A Reliable MAC for Energy-Efficient WSNs in the Era of IoT

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

***Abstract*—Wireless sensor networks (WSNs) are gaining in importance with an increasing need for interconnectivity in the advent of Internet of Things. In some application scenarios, such as building and home automation, WSNs need to comply with deadlines and guarantee a reliable communication, for which a suitable MAC (Medium Access Control) is of paramount importance. However, existing approaches from the literature are either unable to provide such guarantees (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 fulfilling a maximum delay constraint or deadline. We perform a large set of experiments based on detailed simulations with OMNeT++ showing that our technique is energy-efficient and significantly outperforms standard MAC approaches such as CSMA and TDMA.**

I. Introduction

Wireless sensor networks (WSNs) have attracted attention over the last years and are regaining importance with the upcoming trend for interconnectivity in the era of Internet of Things. The network requirements hereby are manifold and strongly depend on the application, which can be, for example, environmental monitoring, home automation, body area networks, surveillance, etc. In particular, communication needs to be reliable and delay-bounded to ensure a specific quality of service (QoS). In addition, energy efficiency is of paramount importance since nodes are usually battery powered and maintenance can be difficult and expensive, e.g., sensor nodes that are deployed in a place difficult to reach.

One of the main sources of power consumption at sensors nodes is the transceiver circuitry, i.e., sending and receiving data [1]. Although receiving data usually needs the same power as transmitting, many communication schemes, such as CSMA (carrier sense multiple access) and TDMA (time division multiple access), are based on carrier sensing or synchronization messages, which drastically increase the active time of the receiver and, therefore, lead to a high power consumption. To counteract this problem, several approaches from the literature reduce a node's active time by putting it into sleep mode, i.e., powering it down.

For example, in duty-cycling WSNs, nodes periodically wake up for a short time to listen for a request message in order to start their data transmission [2]. With scheduled sleeping, nodes are synchronized to wake up and transmit data to avoid collisions and maximize sleep times [3]. Both methods will greatly reduce energy consumption, however, they incur in increased overhead (scheduling beacons need to be sent) and long delays (wake-up preambles), which limit the number of possible applications.

Other approaches have been designed to lower these delays by using an event-triggered paradigm. For example, in [4], a modified CSMA protocol is presented using asynchronous communication that is initiated by sensor nodes without the need for any transmission request. Since nodes do not have to sense the channel periodically for preambles, they can start transmitting immediately instead of waiting for their scheduled time slot. This makes them more energy-efficient at low network loads and offers better delays. However, the lack of synchronization also leads to packet loss at high network loads and worsens both delay and energy efficiency.

In this paper, we propose a MAC technique that addresses the bad performance of classic contention-based protocols at high network loads, while still maintaining their advantages. In particular, the proposed MAC guarantees a reliable and delay-bounded communication. Due to its asynchronous design, it offers good energy efficiency at both low loads as well as in bursty traffic scenarios with high network contention. This makes it an ideal choice for a wide range of applications, such as environmental monitoring, were long idle times are common, but events can trigger multiple nodes at once leading to short bursts of high traffic.

A. Contributions

In this paper, we propose a MAC scheme that guarantees reliability requirements while fulfilling a maximum delay constraint or deadline. Similar to our previous work in [5], the presented method consists in making each node send a sequence of redundant packets with constant inter-packet times. However, in contrast to [5], where we used unidirectional nodes and a simplified network model, we now use bidirectional nodes and extend our model to increase network performance such as flexibility, delay, data rate, etc. Carrier sensing is used to detect ongoing transmissions and skip data packets if these would cause collisions. In addition, every successful transmission is acknowledged, hence, nodes can stop transmitting further packets if they receive an acknowledgment (ACK). This greatly reduces the generated traffic load resulting in a better energy consumption as well as in shorter inter-packet times and improved 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 design technique for unidirectional home-automation networks. Section V presents our experimental evaluation based on simulation and Section VI concludes the paper.

II. Related Work

There are many different approaches from the literature that are concerned with making WSNs more reliable and energy-efficient. In general, these methods can be classified in three categories: synchronous, asynchronous and hybrid networks [6].

In synchronous networks, nodes share a common clock by either periodically synchronizing their local clocks with beacon messages (time triggered) or by external events (event triggered). This common global clock helps to schedule data packets to effectively avoid collisions and allow for a high channel usage or helps scheduling sleep times and therefore minimizing energy consumption [3]. However, synchronization also results in an increased protocol overhead generally reducing the energy efficiency, especially in networks with low traffic. Examples of synchronous networks are TDMA and slotted Aloha [7].

On the other hand, asynchronous networks do not need to synchronize clocks, which is particular advantageous in event driven scenarios with a low event activity and long idle times. However, since nodes can be triggered at any time instance, collisions between packets cannot be avoided and special care must be taken to increase the communication reliability. A widely used asynchronous protocol is CSMA.

Hybrid networks try to combine the advantages of both synchronous and asynchronous WSN. This can, for example, be done by switching from asynchronous to synchronous communication in high contention [8] or by leaving space in TDMA frames, where nodes can transmit packets with CSMA [9]. Although these approaches increase the energy-efficiency with respect to TDMA, they cannot guarantee any bounded delay or fully reliable communication.

In [3], a hybrid approach called DISSense is presented, which uses synchronized sleep/awake duty cycling. By ensuring that wake up times are common to all nodes, their active times can be minimized to duty cycles as low as $0.1\,\%$. Data transmission is done in hybrid frames, in which configuration and synchronization messages are transmitted first, followed by a slot for data transfer using CSMA. Similar to many other duty/cycling MACs, DISSense is designed for periodic traffic, which means it can delay messages up to one sleep/awake cycle. Although is has the option to adapt the duty cycle according to the network characteristics, its performance in bursty traffic is at most the one of CSMA, which we later prove to be less performant than our proposed scheme.

