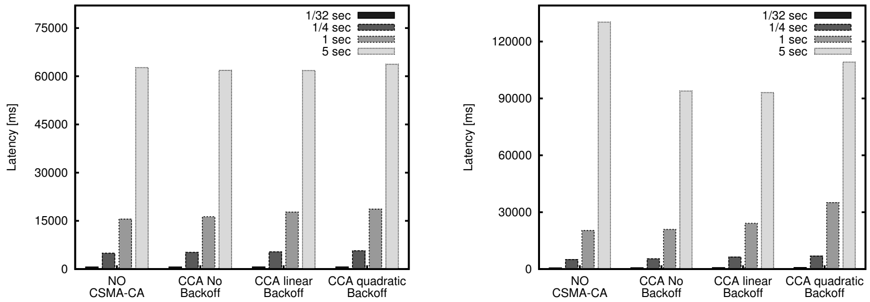
In the scenario presented above, the packets are retransmitted as soon as the timeout event occurs. If queued packets are retransmitted back-to-back under interference, there is a significant waste of energy due to the medium still being busy, while a retransmission based on the original transmission rate increases the overall latency. To quantify these issues, we carry out another experiment with different periodic transmission rates. We transmit bursts of $N = 10$ packets with and without CSMA, using null, linear, and quadratic congestion backoff schemes. Then we apply a bursty interference pattern with long bursts ($x = 50$) and measure the latency and energy consumption at the sender side, averaging the results of several hundred bursts.

Figures 6 and 7 show the results. As expected, if queued packets are retransmitted back-to-back, the approach without CSMA performs poorly. A configuration with quadratic congestion backoff requires only 5% of the energy used without CSMA with an acceptable latency because of the fewer attempts. If,



(a) Latency (b) Energy consumption

Fig. 5. Energy consumption and latency measured at the sender side, when sending bursts of $N = 50$ packets at the highest rate available.

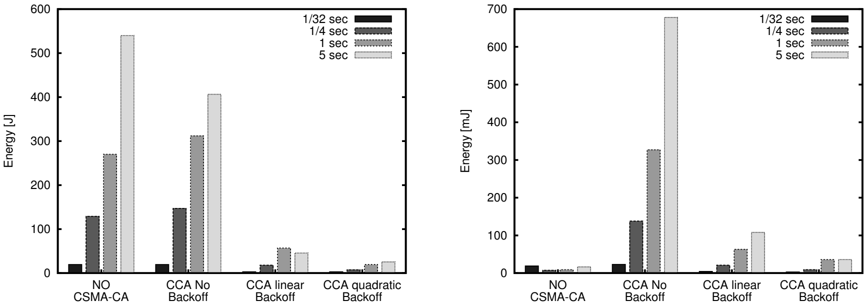


(a) Immediate retransmission (b) Delayed retransmission

Fig. 6. Latency measured at the sender side when sending bursts of $N = 10$ packets at different transmission rates, with different retransmissions schemes.

instead, queued packets are retransmitted according to the original transmission rate, the protocol that does not adopt CSMA performs better in terms of energy efficiency. This is because it attempts to transmit only at the instants defined by the transmission rate, while the approach with CSMA and backoff tries to find the first instant at which the medium is free, often without success. This makes the approach without CSMA more energy-efficient, but comes with an increased latency when sending at low transmission rates, such as one packet every 5 seconds w.r.t. CSMA transmissions. As in the previous experiment, a more aggressive congestion backoff scheme such as the quadratic algorithm shows a good balance between latency and energy consumption.

In addition to the above experiments with long bursts, we also carried out experiments with shorter bursts ($x = 8$, see Section 3). Due to space constraints we do not show the results here. These experiments indicate a better performance of protocols using CSMA, because shorter slots will imply a lower energy consumption since the channel will be sampled a smaller amount of times.



(a) Immediate retransmission (b) Delayed retransmission

Fig. 7. Energy consumption measured at the sender side, when sending bursts of $N = 10$ packets at different transmission rates, with different retransmissions schemes.

In conclusion, our experiments demonstrate that the choice of congestion backoff scheme plays a pivotal role for MAC protocols that use CCA. These results act as a guideline for protocol designers. A CSMA approach with a quadratic backoff –truncated or not– performs well in most scenarios.

