A Reliable MAC for Delay-Bounded and Energy-Efficient WSNs

Philip Parsch and Alejandro Masrur
Department of Computer Science
TU Chemnitz, Germany

***Abstract*—With the advent of Internet of Things (IoT), an increasing number of devices start exchanging information. This puts emphasis on wireless sensor networks (WSNs) to facilitate the interaction with the environment in varied application scenarios such as, for example, building and home automation among others. In this context, a reliable communication is usually required, i.e., it is necessary to guarantee that packets arrive within a specified maximum delay or deadline. In addition, since battery-driven nodes are used and/or for the sake of sustainability, WSNs often have to economize on energy. However, most existing MAC (Medium Access Control) protocols are either unable to provide guarantees on reliability (e.g., CSMA) or they incur too much energy consumption (e.g., TDMA). To overcome this problem, we propose a MAC technique that guarantees reliability requirements while allowing for delay-bounded and energy-efficient communication. We carry out a large number of experiments based on detailed simulations with OMNeT++ comparing the proposed MAC, in particular, with CSMA and TDMA and illustrating its benefits.**

I. Introduction

Wireless sensor networks (WSNs) facilitate the interaction with the environment and are regaining in importance as Internet of Things (IoT) applications advance. Requirements on the underlying medium access control (MAC) layer are manifold and strongly depend on the context. In general, reliability and energy efficiency are of major importance, since nodes are typically battery powered and many applications require data packets to be successfully delivered, often, within an upper time bound or deadline.

To this end, a number of MAC protocols have been presented to increase energy efficiency and reliability in WSNs. Many of them are based on well-known paradigms, such as CSMA or TDMA, and add further functionality to improve the overall performance. These are, for example, mechanisms that switch from CSMA to TDMA under high contention [16] or hybrid schemes with both TDMA and CSMA slots [13]. However, despite increasing the average performance, they also add more complexity and, in the end, reduce to either the performance of CSMA or TDMA in the worst case.

Although many protocols — including the ones mentioned above — are general purpose, i.e., they are applicable to a wide range of applications, they do not always offer good performance for specific applications. This is because they often miss the ability to adapt their parameters, such as retransmission numbers or back-off times, to a given network setting or environment. Using their typically fixed or only slightly variable parameters can lead to low performance in many different settings. On the other hand, other MAC protocols that can be adapted are often complex and use a best effort approach, meaning they cannot provide any guarantees on performance.

978-1-5386-1898-1/17/$31.00

In this paper, we propose a MAC protocol that overcomes inflexibility by existing protocols, while still maintaining low complexity and a guaranteed worst-case performance. To this end, and in contrast to many best-effort approaches, we provide a framework that allows assessing performance and enables network design with regard to its requirements. The proposed MAC has an asynchronous nature leading to low complexity and good energy efficiency which, together with its ability to adapt to network requirements and environment, makes it suitable for a wide range of applications.

A. Contributions

In this paper, we propose a MAC protocol for highly reliable and energy efficient communication in WSNs. In particular, nodes try to send each packet a maximum number of k times. For this, they wait a random back-off time in-between transmission attempts, selected from an interval $[t_{min}, t_{max}]$. By adjusting t_{min} and t_{max}, we can influence the probability of a successful transmission and are therefore able to provide a guarantee on reliability, i.e., that the packet reaches its destination within a specified deadline.

Carrier sensing is used to detect ongoing transmissions by other nodes and skip own ones if these would cause collisions. To this end, we determine the optimal length of the carrier sensing interval with regard to node specific parameters. In addition, every successful transmission is acknowledged, hence, nodes can stop trying to transmit further packets if data has been already received. This greatly reduces the generated traffic and results in improved energy efficiency and delays.

B. Structure of the Paper

The rest of this paper is structured as follows. Related work is discussed in Section II. Next, Section III explains our system model and assumptions. Section IV introduces the proposed MAC protocol. Section V presents our experimental evaluation based on simulation and Section VI concludes the paper.

II. Related Work

In the following we provide a brief overview of different MAC protocols that aim to increase energy efficiency and reliability in WSNs. In general, these can be divided in synchronous, asynchronous and hybrid networks [8].

In synchronous networks, nodes typically share a common clock by periodically sending beacon messages or by synchronizing to external events. This allows them to efficiently schedule transmissions and sleeping times [3] [9], resulting in high maximum throughput and bounded delays. On the other hand, synchrony comes at the cost of additional energy consumption and complexity, making it less suitable for networks with long idle times or for environments with high levels of external interference, as control messages can be lost as well. A typical synchronous protocol is TDMA.

Asynchronous networks, on the other hand, forgo any synchronization, but instead implement mechanisms such as random back-off or retransmission schemes to increase reliability. They offer good energy efficiency and high flexibility, making them ideal for dynamic, low-cost networks. On the other hand, since transmissions are uncoordinated, they incur in higher packet loss rates and a typically unbounded delay, especially for higher network traffic. A widely used asynchronous protocol is CSMA.

Hybrid approaches try to combine the advantages of synchronous and asynchronous networks. This can, for example, be done by switching from CSMA to TDMA under high contention [16] or by creating a super frame, which divides transmission time in synchronous and asynchronous slots [13]. Although these approaches increase the average reliability, they incur additional complexity and are usually limited to the performance of either TDMA or CSMA in the worst case.

In [12], an asynchronous approach for fully reliable communication is presented. Here, nodes transmit data packets with carefully chosen (constant) inter-packet times to guarantee that at least one packet arrives in the worst case. This way, 100 % reliability can be achieved, however, it comes at the cost of increased delays making it only suitable for smaller network sizes. To reduce these delays, the probabilistic approach in [11] transmits data packets at random time instants, similar to the protocol in this paper. This allows providing worst-case guarantees on delay and reliability, however, the scheme in [11] is only applicable for transmit-only networks and, hence, does not perform carrier sensing to adapt to ongoing transmissions on the channel.