Another hybrid approach, called Z-MAC, is presented in [8]. It uses CMSA for low network loads and dynamically switches to TDMA for higher loads depending on the packets error rate and the noise level. In contrast to regular TDMA, time slots can be accessed by any node, but the slot owner is assigned a higher priority, which is realized by scheduling it a priori to other nodes, i.e., it starts earlier to transmit. This maximizes data throughput, since unused slots can still be used, but requires additional carrier sensing, which together with the need for period synchronization, results in a bad energy efficiency in TDMA mode. However, during low network loads, its CSMA mode is still energy efficient.

In the domain of asynchronous communication, Andersson et al. [11] presented a transmission scheme for transmit-only nodes guaranteeing that data always reaches its destination within the shortest possible delay. To this end, each transmitter sends a sequence of redundant packets with carefully selected patterns such that at least one packet is not interfered by other transmitters. The transmission patterns are selected via ILP (integer linear programming) minimizing the transmission durations of all sequences of packets. Since transmit-only nodes are used, packets cannot be acknowledged nor carrier sense can be implemented. As a consequence, the average energy consumption will be high, since upon activation, always the maximum number of packets will be sent as the node cannot detect whether its packets have been received or not. Further, the missing receiver of the nodes will make the network inflexible with regard to reconfiguration, which must be done manually.

In this paper, we propose a MAC scheme that allows for a reliable communication within a maximum delay constraint or deadline. In contrast to the above approaches, our methods offers a high energy efficiency for both low and high network loads and ensures deterministic worst-case delay and reliability. This makes it suitable for a broad range of applications, such as environmental monitoring, where longs sleep times in the order of minutes to days are common. Data can be successfully transmitted, even if events can trigger multiple nodes at once resulting in short periods of high network load.

III. Models and Assumptions

We consider a WSN consisting of n sensor nodes and one or more sink nodes that are spatially distributed in an indoor or outdoor environment. Upon activation — nodes can be either triggered by events or in a periodic fashion — sensor nodes broadcast their data to their corresponding sink in a single-hop (star-topology) fashion. The sink can then either process the data or act as a cluster head and further relay it to other sink nodes.[1]

Transmitting a data packet takes a given amount of time, which depends on the number of bits to be transmitted and

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

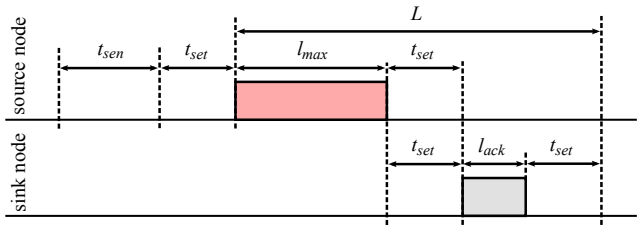

Figure 1. Timing of a successful packet transmission.

the bandwidth of the communication channel. We refer to this time to as *packet length* and denote the maximum length of any packet in the network by l_{max}.

Since nodes are triggered by independent events, they may interfere with one another leading to packet loss. As a result, to achieve reliability, a receiver or sink node acknowledges a packet after successful reception. We denote by l_{ack} the length of an acknowledgment packet. If no acknowledgment arrives for a given packet, the sender or source node retransmits the packet up to $k-1$ times within d_{max}. Here, d_{max} is the maximum delay or deadline that is allowed for a single *hop* in our WSN and k is an integer number denoting an upper bound on transmission attempts by any node in the network.

Thereby, in contrast to CSMA, nodes in our network use a fixed and constant backoff time between transmissions. For any node i, we denote this constant backoff time by p_i. A careful selection of p_i, together with k, allows providing a deterministic guarantee on the worst-case transmission delay of any packet, which is clearly not possible for CSMA.

Fig. 1 illustrates a successful packet transmission between a source and a sink node. Every time a source node needs to transmit a packet, it *senses* the communication channel for t_{sen} time. If the communication channel is free, it switches to send mode – within t_{set} time – and starts transmitting the packet. If the communication channel is being used, as discussed later in detail, the node waits for a pre-configured amount of time to re-try sending its packet. To this end, we assume that nodes can detect the transmission of all other nodes that are connected to the same sink, i.e., there are no hidden terminals within a sinks range.

On the other hand, every sink monitors the transmission channel continuously (similar to CSMA), since communication is asynchronous and packets can be received at any time. This increases the energy consumption, which can, however, be neglected for many WSN applications. For example in home automation networks, sinks are typically electric appliances, which are connected to the house's mains. In return, energy is saved at multiple sensor nodes, where this is a generally much more critical concern.

After transmitting a packet, the source node needs t_{set} time to switch to receive mode and be able to receive an acknowledgment. Note that t_{set} includes possible latencies such as processing and propagation delays. Similarly, after receiving a packet, the sink node needs t_{set} time to switch to send mode and send an acknowledgment. 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. 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 can start receiving further packets – see Fig. 1:

$$L = l_{max} + l_{ack} + 2t_{set}. \tag{1}$$

Finally, each transmitter is assumed to be activated only once within a time interval t_{max} where $d_{max} \leq t_{max}$ holds. This is a logical assumption since multiple activations of the transmitters lead to unnecessary interference. Of course, t_{max} should not lead to unacceptable delay and should be tolerable by the application.

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_{max}, k, and p_i that allow guaranteeing a reliable communication between nodes in our WSN.

As mentioned above, nodes sense the communication channel for a configurable time t_{sen} before starting to transmit a packet. However, since nodes are independent of one another, it may always happen that they are triggered such that there is still a conflict. For example, let us assume that a node i and a node j sense the communication channel for $100\,\mu$s and $10\,\mu$s, respectively, before they start transmitting. It can happen that node j is triggered $90\,\mu$s after node i such that they do not detect each other and packets still get lost.

