A Reliability-Aware Medium Access Control for Unidirectional Time-Constrained WSNs

Philip Parsch and Alejandro Masrur
Department of Computer Science
TU Chemnitz, Germany
philip.parsch@cs.tu-chemnitz.de

ABSTRACT

Wireless sensor networks (WSNs) are gaining in importance with an increasing need for interconnectivity in the advent of Internet of Things. A WSN typically consists of bidirectional nodes that are able to transmit and receive data. However, in applications such as home automation and body area networks, data needs to be conveyed in one direction, i.e., from sensors to a sink, in a single-hop network. Hence, unidirectional nodes can be used instead reducing costs in a considerable manner. Since unidirectional nodes are unable to acknowledge or retransmit packets, the resulting networks are strongly unreliable. To overcome this problem, we propose a medium access control (MAC) technique that can be configured to meet desired reliability requirements while fulfilling a maximum delay constraint or deadline. Our technique is based on a probabilistic analysis of packet losses in the worst case and allows, in contrast to other approaches from the literature, a more energy-efficient design. In order to evaluate the proposed technique, we present a large set of experiments and detailed simulations based on OMNeT++.

1. INTRODUCTION

Wireless sensor networks (WSNs) can be found in multiple application domains such as home automation, body area networks, environmental monitoring, surveillance, etc. These normally substitute wired solutions such as field buses wherever the latter are not viable due to technical or economic limitations.

WSNs rely on either unidirectional or bidirectional nodes. Unidirectional nodes can only transmit or receive data, while bidirectional nodes are capable of both transmitting and receiving. Clearly, in a unidirectional WSN, data packets cannot be acknowledged or retransmitted making it difficult to implement reliable communication. Bidirectional nodes, on the other hand, can implement these features making more reliable, multi-hop communication possible. However, this also leads to a higher computational cost (to implement the more sophisticated communication protocols), which, together with a more complex transceiver circuitry, makes bidirectional WSNs be considerably more expensive than unidirectional ones, in particular, as the number of nodes in the network increases.

In contrast, unidirectional nodes are of low-cost and have a higher energy efficiency compared to bidirectional nodes [3], since they do not need to power a receiver and monitor the communication channel. Considering that the overall costs as well as energy consumption are key factors when designing WSNs, unidirectional nodes have been used in various scenarios in the past: home automation [8], environmental monitoring [4], RFID [7], etc., where potentially hundreds of sensor nodes report data to a sink (either in a periodic fashion or upon the occurrence of specific events) without need for external control [15].

As a result, if the application under consideration tolerates it, unidirectional nodes allow for a more cost-effective WSN. However, since a unidirectional WSN may incur in packet loss, special medium access control (MAC) protocols have to be used to improve reliability. Unfortunately, existing solutions like CSMA (Carrier Sense Multiple Access), TDMA (Time Division Multiple Access) or slotted Aloha [11] cannot be applied, since they rely on carrier sensing or synchronization (which unidirectional nodes are unable to perform).

A number of approaches have been proposed to improve the reliability of unidirectional communication. In [10], for example, a method is presented to recover the packet with the highest signal strength when packets collide at the communication channel. In [3] [15], a mix of unidirectional and bidirectional nodes is used instead. Although these methods improve the average performance of a (mixed or fully) unidirectional WSN, they do not allow for guarantees on the resulting communication reliability.

In [1] [8], sending a sequence of redundant packets has been proposed. Here, by choosing suitable inter-packet times and neglecting external interference, it is possible to achieve a unidirectional WSN that is fully reliable, i.e., where at least one packet of a sequence of redundant packets is guaranteed to arrive within a specified deadline. However, the techniques in [1] and [8] incur great pessimism, which often results in more packets being sent than really necessary and, consequently, in considerably more energy consumption.