A modified CSMA mechanisms is presented in [1], which uses RSSI (received signal strength indication) to obtain additional information about interfering sources and decide whether to transmit or not. This increases the average performance, compared to classic CSMA; however no analysis is provided to analyze its worst-case performance.

Other protocols adjust their MAC parameters depending on the current network status or on environmental changes. For example, back-off times can be prolonged to reduce congestion in bursty traffic [16], or the data rate can be adapted to reduce energy consumption whenever possible [6]. These approaches generally increase the average performance, however, they are again limited to the performance of the underlying MAC layer (typically TDMA or CSMA) in the worst case.

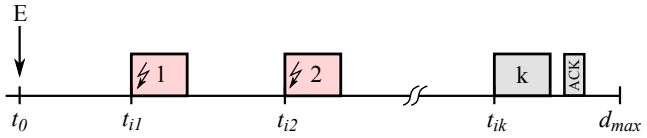

Figure 1. Basic working principle: After being activated by an event E, node i starts transmitting data packets in random time intervals until an acknowledgment is received — here for the k-th packet. This way, up to k packets are sent within a deadline d_{max} in the worst case.

In addition, further MAC protocols have been presented that improve reliability and/or energy efficiency in WSNs. However, many of them make restrictive assumptions limiting their applicability. For example, spatial separation of nodes [4], non-standard modulation [5], multiple receivers per node [2], etc. These are hence not suitable for general purpose networks, such as the one considered in this paper.

In this paper, we propose a MAC protocol for highly reliable and energy efficient communication in WSNs. In contrast to the aforementioned approaches, we do not pursue a best effort approach, but provide a framework that allows calculating the expected worst-case reliability and energy consumption for a given network. To this end, node specific parameters are regarded during calculations, resulting in a typically better performance than fixed-parameter MACs, such as CSMA. The presented approach is general and can be applied to a wide variety of different applications using commonly available hardware.

III. Models and Assumptions

We consider a WSN of one or more (data) sinks and a number of n independent (sensor) nodes that can be activated periodically or by events. Upon activation, nodes broadcast their data in a single-hop (star topology) fashion to their corresponding sink. The sink can then process this data or act as a cluster head and further relay it to other sink nodes in a multi-hop fashion[1]. To ensure proper functionality, data packets must be received within an upper time bound or deadline d_{max}, i.e., this is the maximum time allowed for one hop.

Transmitting a data packet takes an amount of time, depending on the size of the packet and the transmission rate. We refer to this value as *packet length* and denote the maximum length of any packet in the network by l_{max}. To increase reliability — individual transmission can be corrupted, since nodes are not synchronized — every sink acknowledges successful

[1]Note that if the sink acts as a cluster head, it must be regarded as a transmitting node as well, i.e., it must be added to n.

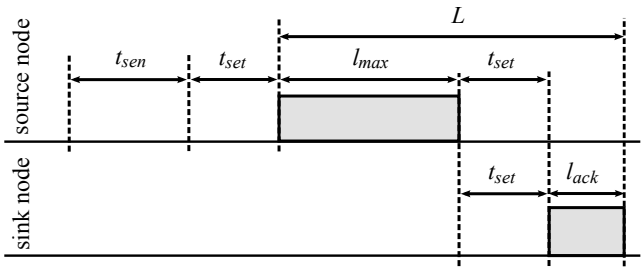

Figure 2. Timing of a successful packet transmission.

receptions of data packets. We refer to the length of an acknowledgment packet (ACK) by l_{ack}. If no ACK is received, the corresponding node re-transmits a packet up to $k-1$ times within d_{max}, where k is an integer number denoting a maximum bound on transmission attempts by any node in the network.

Each node i waits a random time $t_{ix} \in \mathbb{R}_{>0}$ before sending any packet x — including the first packet of a sequence[2]. Here $1 \leq i \leq n$ and $1 \leq x \leq k$ hold and $n \in \mathbb{N}_{>0}$ denotes the number of (sensor) nodes in the system — see Fig. 1. This t_{ix} is referred to as (random) inter-packet or back-off time and is uniformly distributed in the interval $[t_{min}, t_{max}]$ where $t_{min}, t_{max} \in \mathbb{R}_{>0}$ are system design parameters, i.e., these are common to all nodes. After waiting a random time, nodes do not transmit directly, but they first sense the channel to check whether the communication channel is being used, i.e., blocked. If the channel is blocked, they skip the current packet and perform a random back-off before trying again. To this end, nodes can detect transmissions of the other nodes that are connected to the same sink, i.e., there are no hidden terminals within a sink's range.

In case the channel is free, i.e., not blocked, a packet is sent as depicted in Fig. 2. Here, a node first senses the channel for t_{sen} time and then switches its transceiver from receive to transmit mode — in t_{set} time — to transmit its data packet. After transmission, the node switches back to receive mode to be able to receive an ACK from the sink. Similarly, the sink also performs a mode switch to transmit the ACK, which again takes t_{set} time. Note that t_{set} includes possible latencies such as processing and propagation delays.

Once the acknowledgment has been sent, the sink node needs t_{set} additional time to switch back to receive mode and be able to receive further packets. However, since there cannot be any packet in this last t_{set}, we can safely neglect it. That is, another node can start transmitting at earliest $t_{sen} + t_{set}$ time after the channel is free, i.e., after the acknowledgment has been sent — more details later. As a consequence, we denote by L the total delay incurred from the start of transmission by the source node to the time at which the sink node finishes its acknowledgment — see Fig. 2:

[2]This design decision simplifies our analysis in a considerable manner, while not affecting the functionality of the network.

$$L = l_{max} + l_{ack} + t_{set}. \tag{1}$$

Similar to CSMA, every sink monitors the communication channel continuously, since communication is asynchronous and packets can be received at any time. This increases the energy consumption of the sink, which can, however, be tolerated in many WSN applications. For example, in home automation networks, sinks are usually attached to actuators and, hence, typically connected to a bigger power supply such as the house's power line. In return, energy is saved at multiple, battery-powered sensor nodes, where this is a generally much more critical concern (e.g., wireless light switches and temperature sensors, etc.).