The purpose of t_{sen} is to avoid that a node starts transmitting when another node is already sending a packet or awaiting an acknowledgment. To this end, since there is a gap equal to t_{set} between a packet transmission and its corresponding acknowledgment – see again Fig. 1, t_{sen} should be chosen such that:

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

where $\bar{t}$ is the transceiver's sensitivity, i.e., the least amount of time a signal needs to be present at the node's antenna for the transceiver to detect it. Note that the signal must be present *continuously* for detection, i.e., without any interruption. By adding $2\bar{t}$ in (2), we ensure that this is always the case, even if t_{sen} overlaps with the gap between l_{max} and l_{ack}.

With t_{sen} as per (2), a node will always detect whether another node is currently sending a packet or awaiting an acknowledgment. Note that a longer t_{sen} than (2) has no benefit, but it rather increases energy consumption of the node without guaranteeing better results. However, since there can still be collisions, let us now consider the following theorem establishing a relation between inter-packet times of two nodes.

Theorem 1. *Let us consider a WSN as defined in Section III. For any node i in the network, it can be guaranteed that at most one packet is interfered on the communication channel by another node j, if the following condition holds for $1 \leq i \leq n$, $1 \leq j \leq n$, $1 \leq l \leq k$, and $i \neq j$:*

$$\mathrm{mod}\left(\frac{l \times p_i}{p_j}\right) \geq 2(t_{set} + \bar{t}), \tag{3}$$

where t_{set} has been selected as per (2) *and is the time to switch between send and receive mode at the nodes, $\bar{t}$ is the sensitivity of the node as defined above, while p_i and p_j are the (constant) inter-packet times of node i and j respectively.*

Proof. Let us assume that any node i with $1 \leq i \leq n$ starts sending its first packet at time t_0, i.e., it is triggered by $t_0 - t_{sen} - t_{set}$. If this packet of node i is interfered by a packet of j being sent at the same time, to prove this theorem, we need to guarantee that none of the other potential transmissions by node i can be interfered anew by node j. Recall that nodes are activated only once within t_{max}, and hence, if they send more than one packet, these are due to retransmissions.

As a result, the subsequent activation times of node j for packet transmission need to be such that either node i detects node j or vice versa when they sense the communication channel. Now, if node i starts sending a packet, recall that a node j will be able to detect it, if t_{sen} is selected as per (2). From Fig. 1, note that node j will not be able to detect a node i's packet transmission, if node i starts sending less than $\bar{t}$ time before the end of node j's t_{sen} – recall that $\bar{t}$ is the least amount of time a signal needs to be present for a node to detect it. From this point in time and until the end of the following t_{set}, node j is *blind*, i.e., in an interval of length $\bar{t} + t_{set}$.

Similarly, node i is unable to detect a node j's packet transmission, if node j starts sending less than $\bar{t}$ time before the end of node i's t_{sen}. This again result in an interval of length $\bar{t}+t_{set}$ in which node i is *blind*. As a consequence, node i and j will interfere with each other at the communication channel, only if their activation times fall into an interval of length $2(\bar{t} + t_{set})$ from one another.

If node i and j interfere with each other at the communication channel, their subsequent packet transmissions can be prevented from interfering by properly selecting p_i and p_j. In particular, the activation times of node i and j need always to be separated by at least $2(\bar{t} + t_{set})$ time, which leads to (3) and the theorem follows. □

Theorem 1 allows us to guarantee that any two nodes i and j interfere only once with each other, i.e., their packets collide at most once per activation. However, it does not state how often nodes can be activated within t_{max} and how this affects collision between subsequent activations. To this end, let us consider the following analysis.

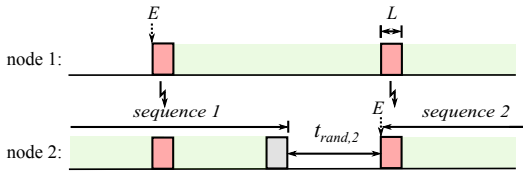

Figure 2. Illustration of Lemma 1 for the case of two nodes. Upon activation by an event E, node 1 starts transmitting its first packet, which is interfered by a packet of node 2 (reddish shading). Since node 2 finishes its sequence before the second packet of node 1 is transmitted and there is no transmission pause after a sequence, i.e., $d_{max} = t_{max}$, it can happen in the worst case that node 2 is triggered again at a time $t_{rand,2}$, such that there is a second collision with node 1.

Lemma 1. *Let us consider a WSN as defined in Section III. If $t_{max} = d_{max}$ holds, i.e., a node can immediately start transmitting anew after it finished its sequence, at least $2(n-1)$ packets of any node i will be lost in the worst case, independent of the inter-packet separation p_i with which packets are transmitted.*

Proof. Let us assume that node j is activated at time t_0 and, hence, sends up to k packets – depending on whether packets need to be retransmitted or not – within $[t_0, t_0 + d_{max}]$ with a constant inter-packet separation p_j. Let us assume that node j's last packet is sent at time $t_0 + d_{max} - l_{max}$ such that this packet is fully transmitted by $t_0 + d_{max}$. If node i starts transmitting at time $t_0 + d_{max} - l_{max}$, i.e., it was activated $t_{sen} + t_{set}$ time before – see Fig. 1 – and did not detect a transmission by node j in t_{sen}, the last packet of node j and the first packet of node i will be lost.

In the worst case, the remaining $n-2$ nodes in the network start transmitting at $t_0+d_{max}-l_{max}+l\times p_i$ where $1 \leq l \leq k$ is an integer number and p_i is node i's inter-packet separation. As a consequence, $n-1$ packets of node i will be lost independent of inter-packet times of node i, of node j, and of the other $n-2$ nodes.