Contributions. In this paper, we propose a MAC scheme that also consists in each transmit-only node sending a sequence of redundant packets. In contrast to [1] and [8], our technique does not fix inter-packet times at design time, but these are rather randomly selected between a lower t_{min} and an upper bound t_{max} at runtime, i.e., every time a packet

RTNS 2015, November 04-06, 2015, Lille, France

DOI: http://dx.doi.org/10.1145/2834848.2834876

is sent. This allows applying well-known probabilistic techniques to quantify reliability. The resulting scheme is suitable for soft real-time application where a quality of service (QoS) needs to be guaranteed. We show that our approach leads to considerably less redundant packets and, hence, an improved energy efficiency. Our contributions can be summarized as follows:

- Based on random inter-packet times, we identify the conditions leading to the worst-case interference on the communication channel. We then compute the probability that, in the worst case, one packet is interfered by another packet being transmitted simultaneously. Further, we show that this probability is independent of the node sending and of the packet being sent.
- We model the transmission of a sequence of redundant packets using a binomial distribution. With this, we compute the probability that at least one packet of a sequence of redundant packets reaches its destination in the worst case. This probability is then used to assess reliability of the unidirectional WSN.
- A design/configuration method is introduced for the proposed MAC. This allows guaranteeing a desired reliability requirement in unidirectional WSNs. As explained later, this method results in appropriate values of t_{min} and t_{max}, i.e., the lower and the upper bound for selecting inter-packet times. If the reliability requirement cannot be met in a given constellation, t_{min} will be either negative or greater than t_{max}.
- Finally, we extend the proposed scheme to consider practical factors such as clock drift at the different nodes and interference from outside the network. For the latter, we derive an external interference model.

The rest of this paper is structured as follows. Related work is discussed in Section 2. Next Section 3 explains our system model and assumptions. Section 4 introduces the proposed MAC technique that allows quantifying reliability of unidirectional WSN. In Section 5, we extend our analysis to consider clock drift and external interference. Section 6 and Section 7 respectively present a numerical and a simulation-based evaluation based on OMNet++. Section 8 concludes the paper.

2. RELATED WORK

Most existing WSNs use expensive nodes with complex transceiver circuitry to counteract unreliability. These nodes implement elaborate protocols and transmission mechanisms such as multi-hopping, automatic retransmission, and routing of data packets. However, in scenarios where simplicity is required and data loss can be tolerated to some extent, unidirectional single-hop WSNs have been used many times in the past: long range outdoor networks [4], wireless sensor body networks [5], indoor networks [12] and RFID systems [16] [7].

Unidirectional WSNs need to be designed carefully to improve reliability, energy efficiency, and to guarantee a bounded delay. One such approach in the domain of ultra-wideband (UWB) networks, presented by Radunović et al. [10], uses networks that consist of three different devices: a high number of low-cost, transmit-only Sensor Nodes (SNs), bidirectional Cluster Heads (CHs) and one or more central servers. The SNs are used to sense the environment and transmit the resulting data to the CHs, which then forward it to servers for further processing. In order to improve robustness against packet collisions between nearby SNs, the CHs use a configurable receiver that only *collects* data packets complying with a pre-specified signal strength. This ensures that the data packet with the strongest signal (e.g., arriving from node in the closest vicinity of the CH) can be received at the event of a collision where otherwise it would be lost. However, if many CHs are needed for a good coverage in a network, costs will increase rapidly. Moreover, this scheme cannot guarantee a given reliability requirement, since packets may potentially never reach their receivers due to collisions, particularly, within one cluster.

In the domain of ultra-wideband (UWB) networks, Weißenhorn and Hirt [14] proposed a probabilistic approach for quantifying the collision probability of periodic data packets. In particular, only the collision probability of individual data packets is considered. In this paper, we follow a similar strategy; however, we are concerned with the probability of loosing a given number of consecutive packets — and not individual packets — to assess reliability of a unidirectional WSN. Towards this, we model the transmission of packets using a binomial distribution which further simplifies the proposed analysis as described in the next sections.

Assuming that external interference is negligible, we proposed in [8] an approach for fully reliable communication based on unidirectional devices. This consists in each node sending a sequence of redundant packets upon activation with constant inter-packet time. For each node in the network, its inter-packet time has to be selected to ensure that at least one of its packet arrives in the worst case. However, a large number of redundant data packets have to be sent for 100% reliability. This leads to low energy efficiency and prolonged delays being impractical for large WSNs.