Finally, each transmitter is assumed to be activated only once within d_{max}. This is a logical assumption since multiple activations of the transmitters lead to unnecessary interference.

IV. Proposed Scheme

In this section, we obtain suitable values for the parameters introduced before in Section III. In particular, we will select values for t_{sen}, t_{min}, t_{max} and k that allow guaranteeing a reliable and energy-efficient communication between nodes in our WSN.

A. Carrier Sensing

As mentioned above, nodes perform carrier sensing for a configurable time t_{sen} to avoid interrupting ongoing transmissions of other nodes. In the following, we analyze the optimal length of the sensing period t_{sen}.

To this end, we have to consider that every transceiver IC has a specific sensitivity, i.e., it requires a certain amount of time to detect whether a channel is busy or not. This is typically the time to receive a few bits at a given transmission speed, which we denote by $\bar{t}$ in the following. Note that a signal must be present for at least $\bar{t}$ *continuously* on the channel for a node to detect it reliably.

The shortest possible sensing interval can be as short as $t_{sen} = \bar{t}$. However, as displayed in Fig. 2, we know that there is a gap of size t_{set} in between data packets and the corresponding acknowledgments. Now, in order to not falsely detect the channel as free during this time, t_{sen} should be chosen such that:

$$t_{sen} = t_{set} + 2\bar{t}. \tag{2}$$

By adding $2\bar{t}$ in (2), we ensure that t_{sen} always overlaps by at least $\bar{t}$ *continuously* with either the data or acknowledgment packet. Note that a longer t_{sen} than (2) has no benefit, but it rather increases energy consumption of the node without guaranteeing better results.

Although carrier sensing greatly reduces the chance of collisions, it cannot fully prevent them. That is, since nodes can be triggered by independent events, it may happen that the carrier sense intervals of two or more nodes overlap such that they cannot see each other. For example, let us assume that

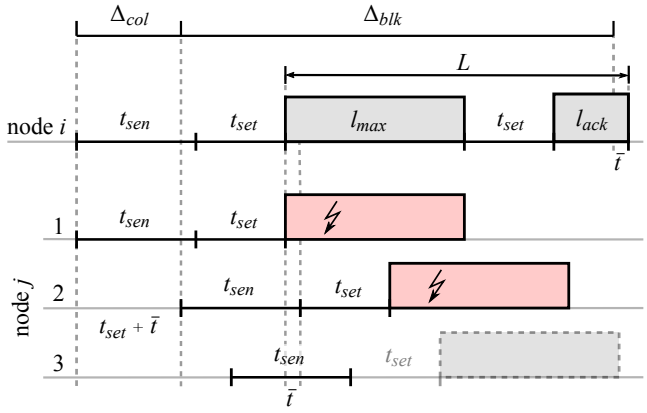

Figure 3. Illustration of the collision and blocking intervals Δ_{col} and Δ_{blk}. Whenever a node j starts carrier sensing in these intervals, its packet will either collide or be blocked by a packet of another node i. Case 1 and 2 show a collision between node i and j, whereas case 3 shows node j skipping its packet.

a node 1 and a node 2, sense the communication channel for $50\,\mu$s and $10\,\mu$s respectively, before they start transmitting. It can happen that node 2 is triggered $40\,\mu$s after node 1 such that their sensing intervals do not overlap with the other's packet transmission. As a result, they do not detect each other and packets still get lost. This is analyzed in the following section with more detail.

B. Reliability

Recall that each node in the network tries to send each packet a total number of k times waiting a random time interval t_{ix} in $[t_{min}, t_{max}]$ before each attempt. Let us now consider the definition below.

Definition: We define reliability of a WSNs as the probability that, in the worst case, at least one out of k packets of any node i reaches its destination within a specified deadline d_{max}.

To compute this probability, we need to consider the worst-case transmission conditions: (i) all n nodes in the network are sending (ii) there exists a maximum fraction of the interval $[t_{min}, t_{max}]$ for which any selected value of t_{ix} leads to a failed transmission. While condition (i) is straight forward, condition (ii) requires more analysis.

There are two possibilities that lead to a failed transmission attempt, either a collision or a blocked channel. Both possibilities can be expressed as time intervals, i.e., Δ_{col} for collisions and Δ_{blk} for a blocked channel. These describe the fraction of time in $[t_{min}, t_{max}]$ for which any selected value of t_{ix}, i.e., the point in time at which nodes start with carrier sensing, leads to either collision or blocking on the channel.

Regarding collisions, let us have a look at Fig. 3. As depicted in case 1, a packet of node j collides with a packet of another node i, if both start carrier sensing at the same time. Now, if we shift the packet of node j further to the right, i.e., let it start carrier sensing later, there will still be a collision as shown in case 2. Only when this shift is greater than $t_{set} + \bar{t}$, there will be no collision, but node j's packet will be skipped. That is, its carrier sensing interval now overlaps for more than $\bar{t}$ with node i's packet, which is sufficient for node j to detect it. As a result, the collision interval of a single packet can be described as:

$$\Delta_{col} = t_{set} + \bar{t}, \tag{3}$$

where t_{set} is the mode switch time and $\bar{t}$ the sensitivity of the transceiver.