Similarly, node j can interfere with further packets of node i, if it is activated anew before time $t_0 + 2d_{max} - l_{max}$, i.e., before node i finishes transmitting its k packets. Since we consider $t_{max} = d_{max}$, i.e., a node can immediately start transmitting anew after it finished its sequence, it is possible that, in the worst case, node j interferes with up to two packets from each other node, i.e., up to $2(n-1)$ packets can be lost – see Fig. 2. The lemma follows. □

As a result of Lemma 1, in the worst case, each node might need to transmit up to $2(n-1)+1 = 2n-1$ packets within d_{max}. In addition, each node should only be activated once in an interval of length t_{max} with $t_{max} = d_{max}$. Since energy consumption is proportional to the number of packets sent, it is useful to the node's battery lifetime, to decrease k by introducing a transmission pause after each sequence. This is shown in the following corollary:

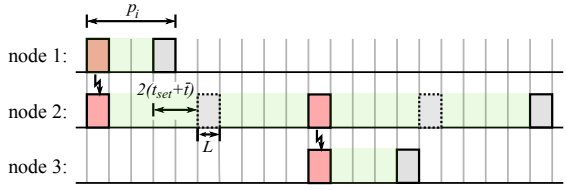

Figure 3. Illustration of Lemma 2 for the case of three nodes, i.e., $n = 3$. There can be $n - 1 = 2$ packet collisions on the communication (reddish shading). In addition, since (3) holds according to Theorem 1, nodes will be able to detect transmissions of one another. As a result, in the worst case, one node's packet transmission can be delayed additional $n - 1 = 2$ times (second node's dotted packets).

Corollary 1. *Let us consider a WSN as defined in Section III. If $t_{max} \geq 2d_{max}$ holds, i.e., a node will wait for an inter-sequence pause of at least d_{max} after each sequence before transmitting anew, at least $n - 1$ packets of any node i will be lost in the worst case, independent of the inter-packet separation p_i with which packets are transmitted.*

Proof. Immediate from Lemma 1. □

In summary, we can observe that according to Lemma 1, the missing pause after a sequence leads to an additional $n - 1$ unavoidable packet losses per sequence compared to Corollary 1. This means that, on the one hand, the missing pause allows triggering the node more frequently and, hence, theoretically increases the total data throughput. On the other hand, the energy efficiency is decreased due to a higher number of possible packet collisions in the worst case. However, we found out that the higher packet collision rate mostly dominates and finally leads to a lower data throughput that is comparable to a system with inter-sequence pauses. Since energy is a crucial factor for WSN, we therefore assume the system implements a transmission pause after each sequence as stated in Corollary 1 for the rest of the paper.

A. Selecting the number of transmissions k

With Corollary 1, we know that there are at least $n - 1$ unavoidable packet losses per sequence in the worst case. However, in order to derive *safe* values for k, we also have to account for the effects of carrier sensing, i.e., when nodes skip packets due to a blocked channel. This is analyzed in the following lemma.

Lemma 2. *Let us consider a WSN as defined in Section III. For any two nodes i and j in the network, p_i and p_j have been selected such that they comply with* (3)*. If interference from outside the network can be neglected, the following number of transmissions k suffices to guarantee reliability:*

$$k = 2n - 1. \tag{4}$$

Proof. We know from Theorem 1 that any two nodes i and j in the network only interfere once with each other, if p_i and p_j are selected as per (3). This means that, in the worst case, a packet of a node i can be interfered $n-1$ times by the remaining $n-1$ nodes. On the other hand, if p_i and p_j comply with (3), they will be able to detect transmissions of one another. However, in the worst case, a packet of a node i can also be delayed $n-1$ times by transmissions of the remaining $n-1$ nodes. The maximum number of retransmission attempts is hence $2(n-1)$, i.e., in the worst case, a node may try to send a packet a total number $k = 2(n-1)+1 = 2n-1$ times before being successful – Fig. 3 illustrates this for $n = 3$. The lemma follows. □

B. Deriving inter-packet time boundaries

In the following, we are concerned with finding *safe* values for p_i for any i and $1 \leq i \leq n$. To this end, let us first consider the following lemma guaranteeing at most 1 collision between any two nodes.

Lemma 3. *Let us consider a WSN as defined in Section III. In order to guarantee that at most one packet of a node i is interfered on the communication channel by another node $i-1$, the following condition must hold for any p_i and p_{i-1} where $1 < i \leq n$:*

$$p_i - p_{i-1} \geq 2(t_{set} + \bar{t}), \tag{5}$$

given that $p_{i-1} < p_i < 2p_{i-1}$ holds, i.e., $\lfloor \frac{p_i}{p_{i-1}} \rfloor = 1$.

Proof. According to Theorem 1, if (3) holds for all i and j where $i \neq j$ and $1 \leq l \leq k$, it can be guaranteed that any two nodes i and j interfere only once with each other.

Now, for a any i, $j = i - 1$, and $l = 1$, from (3) we have $\mod\left(\frac{p_i}{p_{i-1}}\right) \geq 2(t_{set} + \bar{t})$, which again has to hold according to Theorem 1. Since $p_i > p_{i-1}$ and $\lfloor \frac{p_i}{p_{i-1}} \rfloor = 1$ hold for $1 < i \leq n$, we have that $p_i - p_{i-1} \geq 2(t_{set} + \bar{t})$ and the lemma follows. □

Lemma 4. *Let us consider a WSN as defined in Section III. If one packet of a node i is interfered by a packet of node j, in order to guarantee that the next packet sent by node i is not interfered again by node j, the following condition must hold for the minimum inter-packet time $p_{min} = \min_{1 \leq i \leq n}(p_i)$:*

$$p_{min} \geq L + t_{set} + t_{sen}. \tag{6}$$

Proof. Let us assume that node i is triggered at time t_0 and successfully sends its data to the corresponding sink. This means that the node is busy until $t_0 + t_{sen} + t_{set} + L$ – see Fig. 1. If the next packet of node i starts directly afterwards, i.e., without any further waiting time, the waited period time is $p_i = L + t_{set} + t_{sen}$, as stated in (6).