6 Improvements

The results presented in Section 4 show two methods that can make MAC protocols more robust against interference: (1) holding a packet longer so that multiple handshake attempts are possible, and (2) implementing packet trains as a means to quickly send multiple packets that have accumulated during interference. Section 5 further shows that the power consumption can be reduced by applying suitable congestion backoff schemes when using CCA. We extend the X-MAC implementation in Contiki 2.3 with these mechanisms, and evaluate it under random interference patterns.

6.1 Design and Implementation of a Robust X-MAC

We design a new version of X-MAC, called X-MAC/Q, that is able to maintain high packet reception rates and low power consumption despite being challenged by interference. The new version contains a packet queue implemented by using a statically allocated array of packets and their corresponding attributes. By default, the queue stores up to four packets, the optimal value for LPP as discussed in Section 4.1. Since only unicast packets are acknowledged in the X-MAC protocol implementation, we only queue unicast packets.

Packet Queue with Fast Drain. Unlike the original implementation of X-MAC in Contiki, our augmented implementation revolves around the packet queue. This distinction starts from the existing packet transmission method, *qsend_packet()*, where all unicast packets are put into a queue. The packets will not be sent directly, but instead linger shortly for a configurable time ($\frac{1}{32}$ s

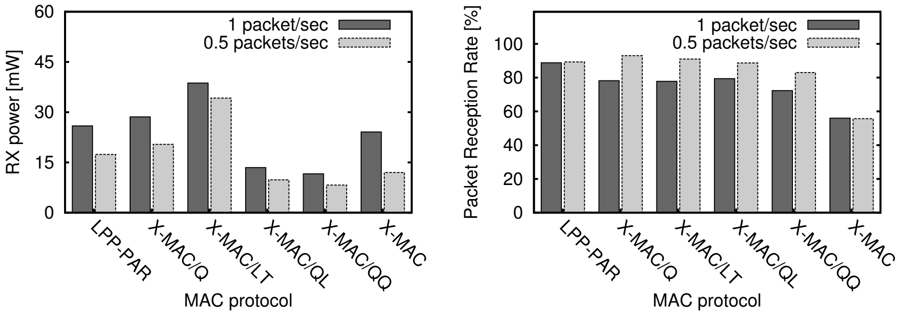


Fig. 8. Our experiments show that the proposed mechanisms increase the robustness of X-MAC to interference.

in our experiments.) The linger time makes it possible to accumulate packets into the queue, which allows the layer on top of X-MAC to create a burst of packets. When the accumulation timer has expired, X-MAC/Q gets the oldest packet from the queue, and immediately starts sending strobes to the addressed receiver of the packet. To enable fast queue draining, each strobe contains the amount of packets for the destination that the sender has in its queue. If the sender receives a strobe acknowledgment within a configured waiting time, it sends one packet at a time, including the strobe procedure, separated by a very short time ($\frac{1}{128}$ s) instead of the usual duty-cycle interval. If the sender does not receive the strobe acknowledgment, a new attempt comes after $\frac{1}{32}$ s. Packets are removed from the queue when they have either been successfully sent, or timed out after 10 s. The X-MAC reception method requires only two changes. First, each received strobe will contain the amount of packets x that the receiver should receive in a train. Second, the receiver stays awake until it has received x packets since the strobe.

Clear Channel Assessment with Congestion Backoff. Based on the results in Section 5, we extend X-MAC/Q to include clear channel assessments with a linear and a quadratic congestion backoff timers. The version with the linear backoff is called X-MAC/QL, whereas the version with quadratic backoff is called X-MAC/QQ. Before sending out the first strobe the new versions turn on the CCA to check if the channel is clear. If the CCA check fails, we wait for ($\frac{1}{128} \times$ number_of_attempts) or ($\frac{1}{128} \times$number_of_attempts^2) milliseconds before another attempt for X-MAC/QL and X-MAC/QQ respectively.

6.2 Experimental Evaluation

We repeat the experiments with the bursty interferer described in Section 4.2 using our improved versions of X-MAC. For comparison, we also show the LPP-PAR and another X-MAC improvement that we call X-MAC/LT. X-MAC/LT is similar to X-MAC except for one parameter, *strobe_time*, which we increase from offtime + 20 × ontime to 4 × offtime + 20 × ontime. Because X-MAC/LT holds packets longer, we expect a higher PRR compared to X-MAC.