Similarly, we can determine the blocking interval Δ_{blk}, as shown by case 3 in Fig. 3. Here, a node j is able to detect a busy channel, if it starts carrier sensing more than $t_{set} + \bar{t}$ later than another node i, i.e., after the collision interval. This continues until the point in time where node j starts carrier sensing less than $\bar{t}$ time before node i's acknowledgment has finished. That is, the overlap of node j's t_{sen} with node i's acknowledgment is small enough and node j starts detecting a free channel. As a result, the blocking interval of a single packet can be described as follows:

$$\begin{aligned}\Delta_{blk} &= (t_{sen} - (t_{set} + \bar{t}) + t_{set} + L - \bar{t}) \\ &= (t_{sen} + L - 2\bar{t}) = t_{set} + L.\end{aligned} \tag{4}$$

Although there are differences between a packet failed due to a collision or a blocked channel as discussed later, it does not make any difference for reliability, since no packet will arrive at the sink in both cases. We can consequently combine Δ_{col} and Δ_{blk} to describe the total fraction of time in $[t_{min}, t_{max}]$ for which any selected value of t_{ix} will lead to a failed transmission between any two single packets:

$$\begin{aligned}\Delta_{tot} &= \Delta_{col} + \Delta_{blk} \\ &= (t_{sen} + t_{set} + L - \bar{t}).\end{aligned} \tag{5}$$

So far, we discussed the carrier sense mechanism and the blocking and collision intervals. Next, we analyze how to obtain t_{min} and t_{max}, i.e., the bounds in which every node i uniformly selects its inter-packet times t_{ix}. To this end, let us consider the case where t_{min} is set such that there can be at most one packet of each node in an interval of length $t_{max} - t_{min}$. That is, the value of t_{min} has to fulfill the following condition:

$$\begin{aligned} t_{min} &\geq t_{max} - t_{min}, \\ t_{min} &\geq \frac{t_{max}}{2}. \end{aligned} \tag{6}$$

Note that t_{min} is the minimum and t_{max} the maximum separation between two consecutive transmission attempts of a node. If t_{min} is smaller than $\frac{t_{max}}{2}$, each node can send multiple packets within the interval $t_{max} - t_{min}$, for example, if it (randomly) selects t_{min} multiple times. This, however, leads to a lower worst-case performance and is therefore not meaningful — more details can be found in [11], where this is analyzed for a transmit-only network.

Given the fact that there can be at most one packet per node in $t_{max} - t_{min}$, we can compute the maximum possible probability of packet loss for every packet being sent. This is the ratio between Δ_{tot} of all packets — if all nodes are transmitting, there can be $n-1$ other packets that can cause interference — and $t_{max} - t_{min}$:

$$q = \frac{(n-1)\Delta_{tot}}{t_{max} - t_{min}}. \tag{7}$$

The probability of successful packet transmission in the worst case is given by $1-q$. Note that for (7) to be valid the following condition must be satisfied (i.e., $q \leq 1$ must hold):

$$\begin{aligned} (n-1)\Delta_{tot} &\leq t_{max} - t_{min}, \\ t_{min} &\leq t_{max} - (n-1)\Delta_{tot}. \end{aligned} \tag{8}$$

Since network parameters, such as t_{min}, t_{max}, t_{sen}, etc., are common to all nodes, q is independent of the node and packet being sent. This allows us to model reliability, i.e., the probability p that at least one out of k transmission attempts reaches its destination for any node n, using a binomial distribution.

To this end, we need to consider all possible combinations, i.e., the first packet arrives, the second packet arrives, etc., which is a cumbersome procedure. To facilitate calculations, we instead compute $1-p$, i.e., the probability that, in the worst case, no transmission attempt is successful. This is the probability that k consecutive packets are lost and can be computed by the well-known equation $\binom{k}{x} q^x (1-q)^{(k-x)}$ where $\binom{k}{x} = \frac{k!}{x!(k-x)!}$ is the binomial coefficient. Replacing q as per (7), we obtain with $x = k$, i.e., k out of k packets are lost:

$$1 - p = \left(\frac{(n-1)\Delta_{tot}}{t_{max} - t_{min}} \right)^k. \tag{9}$$

For (9) to be valid, we have to ensure that nodes are always able to send k packets within d_{max}. Towards this, recall again that every node i waits a random time t_{ix} chosen from $[t_{min}, t_{max}]$ before sending any packet. In the worst case, node i will select t_{max} for each of its k packets. To guarantee that even the last packet of node i has been transmitted before d_{max}, the following must hold:

$$t_{max} \leq \frac{d_{max} - (t_{sen} + t_{set} + L)}{k}. \tag{10}$$

Given a value of t_{max} as per (10), we can reshape (9) to compute the value of t_{min} that satisfies a desired reliability p for the whole WSN:

$$t_{min} \leq t_{max} - \frac{(n-1)\Delta_{tot}}{\sqrt[k]{1-p}}. \tag{11}$$

We can see from (11) that full reliability, i.e., $p = 1$, is only possible for $n = 1$, independent of all other parameters. For $n > 1$, if p tends to 1, t_{min} tends to minus infinity as per (11). In other words, 100% reliability as with TDMA or the MAC in [12] cannot be achieved with the proposed approach. However, our scheme allows for a reliability that is acceptably close to 100%, while considerably reducing the number of transmission attempts and, hence, making better use of energy.

C. Energy Consumption

Let us now analyze the energy consumption E_{avg} of a node, i.e., the energy required on average for sending up to k packets within d_{max}. To this end, recall that there are three possibilities for every data transmission: (i) the packet is successfully received, (ii) the packet collides and (iii) the channel is blocked and a back-off is performed.

In case a packet is successfully received, as shown for node j in Fig. 3, a node senses the channel for t_{sen}, sends a packet with l_{max} and receives an acknowledgment in l_{ack} time. In case a packet collides, the node also waits for an acknowledgment, since it does not know a priori that its packet collided. It consequently requires the same energy as a successful transmission. In contrast, if the channel is blocked, a node only senses the channel for t_{sen} and then performs a back-off. We assume that a node sleeps during the back-off time and, hence, does not consume energy. As a result, the energy needed for a blocked, collided or successful packet can written as:

$$\begin{aligned} E_{blk} &= t_{sen} \cdot P_{rx}, \\ E_{snd} &= (t_{sen} + l_{ack}) \cdot P_{rx} + l_{max} \cdot P_{tx} + 2t_{set} \cdot P_{set}, \end{aligned}$$

where E_{blk} is the energy needed for skipping a packet and E_{snd} is the energy for a successful or collided packet. P_{rx} and P_{tx} are power levels of the node in receive and transmit mode respectively and P_{set} describes the power consumption during t_{set}, for which we assume $P_{set} = max(P_{tx}, P_{rx})$.