Let us again assume node i is triggered at time t_0, but this time its first packet is interfered by a packet of node j. According to Theorem 1 that node j was triggered in a time interval $[t_0 - (t_{set} + \bar{t}), t_0 + (t_{set} + \bar{t})]$. This means that after the (collided) packet of node i is transmitted, node j can still occupy the channel for up to $t_{set} + \bar{t}$ time until $t_0 + t_{sen} +$

$t_{set} + l_{max} + (t_{set} + \bar{t})$ – see Fig. 1. Both nodes will now listen for acknowledgments, which are not sent since packets were corrupted. This takes additional $2t_{set} + l_{ack}$ time, which is always greater than the maximum possible delay $(t_{set} + \bar{t})$ of node j. As a consequence, the earliest point in time when node i can start with its second packet is $t_0 + L + t_{set} + t_{sen}$, which results in a p_i equivalent to (6). Due to (5), there will be no further collision with node j. The lemma follows. □

Lemma 5. *Let us consider a WSN as defined in Section III. The (constant) inter-packet time is upper bounded by* $p_{max} = \max_{1 \le i \le n}(p_i)$*:*

$$p_{max} \le \frac{d_{max} - (t_{sen} + t_{set} + L)}{k}. \quad (7)$$

Proof. Without loss of generality, let us assume that node i is activated at time t_0. In order that n packets can be sent within $[t_0, t_0 + d_{max}]$, the n-th packet has to start at latest at $t_0 + d_{max} - (t_{sen} + t_{set} + L)$. This way, the sink node finished switching back to receive mode after acknowledging node i's n-th packet exactly at $t_0 + d_{max}$. □

C. Calculating inter-packet times

As we have investigated the upper and lower bound of inter-packet times in the previous analysis, we can now derive a formula to find *safe* values for p_i for any i and $1 \le i \le n$.

Theorem 2. *Let us consider a WSN as defined in Section III. If the first packet of a node i is interfered by a packet of node j, in order to guarantee that the next $(2n-2)$ packets sent by node i are not interfered again by node j, the following condition must hold for the minimum inter-packet time* $p_{min} = \min_{1 \le i \le n}(p_i)$*:*

$$p_{min} \ge (2n-2) \times (n-1) \times 2(t_{set} + \bar{t}) + 2(t_{set} + \bar{t}), \quad (8)$$

where as before $p_{min} < p_i < 2 \times p_{min}$ holds, i.e., $\lfloor \frac{p_i}{p_{min}} \rfloor = 1$, for $1 \le i \le n$. In addition, $\lfloor \frac{k \times p_{min}}{(k-1) \times p_{max}} \rfloor = 1$ also holds, i.e., $(k-1) \times p_{min} < (k-1) \times p_{max} < k \times p_{min}$, for $1 < k \le n-1$ and $p_{max} = \max_{1 \le i \le n}(p_i)$.

Proof. Let us again assume that the first packet sent by a node i is interfered at time t_0 by a packet of node j. In order that the next $(2n-2)$ packets sent by node i are not interfered again by node j, (3) needs to hold for all i and j where $i \ne j$ and $1 \le k_i \le n-1$ as per Theorem 1.

Without loss of generality, let us assume that all p_i are sorted in order of increasing values, i.e., $p_i > p_j$ if $i > j$. Hence $p_{min} = \min_{1 \le i \le n}(p_i) = p_1$ and $p_{max} = \max_{1 \le i \le n}(p_i) = p_n$ hold.

Let us first consider $i = 1$ and $j = n$. If $l = 1$ holds, from (3) we have that $\mod\left(\frac{p_1}{p_n}\right) \ge 2(t_{set} + \bar{t})$ is equal to $p_1 \ge 2(t_{set} + \bar{t})$ since $p_1 < p_n$. For $l = 2$, from (3) we have that $\mod\left(\frac{2 \times p_1}{p_n}\right) \ge (t_{set} + \bar{t})$ is equal to $2 \times p_1 - p_n = p_1 - 2 \times (n-1) \times (t_{set} + \bar{t})$, as $\lfloor \frac{2 \times p_1}{p_n} \rfloor = 1$ holds – see again proof of Lemma 3. Similarly, for $l = 3$, we have that $\mod\left(\frac{3 \times p_1}{p_n}\right) \ge (t_{set} + \bar{t})$ is equal to $3 \times p_1 - 2 \times p_n = p_1 - 2 \times 2 \times (n-1) \times (t_{set} + \bar{t}) \ge 2(t_{set} + \bar{t})$, as $\lfloor \frac{3 \times p_1}{2 \times p_n} \rfloor = 1$ holds. For $l = 2n-1$, we have that $\mod\left(\frac{(2n-1) \times p_1}{p_n}\right) \ge 2(t_{set} + \bar{t})$ is equal to $(2n-1) \times p_1 - (2n-2) \times p_n = p_1 - (2n-2) \times 2 \times (n-1) \times (t_{set} + \bar{t}) \ge 2(t_{set} + \bar{t})$, as $\lfloor \frac{(2n-1) \times p_1}{(2n-2) \times p_n} \rfloor = 1$ also holds. As a result, we have that $p_1 \ge (2n-2) \times (n-1) \times 2(t_{set} + \bar{t}) + 2(t_{set} + \bar{t})$ which is lower bound for $p_{min} = p_1$ stated in (7). Since $p_n = p_{max}$, note that choosing another j where $1 < j < n$ yields a lower bound that is closer to that of Lemma 3. In other words, the lower bound of (7) is the greatest necessary value of p_{min}. The theorem follows. □

In summary, both Lemma 4 and Theorem 2 provide a lower bound on p_{min} for the case that $p_{min} < p_i < 2 \times p_{min}$ where $1 \le i \le n$. However, in contrast to Lemma 4, the lower bound of Theorem 2 guarantees that, if a packet of node i gets interfered by any node j, its next $2n-2$ packets will not be interfered again by node j. This result, together with Lemma 3, allows us to design a reliable communication network, since we can guarantee that at least one packet of each node reaches its receiver in the worst case.