Figure 8 shows that both X-MAC/Q and X-MAC/LT significantly increase the PRR compared to the default X-MAC. When the applications send one packet every two seconds, the PRR is similar to the one of LPP-PAR. Also, both new X-MAC versions show a similar rate, but the left graph in Figure 8 shows that the power consumption is much higher for X-MAC/LT than for X-MAC/Q. X-MAC/QQ and X-MAC/QL achieve a good PRR with very low power consumption. Since both protocols wait for an increasing amount of time when the medium is kept busy, they send less strobes and avoid to wait for strobe acknowledgments that will not arrive, thus saving a significant amount of power. Compared to X-MAC/QQ, X-MAC/QL consumes slightly more energy but achieves a higher PRR. This follows the results presented in Section 5.2: the linear backoff causes more frequent samples of the channel than the quadratic one does, leading to higher power consumption. On the other hand, the quadratic algorithm may grow its sampling interval exponentially up to a point where expired packets will be removed from the queue.

In all our experiments, we set the protocol parameters based on the configurations of similar MAC protocols in popular applications [8, 9], since our goal is not to optimize parameters but to identify mechanisms that enable good performance during interference. One way of increasing the handshake frequency would be to change the parameters. In X-MAC, this is the *offtime* parameter. We have rerun the same experiment as in Figure 8, but halved the *offtime* to 1/4 s for X-MAC and X-MAC/Q. Our results show similar improvements in PRR and power consumption for both protocols. For the CCA versions with a linear backoff, the improvements of the PRR were smaller but the power consumption was decreased by around 40%.

In summary, our results show significant improvements of the packet reception rate for X-MAC/Q with a moderate increase in power consumption. X-MAC/QQ and X-MAC/QL's power consumption is even lower than X-MAC's despite that they achieve a much higher PRR.

7 Related Work

Radio interference has been a topic of significant interest in the sensor network community. Most of the earlier work focused on deriving fair transmission schedules by synchronizing the transmission of neighboring nodes in the presence of interference [15–18]. Our work also addresses MAC performance, but our goal is to identify experimentally some mechanisms that improve the robustness of MAC protocols against interference.

Zhou et al. present some important differences between the interference behavior of real and ideal scenarios [19, 20]. Others study interference effects on real deployments: Rangwala et al. propose an interference-aware fair-rate control evaluated on real hardware [21]. Others have proposed frequency hopping solutions for 802.15.4 networks in order to overcome Wi-Fi interference [22, 23].

Motivated by the empirical works mentioned above, we (1) analyze experimentally the impact of interference on various MAC protocols, and (2) propose mechanisms to increase packet delivery rate and reduce energy consumption.

An important group of work pertaining to this study is the set of notable MAC protocols evaluated on empirical testbeds, in particular X-MAC[2], LPP[3], LPL[12]. Most of these evaluations focused on energy efficiency and delay under different traffic patterns while we evaluate the protocols behaviour under various degrees of interference. Bertocco et al. investigate efficient CCA thresholds in presence of in-channel wide-band additive white Gaussian noise [7]. In this work, we study the role of CCA and congestion backoff schemes with respect to energy consumption and latency under generic patterns of interference. So far, thorough studies on backoff schemes have been performed only with respect to contention resolution [24], [25], and [26], where Jamieson et al. propose a MAC protocol that uses a fixed-size contention window and a non-uniform probability distribution of transmitting in each slot within the window.

Moss and Levis envisioned how a long congestion backoff could at the same time optimize energy and delivery rates in congested networks [14]. However, they do not determine optimal backoff periods and do not quantify the effects of different schemes. We demonstrate experimentally the impact of the congestion backoff time on energy efficiency and latency in networks with high interference.

8 Conclusions

In this paper, we experimentally study the impact of interference on several MAC protocols. Using the results from our experiments, we identify mechanisms that make MAC protocols more robust against interference. We augment an existing X-MAC implementation with these mechanisms, and demonstrate improved packet reception rates and reduced power consumption in cases where the radio communication is challenged by interference.

Acknowledgments

This work has been partially supported by the European Commission under the contract No. FP7-2007-2-224053 (CONET). This research has been also partially financed by VINNOVA, the Swedish Agency for Innovation Systems, by SSF, and by the IRCSET Postdoctoral fellow Grant PD200857, SFI Grant No. SFI08CEI1380. This work has been partially supported by the Cluster of Excellence 306/1 ”Inflammation at Interfaces” (Excellence Initiative, Germany).