In the best case, a packet arrives at the first attempt and only E_{snd} is consumed. In the worst case, all k attempts to transmit a packet result in collisions, which requires $k \cdot E_{snd}$. These are the lower and upper bound for the average energy E_{avg}:

$$E_{snd} \leq E_{avg} \leq k \cdot E_{snd}.$$

We first have to distinguish between the probability that a single packet collides q_{col} or is skipped due to a blocked channel q_{blk}. With $q = q_{col} + q_{blk}$ as per (7), this results in:

$$\begin{aligned} q_{col} &= \frac{(n-1)\Delta_{col}}{t_{max} - t_{min}}, \\ q_{blk} &= \frac{(n-1)\Delta_{blk}}{t_{max} - t_{min}}. \end{aligned}$$

Further, we have to consider all different possibilities of packet failure and arrival, i.e., the first packet arrives, the first fails and the second arrives, etc., including all k packets fail. To this end, note that, if a packet is successful, its weighted energy is $(1-q) \cdot E_{snd}$, i.e., the probability that the packet does not fail, multiplied by the energy for successful transmission.

If a packet fails, the weighted energy is either $q_{col} \cdot E_{snd}$ in case of a collision or $q_{blk} \cdot E_{blk}$ in case of being blocked.

If now x_1 packets collide and x_2 packets are blocked, with $x_1 + x_2 \leq k$, until a packet can be successfully sent (or all k attempts are unsuccessful), we have that the energy required is:

$$E(x_1, x_2) = x_1 \cdot E_{snd} + x_2 \cdot E_{blk} + \min(1, k - x_1 - x_2) \cdot E_{snd},$$

and that the probability of having this case is given by:

$$q(x_1, x_2) = \min_{>0}(1, \min(1, k - x_1 - x_2) \cdot (1 - q)) \times \binom{x_1 + x_2}{x_1} \cdot q_{col}^{x_1} \cdot q_{blk}^{x_2},$$

where $\binom{x_1+x_2}{x_1} = \frac{(x_1+x_2)!}{x_1!x_2!}$ is the binomial coefficient returning the number of combinations for given values of x_1 and x_2. In addition, $\min_{>0}(\cdot)$ returns the minimum value that is greater than 0, i.e., it returns 1 when $k - x_1 - x_2 = 0$ holds, otherwise it returns $1 - q$.

As a result, the average energy E_{avg} is given by the following expression:

$$E_{avg} = \sum_{x_1=0}^{k} \sum_{x_2=x_1}^{k} q(x_1, x_2) \cdot E(x_1, x_2). \quad (12)$$

In conclusion, (12) can be used to calculate the average energy consumption of a transmission attempt and, hence, allows assessing the expected battery lifetime of a node. Together with the equations in Section IV-B, this allows designing the network for both reliable and energy-efficient operation.

V. Simulation and Evaluation

In this section, we evaluate our MAC protocol by a set of simulation-based experiments and compare it to other protocols as explained below. To this end, we used the OMNeT++ framework [15] and MiXiM [7], an extension for wireless networks, to obtain statistical data for different network settings and very high numbers of transmissions — at least 1,000,000 packets where simulated for each of the presented curves to reach good average results.

The simulated WSN consists of one data sink and a selectable number of sensor nodes n that are randomly distributed within an area of $30 \times 30\,m^2$. The transmission power has been configured to ensure good link quality over the whole area and enable a good connection to the sink, which is located in the center of this area. For simplicity, we assume that data is conveyed in a single-hop fashion. This facilitates the following analysis while not invalidating results.

All sensor nodes are simple data sources that transmit packets according to the compared MAC protocols below. To reproduce a high network load and therefore conditions close to the worst case, all nodes were triggered as frequently as possible, i.e., without or with only very short pauses between subsequent activations. This results in long delays, high energy consumption, etc., which can be regarded as exceptional operation conditions. During normal operation, i.e., less congestion, nodes are expected to yield better results.

Table I
CC110L radio and simulation parameters

Parameter	Value
Bit rate	$128\,kbps$
RX/TX switching time t_{set}	$30\,\mu s$
Sensitivity $\bar{t}$	$10\,\mu s$
Carrier sense duration t_{sen}	$50\,\mu s$
CSMA slot length	$320\,\mu s$
RX threshold	$-95\,dBm$
TX power	$+12\,dBm$
Area	$30 \times 30\,m^2$

The following MAC protocols are compared in simulation:

- The *proposed* scheme is our MAC protocol as presented in Section IV of this paper.
- The *TDMA* (time division multiple access) scheme is a synchronous protocol, in which nodes transmit during dedicated time slots to avoid collisions. For this, they share a common clock by periodically receiving synchronization beacons from the sink.
- The *CSMA* (carrier sense multiple access) scheme is an asynchronous protocol that implements non-persistent carrier sensing and an exponential back-off scheme, similar to 802.15.4.
- The *deterministic* scheme from [12] uses fixed back-off times and allows fully reliable communication (clearly, provided that external interference can be neglected).

Note that we selected baseline *CSMA* and *TDMA* for comparison, since these are the core technologies of many other relevant approaches [9] [13] [16]. During high contention — which we simulate in our experiments — these typically fall back to the performance of either *TDMA* or *CSMA*. For example, [13] reduces to *CSMA* and [9] [16] to *TDMA* at high congestion. For more information, see Section II.