V. Simulation and Evaluation

In this section, we present the results of a simulation based on the OMNeT++ network simulation framework [12] and an extension for mobile and wireless networks named MiXiM [13]. This allows us to effectively simulate our network with different physical parameters and to record statistical values for very large numbers of transmissions.

The simulated network consists of one receiver and a selectable number of n transmitters that are all within range of one another and, hence, interfere with one another. The receiver node is a simple data sink, whereas transmitter nodes are data sources that transmit packets with a certain pattern according to the compared MACs as explained below. Note that the *proposed* MAC also supports multihop communication via sink nodes, as discussed in Section III. However, for simplicity, a single hop and single sink setting is used.

All transmitter nodes run independently of one another and are triggered by random time events to ensure that different possible combinations of packet transmissions are considered. Recording and processing of simulation data is done by the framework at runtime. In particular, the time stamps of the different packets sent are compared to determine whether packets overlap and, hence, get lost.

We consider the case of high congestion, i.e., nodes are triggered as frequently as possible so as to simulate conditions close to the worst case. This gives us an upper bound of the simulated values, which means that under normal operation

(less congestion), nodes are expected to yield better results. Each simulation was performed for different numbers of nodes n, for which 100,000 different packet sequences have been simulated each time.

The data rate or bandwidth of transmission has been fixed to $128\,kbit/s$, which is typical in $433\,MHz$ or $868\,MHz$ home automation settings. The packet size is set to $32\,bytes$ ($4\,bytes$ preamble, $26\,bytes$ data, and $2\,bytes$ check sum), resulting in a transmission time of $2000\,\mu s$, i.e., this is the value of l_{max}. The length of the acknowledgment is set to $5\,bytes$, i.e., it contains a $4\,bytes$ preamble and the identifier of the received packet, which is sufficient for the node to recognize it as an acknowledgment. Its transmission duration is $l_{ack} = 312.5\,\mu s$. The time for switching from receiver to transmitter mode and vice versa[2] has been set to $30\,\mu s$, which together with a processing delay of $60\,\mu s$ results in $t_{set} = 90\,\mu s$. And finally, the minimum detection time $\bar{t}$ is set to $\bar{t} = 10\,\mu s$, i.e., the time to detect one bit including some tolerance.

We consider the following three packet transmission schemes and compare them in the simulation:

- The *proposed* scheme is our transmission scheme as presented in Section IV.
- The *TDMA* scheme is based on the Time Division Multiple Access method. In contrast to the other transmission schemes, all nodes in the system share a common clock by periodically receiving synchronization beacons from the sink node.
- The *CSMA* scheme is based on non-persistent Carrier Sense Multiple Access method. Non-persistent means that the channel is not sensed continuously during backoffs, but only for a short time before each transmission to reduce energy consumption.

Note that baseline *CSMA* and *TDMA* were chosen, since these are the core technologies on which most other relevant approaches are based [3] [8] [9]. In particular, note that [3] reduces to *CSMA* and [8] [9] to *TDMA* at high network load. For more details see Section II.

Each time slot of the *TDMA* scheme has the length of a packet transmission plus an additional 10% safety margin to tolerate slight clock drifting. The synchronization beacon is set to a length of $4\,bytes$ ($275\,\mu s$ with safety margin) and we assume that all nodes use a standard crystal oscillator with $100\,ppm = 0.01\,\%$ accuracy. As a consequence, the synchronization interval is set to $1\,s$ resulting in a maximum time error of $\pm 100\,\mu s$, which is exactly the previously mentioned $10\,\%$ safety margin of a time slot. The total number of slots is fixed to the current n of each iteration of the following experiments.

In case of the *CSMA* scheme, the timings of the IEEE 802.15.4 MAC layer [15] were used and adapted to match the different transmission speed and packet sizes in our network setting. To this end, the slot size was set to $320\,\mu s$, the maximum number of backoff retries to 4 and the maximum retransmissions number to $k = 7$. The *contention window* was set to $[32, 1024]$ slots, similar to 802.11, to allow for a high number of contending nodes. Further, no RTS/CTS scheme was used to keep the protocol overhead low and to be comparable to our *proposed* scheme and *TDMA*.

[2]The parameters were taken from CC110L Value Line Transceiver [14].

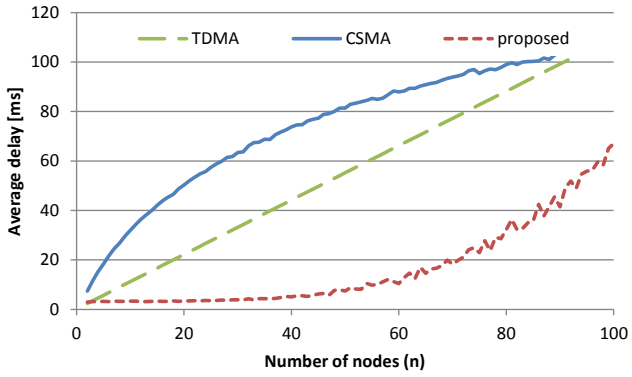

Figure 4. The average delay of the three transmission schemes, i.e., the time from triggering a node until the successful reception of its data, is depicted.