References

1. C.A. Boano, Z. He, Y. Li, T. Voigt, M. Zuniga, and A. Willig. Controllable Radio Interference for Experimental and Testing Purposes in Wireless Sensor Networks. In *Proc. of the 4th Workshop on Practical Issues in Building Sensor Network Applications (SenseApp)*, Zurich, Switzerland, October 2009. IEEE Computer Society.
2. M. Buettner, V. Yee, E. Anderson, and R. Han. X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In *ACM SenSys*, 2006.
3. R. Musaloiu-E., C-J. M. Liang, and A. Terzis. Koala: Ultra-Low Power Data Retrieval in Wireless Sensor Networks. In *IPSN '08*, 2008.

4. TinyOS Community Forum. TinyOS TEP 126 - CC2420 radio stack. www.tinyos.net/tinyos-2.x/doc/html/tep126.html.
5. T. Voigt and F. Österlind. CoReDac: Collision-free command-response data collection. In *13th IEEE Conference on Emerging Technologies and Factory Automation*, Hamburg, Germany, September 2008.
6. G. Lu, B. Krishnamachari, and C. Raghavendra. An adaptive energy-efficient and low-latency mac for data gathering in wireless sensor networks. In *International Parallel and Distributed Processing Symposium (IPDPS)*, 2004.
7. M. Bertocco, G. Gamba, and A. Sona. Experimental Optimization of CCA Thresholds in Wireless Sensor Networks in the Presence of Interference. In *Proc. of IEEE Workshop on ElectroMagnetic Compatibility (IEEE EMC)*, June 2007.
8. A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In *First ACM Workshop on Wireless Sensor Networks and Applications (WSNA 2002)*, September 2002.
9. G. Tolle et al. A macroscope in the redwoods. In *SenSys*, pages 51–63, 2005.
10. A. Dunkels, F. Österlind, N. Tsiftes, and Z. He. Software-based on-line energy estimation for sensor nodes. In *Proceedings of the 4th Workshop on Embedded Networked Sensors (EmNetS)*, Cork, Ireland, June 2007.
11. K. K. Chintalapudi and L. Venkatraman. On the design of mac protocols for low-latency hard real-time discrete control applications over 802.15.4 hardware. In *the 7th Conf. on Information Processing in Sensor Networks (IPSN)*, April 2008.
12. J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *ACM SenSys*, 2004.
13. A. El-Hoiydi and J.D. Decotignie. Wisemac: An ultra low power mac protocol for the downlink of infrastructure wireless sensor networks. In *ISCC*, June 2004.
14. D. Moss and P. Levis. Box-macs: Exploiting physical and link layer boundaries in low-power networking. Technical Report SING-08-00, Stanford University, 2002.
15. L. Tassiulas and S. Sarkar. Maxmin fair scheduling in wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 23(1):163–173, 2005.
16. L. Chen, SH Low, and JC Doyle. Joint congestion control and media access control design for ad hoc wireless networks. In *INFOCOM*, 2005.
17. Y. Yi and S. Shakkottai. Hop-by-hop congestion control over a wireless multi-hop network. *IEEE/ACM Transactions On Networking*, 15(1):133–144, 2007.
18. Y. Yi, G. de Veciana, and S. Shakkottai. On optimal MAC scheduling with physical interference. In *INFOCOM*, 2007.
19. G. Zhou, T. He, JA Stankovic, and T. Abdelzaher. RID: Radio interference detection in wireless sensor networks. In *INFOCOM*, 2005.
20. G. Zhou et al. Models and solutions for radio irregularity in wireless sensor networks. *ACM Trans. Sen. Netw.*, 2(2):221–262, 2006.
21. S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis. Interference-aware fair rate control in wireless sensor networks. In *ACM SIGCOMM*, 2006.
22. R. Musaloiu-E. and A. Terzis. Minimising the effect of wifi interference in 802.15.4 wireless sensor networks. *Int. Journal of Sensor Networks*, 3:43–54, Dec. 2007.
23. J. Hauer, V. Handziski, and A. Wolisz. Experimental study of the impact of wlan interference on ieee 802.15.4 body area networks. In *EWSN*, February 2009.
24. Z. Yuan et al. A backoff copying scheme for contention resolution in wireless sensor networks. In *Wintech*, September 2009.
25. A. Athanasopoulos et al. 802.15.4: The effect of different back-off schemes on power and qos characteristics. *IEEE ICWMC*, 2007.
26. K. Jamieson, H. Balakrishnan, and Y.C. Tay. Sift: a mac protocol for event-driven wireless sensor networks. In *EWSN*, February 2006.