In our experiments, we fixed the transmission rate to $128\,kbps$ and used the radio parameters from the CC110L transceiver [14], as displayed in Table 1. Data packets consist of 4 bytes preamble, 2 bytes CRC (cyclic redundancy check) and 26 bytes payload, resulting in $l_{max} = 2000\,\mu s$. In case of the acknowledgment, no CRC is required, but a field for the (source) node's address of 1 byte. Its length is consequently $l_{ack} = 312.5\,\mu s$. Further, the transceiver switching time t_{set} has been set to $30\,\mu s$ and the receiver sensitivity to $\bar{t} = 10\,\mu s$, i.e., the time to detect one bit including some tolerance.

For the *proposed* scheme, the packet number was set to $k = 3$ and the deadline to $500\,ms$, which is common in home automation networks, e.g., a wireless light switch needs to

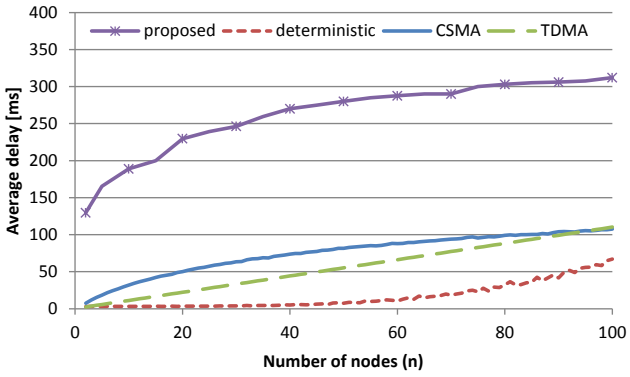

Figure 4. The average delay, i.e, the average time required for successfully transmitting data.

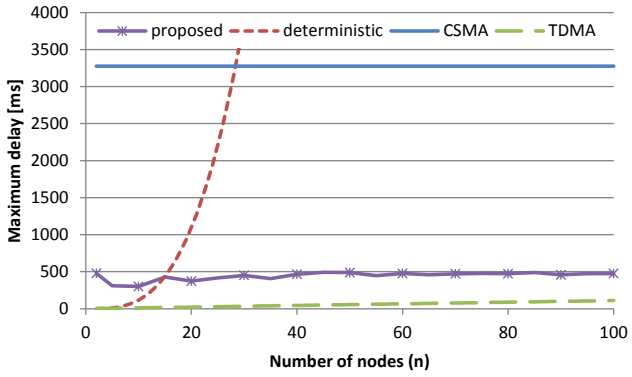

Figure 5. The maximum delay, i.e., the longest time it can take from triggering a node until the successful reception of its data.

operate within this time to guarantee the desired functionality. The remaining parameters were calculated using the provided formulas from Section IV, i.e., t_{max} was computed by (10) whereas we assumed $t_{min} = \frac{t_{max}}{2}$ for maximum reliability.

In case of *TDMA*, every transmission cycle starts with a synchronization beacon followed by n time slots, where n is the number of nodes of the current iteration. The beacon is set to a length of $375\,\mu s$, i.e., it consists of a 4 bytes preamble and a 2-byte sync word. On the other hand, time slots are set to $l_{max} = 2000\,\mu s$. To account for clock drift — we assume standard crystal oscillators with $100\,ppm = 0.01\,\%$ accuracy — all slots are enclosed by a $5\,\%$ guard interval, i.e., $100\,\mu s$ before and after the slot. Taking this value as the maximum allowable clock drift, the beacon interval is set to $1\,s$. In other words, the beacon prevents clocks from deviating for more than $100\mu s$.

The *CSMA* protocol is based on the IEEE 802.15.4 MAC layer, which we modified to match the different packet sizes and transmission rates in our network. That is, we set the number of back-off retries to 4, retransmissions to 7 and select a slot size of $320\,\mu s$. Further, we increased the contention window size to $[32, 1024]$ slots as per 802.11 to improve performance at high contention. Lastly, to be comparable to the other approaches, we use the same packet structure as before (i.e., l_{max}) and do not implement RTS/CTS schemes.

A. Communication Delay

The average delay, i.e., the average time it takes from triggering a node until the successful reception of its data, is shown in Fig. 4. As it can be observed, a higher n increases the average delay for all transmission schemes. In the case of *TDMA*, it rises linearly due to a higher number of slots within the cycle. For the other schemes, an increasing n leads to more congestion and, therefore, results in higher collisions rates. In case of the *proposed* scheme and *CSMA*, this causes the delay to rise linearly, where the slight curvature results from saturation effects, i.e., due to a limited number of back-off retries and re-transmissions. For the *deterministic* scheme, the average delay increases quadratically as a result of longer back-off times for a greater n [12].

As shown in Fig. 4, the *proposed* scheme has the highest average delay of all schemes. Unlike *CSMA*, which tries to achieve fast transmission, i.e., by first using short back-off times, the *proposed* scheme equally distributes its k transmissions within the full deadline d_{max}. This results in longer average delays, however, it also leads to a more balanced traffic load. For example, in bursty networks, e.g., when an event triggers multiple nodes at the same time, the *proposed* scheme yields no traffic peaks, resulting in less collisions in total. This is beneficial for energy efficiency and reliability, as shown in the following experiments.

The maximum delay, i.e., the longest time from activation of a node until successful reception of its data, is shown in Fig. 5. Here, the *proposed* scheme offers a relatively low maximum delay equal to the deadline d_{max}. Only *TDMA*, a throughput optimized synchronous protocol, can achieve a lower maximum delay, i.e., the time of one full TDMA cycle. In contrast, *CSMA* and the *deterministic* scheme have an exponentially rising delay. For CSMA, this rises with the number of back-off retries and transmission attempts, hence, it is independent of n and shows a constant value in Fig. 5. For the *deterministic* scheme, however, the maximum delay increases exponentially with 3^n. This results in very high delays, for example, $1.1\,s$ for $n = 20$ and $2.5\,min$ for $n = 100$, making this scheme only suitable for networks that tolerate very long delays.