A. Communication Delay

Let us first analyze the average delay, i.e., the time from triggering a node until the successful reception of its data. As depicted in Fig.4, it can be seen that for all three transmission schemes, the average delay rises for an increasing number of n. Clearly, for the *proposed* and *CSMA* scheme, a higher n leads to a higher network load resulting in a higher collision rate and, hence, in a higher average delay. This delay rises linearly for *CSMA* and quadratically for the *proposed* scheme, since periods found by (8) show cubic growth, i.e., they also rise quadratically for higher n. In case of *TDMA*, a higher n implies more slots within a frame and, hence, the average delay rises in a linear fashion.

As we can see in Fig.4, our *proposed* scheme offers the lowest average delay for $n \leq 100$. This delay, however, will be higher than for the other two schemes for very high n due to its cubic growth. In addition, the worst-case delay rises with the power of 3. Other MACs might consequently be suited better for networks with very high n depending on the maximum tolerable delay.

B. Energy Consumption

Next, let us examine the long term energy consumption per node as displayed in Fig. 5. To this end, we simulated an exemplary network running for 1 year in which each node was triggered 1000 times per day on average. Again, traffic is considered to be bursty, i.e., we assume that events trigger multiple nodes simultaneously leading to short bursts of high network load. This will result in an increased energy consumption, which can be seen as an upper bound, i.e., during normal operation, it will usually be lower.

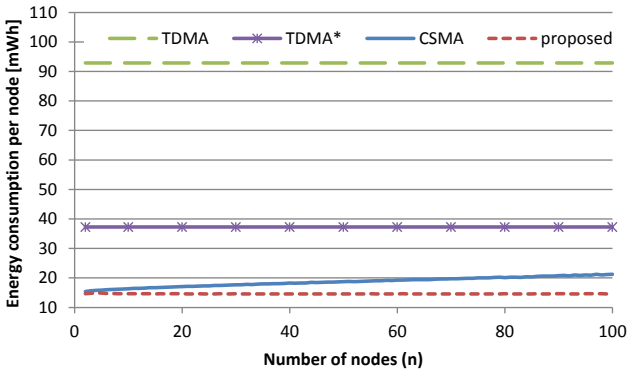

Figure 5. The average energy consumption per node of a 1 year deployment with an activity rate of 1000 sequences per day per node.

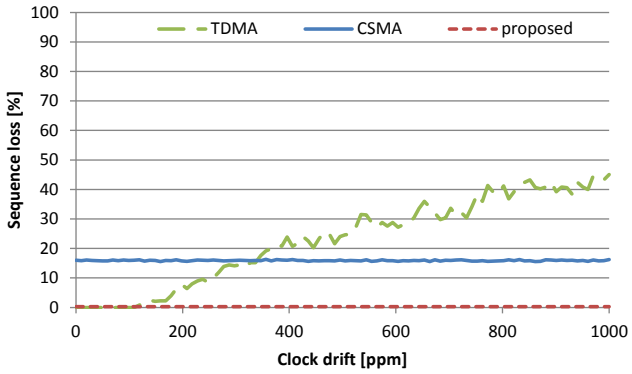

Figure 6. The average sequence loss, i.e., when all packets of a transmission are lost, is depicted with respect to an increasing clock drift.

In order to calculate the energy consumption, we recorded the times each node actively spent in receive and transmit mode and multiplied them with the corresponding power levels from the CC110L transceiver. In receive mode, 33mW is consumed, whereas in transmit mode each node consumes 66mW (+12dBm output power at $V_{DD} = 2\,V$). In the remaining time, each node is assumed to be in sleep mode and not to consume any energy.

It can be seen that both the *proposed* and *CSMA* scheme offer low energy consumptions, as their asynchronous designs do not need synchronization and, hence, offer good efficiency. In case of the *CSMA* scheme, the energy consumption slowly increases with rising n, as more collisions occur and more retransmissions are necessary.

In contrast, the *TDMA* scheme results in high energy consumption per node, mostly because of its synchronization overhead, i.e., when nodes wake up periodically to receive beacon messages and adjust their internal clocks. Since packets have to be transmitted only once, i.e., there are no collisions within the network, the transmit time per cycle stays constant for an increasing n. On the other hand, the receive time is determined by the frequency and length of the beacon messages. These only depend on the precision of the oscillator and the slot tolerance and are therefore independent of n. As a consequence, both receive and transmit time stay constant for rising n and so does the energy consumption.

TDMA's additional energy consumption for transmitting (with respect to the proposed approach) basically results from its synchronization overhead. However, this can be reduced by using precision oscillators, as displayed by the *TDMA** curve in Fig. 5. Here, the energy consumption can be decreased to around $40\,\%$ by using oscillators with $30\,ppm$ instead of $100\,ppm$. In addition, special synchronization protocols such as FTSP [16] implement techniques to estimate and compensate clock drift. Although these methods can greatly reduce the synchronization overhead, the total energy consumption will in general still be higher than for asynchronous protocols, especially in networks with long idle times or low traffic load.

C. Clock Drift

All electronic devices show deviations in their electric characteristics, which depend on a number of different factors such fabrication-induced variability, operating temperature, etc. This deviation also affects the nodes' oscillators, consequently, each node will count time at different rates possibly leading to unexpected system behavior. To demonstrate these effects, we performed a simulation with an exemplary network of $n = 20$ nodes and measure how clock drift affects the number of lost sequences, i.e., when all data packets of a sequence are lost.

As depicted in Fig. 6, we can see that both the *proposed* and *CSMA* scheme are barely affected by clock drifts $< 1000\,ppm$. This is because their asynchronous design, together with a high packet number and carrier sensing, makes them inherently robust against timing errors. Similarly, simulations also showed that other performance parameters such as energy consumption, delay, etc., are, on average, neither affected by clock drift. In summary, we can therefore safely neglect clock drift in most applications, since even cheap crystal oscillators feature an accuracy $< 100\,ppm$.