B. Energy Consumption

Let us now take a closer look at the energy consumption of the different MAC protocols. In Fig. 6, we can see the results of an exemplary network running for a period of 1 year, in which nodes were triggered 1000 times per day on average. Again, to achieve high network load, event activity is limited to short intervals throughout the day. This results in short peaks of traffic, providing us with an upper bound on energy consumption. During normal operation, i.e., when

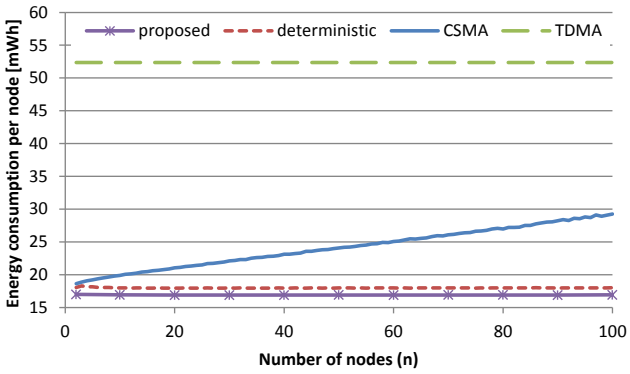

Figure 6. The average energy consumed per node in a 1 year deployment, if it is triggered 1000 times per day.

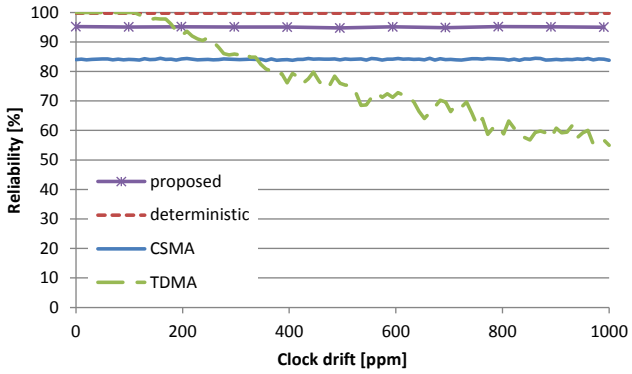

Figure 7. The average transmission reliability with respect to an increasing clock drift.

there are less events or these are better distributed, energy consumption will be usually lower.

The amount of energy consumed was calculated by multiplying the recorded times each node spent in receive, transmit and sleep mode with the corresponding power levels from the CC110L transceiver. That is, a node requires 33 mW in receive, 66 mW in transmit and $0.36\,\mu W$ in sleep mode at $V_{DD} = 1.8\,V$.

In case of the asynchronous protocols, i.e., the *CSMA*, *deterministic* and *proposed* scheme, the missing synchronization leads to a generally lower energy consumption, which increases for higher n, since more collisions occur and packets must be re-transmitted more often. This effect is more dominant for *CSMA*, whereas it can barely be noticed for the *proposed* and *deterministic* schemes. For the *deterministic* scheme, a rising n leads to longer back-off times, resulting in roughly constant collision numbers. The proposed scheme, on the other hand, generally achieves low collision numbers due to its evenly distributed transmissions within d_{max}.

The *TDMA* scheme has the highest overall energy consumption, since nodes must wake up periodically to receive synchronization beacons and to adjust their internal clocks. The energy consumed hereby is determined by the beacon duration and its frequency, which depends on the precision of the oscillator and the length of the guard time intervals — both are independent of n. In addition, since packets must be transmitted only once due to the interference-free slot design of *TDMA*, the energy consumption is independent of n and shows a constant value in Fig. 6.

Note that we used a precision oscillator with $30\,ppm$ accuracy for *TDMA* in Fig. 6 instead of the previous $100\,ppm$. This way, the power consumption has been decreased by around $60\,\%$. To further improve efficiency, special synchronization protocols can be used, for example, FTSP [10], which implements techniques to estimate and compensate clock drift. Although the synchronization overhead can be greatly reduced by these methods, energy consumption will typically still be higher than for asynchronous protocols, especially, in networks with low event activity and long idle times.

C. Clock Drift

In the following, we discuss the effects of clock drift, i.e., how the naturally occurring variance of an oscillator's frequency affects a node's behavior. To this end, we simulated an exemplary network with $n = 20$ nodes for which we slowly increased clock drift to $1000\,ppm$ ($= 0.1\,\%$).

In Fig. 7, we can see that all asynchronous approaches, i.e., the *CSMA*, *deterministic* and *proposed* scheme, are barely affected by an increasing clock drift due to their, if necessary, repeated number of transmissions, carrier sensing and lack of synchronization. For *TDMA*, however, nodes can possibly violate their slot boundaries and collide with neighboring slots for rising drift values. This effect starts when the simulated drift is higher than $100\,ppm$, i.e., when it exceeds the maximum value we have considered for design. Although not shown in Fig. 7, other system parameters, such as delay, energy consumption, etc., are affected in a similar way. That is, for *TDMA* these typically increase with higher drifts, whereas (almost) no change was observed for the other protocols.

D. External Interference

So far we considered internal interference only, i.e., packet collisions occur from simultaneous transmission within the network. In real-world deployments, however, there will inevitably be interference from sources outside the network, for example, from neighboring devices using WiFi or Bluetooth. To show possible effects on the different MAC protocols, we simulated a network with $n = 20$ nodes and stepwise increased interference from $0\,\%$ (no interference) to $100\,\%$ (blocked channel). We considered that interference pulses have a random length of $0.3\,ms$ to $10\,ms$ and always corrupt ongoing transmissions (SINR $< 0\,dB$).

Clearly, a rising level of interference impairs the overall performance such as delay, reliability, energy consumption of

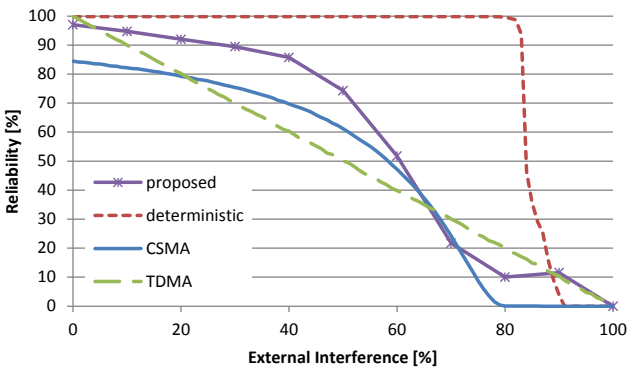

Figure 8. The average transmission reliability for an increasing level of external interference.

all MAC protocols. Especially *TDMA* is vulnerable to interference, since no acknowledgment or carrier sense mechanisms are implemented, and losing synchronization beacons leads to additional timing errors. As shown in Fig. 8, its reliability drops linearly for a rising level of interference, since packets are transmitted only once.

In contrast, the other approaches are generally robust against interference and show good reliability even in noisy environments (interference $< 40\,\%$). For higher values, reliability starts to rapidly decrease, since transmission and back-off retries start not to be sufficient anymore. The *deterministic* scheme offers the highest robustness, because it implements the highest retransmission numbers. In case of the *proposed* scheme, we can observe a higher reliability than for *CSMA*, although it has lower retransmission numbers.

VI. Concluding Remarks

In this paper, we presented a MAC protocol for designing highly reliable and delay-bounded WSNs. More specifically, each node sends data packets up to k times with random back-off times selected from $[t_{min}, t_{max}]$. The proposed MAC allows configuring network parameters so as to guarantee a desired reliability in the form of the probability that one packet reaches its destination within a specified deadline d_{max}. Due to its asynchronous design and reduced carrier sense times, the proposed scheme makes efficient use of energy during low network load and offers reliable communication during high loads. It is therefore suitable for a broad range of applications in the era of IoT.

In contrast to many existing MAC protocols, we do not pursue a best effort approach, but provide a framework for calculating the expected system behavior. Our scheme allows for adaptability and can also dynamically adapt to different network states or the environment. For example, if a node has less important data to transmit, it can dynamically extend its deadline to reduce congestion and, therefore, increase the transmission reliability of the remaining nodes.

By performing extensive simulations using OMNeT++, we showed that the proposed MAC achieves low collisions rates within the network and high robustness against clock drift and external interference. In addition, energy efficiency is strongly improved due to a more balanced traffic load, which, as a result, leads to significantly improved performance in comparison to conventional protocols, such as TDMA and CSMA. With respect to approaches with deterministic back-off times, the proposed technique allows for shorter reaction times in the worst case while still guaranteeing a high reliability.

References

[1] E. Celada-Funes, D. Alonso-Roman, C. Asensio-Marco, and B. Beferull-Lozano. A Reliable CSMA Protocol for High Performance Broadcast Communications in a WSN. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2014.

[2] B. Firner, C. Xu, R. Howard, and Y. Zhang. Multiple Receiver Strategies for Minimizing Packet Loss in Dense Sensor Networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2010.

[3] L. Guntupalli, F. Y. Li, and J. Martinez-Bauset. Event-Triggered Sleeping for Synchronous DC MAC IN WSNs: Mechanism and DTMC Modeling. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2016.

[4] J. Guo and H. Jafarkhani. Sensor Deployment With Limited Communication Range in Homogeneous and Heterogeneous Wireless Sensor Networks. *IEEE Transactions on Wireless Communications*, 15:6771–6784, 2016.

[5] G. Kaddoum, E. Soujeri, and Y. Nijsure. Design of a Short Reference Noncoherent Chaos-Based Communication Systems. *IEEE Transactions on Communications*, 64:680–689, 2016.

[6] P. S. Khairnar and N. B. Mehta. Discrete-Rate Adaptation and Selection in Energy Harvesting Wireless Systems. *IEEE Transactions on Wireless Communications*, 14:219–229, 2015.

[7] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin. Simulating Wireless and Mobile Networks in OMNeT++: The MiXiM Vision. In *Proceedings of the International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools)*, 2008.

[8] K. Langendoen and A. Meier. Analyzing MAC Protocols for Low Data-Rate Applications. *ACM Transactions on Sensor Networks*, 7:19, 2010.

[9] C.-J. Liu, P. Huang, and L. Xiao. TAS-MAC: A Traffic-Adaptive Synchronous MAC Protocol for Wireless Sensor Networks. *ACM Transactions on Sensor Networks*, 12:1:1–1:30, 2016.

[10] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi. The Flooding Time Synchronization Protocol. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys)*, 2004.

[11] P. Parsch and A. Masrur. A Reliability-Aware Medium Access Control for Unidirectional Time-Constrained WSNs. In *Proceedings of the International Conference on Real Time and Networks Systems (RTNS)*, 2015.

[12] P. Parsch and A. Masrur. A Reliable MAC for Energy-Efficient WSNs in the Era of IoT. In *Proceedings of the Euromicro Conference on Digital System Design (DSD)*, 2016.

[13] B. Shrestha, E. Hossain, and K. W. Choi. Distributed and Centralized Hybrid CSMA/CA-TDMA Schemes for Single-Hop Wireless Networks. *IEEE Transactions on Wireless Communications*, 13:4050–4065, 2014.

[14] Texas Instruments. CC110L Datasheet. URL: http://www.ti.com/lit/ds/symlink/cc110l.pdf.

[15] A. Varga. The OMNeT++ Discrete Event Simulation System. In *Proceedings of the European Simulation Multiconference (ESM)*, 2001.

[16] S. Zhuo, Z. Wang, Y. Q. Song, Z. Wang, and L. Almeida. A Traffic Adaptive Multi-Channel MAC Protocol with Dynamic Slot Allocation for WSNs. *IEEE Transactions on Mobile Computing*, 15:1600–1613, 2016.