In contrast, the synchronous *TDMA* is sensitive to timing errors. Nodes may possibly violate their time slot boundaries and collide with other data packets or even with synchronization beacons. As we can see in Fig. 6, sequence loss slowly starts from $100\,ppm$, i.e., the maximum allowed drift that we accounted for when designing the network parameters, and increases for higher values until starting to saturate.

D. External Interference

External interference can occur, for example, when microwaves, wireless toys, etc. are turned on, or when there exist neighboring WSNs that have not been regarded during the design phase. To this end, we simulated an exemplary network with $n = 20$ nodes and slowly increased the duty cycle, i.e., the ratio of time interference is present and absent on the channel, from zero (no interference) to $100\,\%$ (blocked

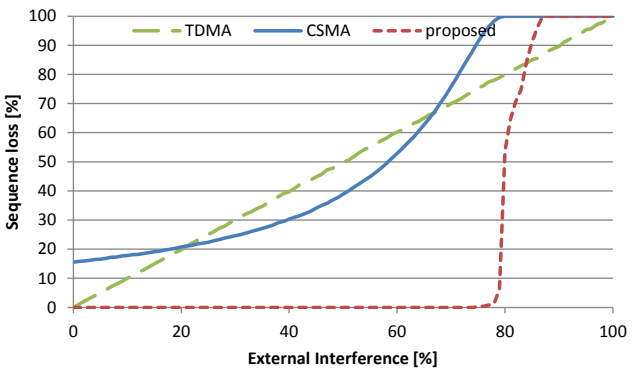

Figure 7. The average sequence loss, i.e., when all packets of a transmission are lost, is depicted for an increasing level of external interference.

channel). The inference pulses were randomly selected between $0.3\,ms$ to $10\,ms$ assuming that any overlapping with ongoing transmissions leads to corrupt data (SINR $< 0\,dB$). The results are displayed in Fig. 7.

As expected, external interference decreases the performance, such as latency, energy consumption, sequence loss, etc., of all three transmission schemes. In case of *TDMA*, no carrier sense and acknowledgment are implemented, hence, data packets are especially sensitive to interference. Further, loosing synchronization beacons results in additional clock drift errors. Since packets are not retransmitted, the sequence loss rises linearly with a higher interference level.

On the other hand, *CSMA* and the *proposed* scheme can effectively prevent interference with carrier sensing and, hence, allow for good performance even in noisy environments with up to $30\,\%$ external interference. For *CSMA*, however, the loss rate of packet sequences, i.e., when all retransmissions are lost, starts to rapidly increase at higher noise levels $> 50\,\%$. This is because the maximum number of packets as well as the limited backoff retries (i.e., contention window) start not to be sufficient. For the *proposed* scheme, this is reached for even higher noise levels, since there are considerably less packet collisions with other nodes.

VI. Concluding Remarks

In this paper, we proposed a MAC technique for designing reliable and delay-bounded WSNs. More specifically, each node sends a sequence of redundant packets with constant inter-packet times. We showed that by carefully selecting network parameters, it is possible to guarantee that at least one packet of each node reaches its corresponding receiver on time. Due to its asynchronous design and reduced listening times, the proposed scheme offers a very good energy consumption during low network load as well as reliable communication during high loads. It is therefore suitable for a broad range of applications.

By performing a large set of experiments based on detailed simulations on OMNeT++, we showed that the proposed technique never leads to packet losses (within the network) and proved its intrinsic robustness against external interference and clock drift. Further, the experiments showed that our technique is energy-efficient and significantly outperforms approaches that use conventional MACs such as CSMA and TDMA.

References

[1] Z. Cheng, M. Perillo, and W. Heinzelman, "General Network Lifetime and Cost Models for Evaluating Sensor Network Deployment Strategies," *IEEE Transactions on Mobile Computing*, vol. 7, pp. 484–497, 2008.

[2] F. Wang and J. Liu, "Duty-Cycle-Aware Broadcast in Wireless Sensor Networks," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2009.

[3] U. Colesanti, S. Santini, and A. Vitaletti, "DISSense: An Adaptive Ultralow-power Communication Protocol for Wireless Sensor Networks," in *Proceedings of the International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, 2011.

[4] 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.

[5] P. Parsch, A. Masrur, and W. Hardt, "Designing Reliable Home-Automation Networks based on Unidirectional Nodes," in *Proceedings of the IEEE International Symposium on Industrial Embedded Systems (SIES)*, 2014.

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

[7] L. G. Roberts, "Aloha Packet System With and Without Slots and Capture," *Computer Communication Review (SIGCOMM)*, vol. 5, no. 2, pp. 28–42, 1975.

[8] I. Rhee, A. Warrier, M. Aia, J. Min, and M. Sichitiu, "Z-MAC: A Hybrid MAC for Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 16, pp. 511–524, 2008.

[9] J. Afonso, L. Rocha, H. Silva, and J. Correia, "MAC Protocol for Low-Power Real-Time Wireless Sensing and Actuation," in *Proceedings of the IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2006.

[10] V. Salmani and P. H. Chou, "Bin-MAC: A Hybrid MAC for Ultra-Compact Wireless Sensor Nodes," in *Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2012.

[11] B. Andersson, N. Pereira, and E. Tovar, "Delay-Bounded Medium Access for Unidirectional Wireless Links," in *Proceedings of the International Conference on Real-Time Networks and Systems (RTNS)*, 2007.

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

[13] 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.

[14] *Datasheet: CC110L Value Line Transceiver*, Texas Instruments, 2014.

[15] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "Performance Evaluation of the IEEE 802.15.4 MAC for Low-Rate Low-Power Wireless Networks," in *Proceedings of the IEEE International Conference on Performance, Computing, and Communications (IPCCC)*, 2004.

[16] 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.