The approach by Andersson et al. [1] [2] follows a similar principle. Here inter-packet times are found by using an ILP (Integer Linear Programming) solver and optimized for the shortest total transmission duration. However, in contrast to [8] where all nodes experience a similar communication delay, the resulting packet sequences greatly differ in length. Hence, some transmitters send their packets very fast, whereas others have delays that are multiple times longer. Similar to [8], this approach assumes that there is no interference from outside the network and requires a large number of packets to be sent in order to guarantee 100% reliability for the unidirectional WSN.

As stated above, the approach in this paper is also based on sending a sequence of redundant packets; however, in contrast to the state of the art, our technique allows reducing the number of redundant packets to guarantee a given reliability requirement. Although the proposed technique cannot reach 100% reliability, it allows reducing the overall energy consumption and is more suitable for soft real-time application where QoS requirements need to be met.

3. SYSTEM MODEL AND ASSUMPTIONS

We consider a WSN of simple transmit- and receive-only nodes that are spatially distributed as depicted in Fig. 1. A transmit-only node is *activated* by an event, for example, a user turning on the light in the context of home automation, or a sensor being triggered in a body area network, etc. Upon activation, a transmit-only node broadcasts its

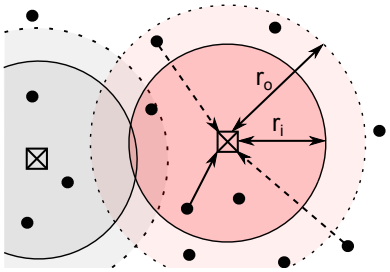

Figure 1: Example of a unidirectional WSN with transmit-only (solid circles) and receive-only nodes (checked boxes): r_i represents the range within which the receiver collects packets, while r_o indicates the range in which transmitters can interfere with each other.

data within a certain range and, hence, every receive-only node in range will process this data accordingly. If data received is not required by a given receiver, this will discard it immediately after reception.

Since transmit-only nodes are unable to detect whether data has reached its corresponding receivers, they transmit a sequence of $k \in \mathbb{N}_{>0}$ redundant packets — to increase chances of a successful reception — where k is a natural number greater than zero and a design parameter common to all nodes in the system. We consider that each node i waits a random time $t_{ij} \in \mathbb{R}_{>0}$ before sending any packet — including the first packet of a sequence.[1] Here $1 \leq i \leq n$ and $1 \leq j \leq k$ hold and $n \in \mathbb{N}_{>0}$ denotes the number of transmit-only nodes in the system. This t_{ij} is referred to as (random) inter-packet time and is uniformly distributed in the interval $[t_{min}, t_{max}]$ where $t_{min}, t_{max} \in \mathbb{R}_{>0}$ are also system design parameters, i.e., these are common to all nodes.

Packets transmitted by nodes consist of an identifier, a data, and a check sum or CRC (Cyclic Redundancy Check) field. Usually, these have a relatively constant size consisting of a few bytes. Transmitting one such packet takes an amount of time that depends on the number of bits to be transmitted and the bandwidth of the communication channel. We refer to this time to as *packet length*. In this paper, we denote by $l_{max} \in \mathbb{R}_{>0}$ the maximum length of any packet in the network. We assume that even the smallest overlapping between any two packets on the communication channel produces interference and, hence, packet loss.

We further assume that each receive-only node constantly monitors the communication channel and, hence, no special measures have to be taken before sending data. For example, no preamble needs to be sent to wake up a receiver, etc. In many applications, this assumptions does not pose any additional restrictions, since receive-only nodes are usually attached to unrestricted power supply, e.g., a lamp in a home automation setting is attached to the electric network. In other cases, where power supply is restricted, the presented method can be extended to account for preambles, e.g., by adjusting the packet length or increasing the number of packets sent. However, this is out of scope in this paper.

For each transmit-only node in the network, it must be guaranteed that one packet arrives within a deadline measured from the node's activation time. In the context of home automation, typical deadlines are around 500ms. This is, for example, the time by which a wireless light switch should turn on the light, or a motion sensor should detect the presence of a person. A greater delay is often unacceptable, since it negatively impacts the quality of the system. Similar deadlines are also common in body area networks. In this paper, we consider that this deadline is the same for all nodes and denoted by $d_{max} \in \mathbb{R}_{>0}$ — this is typical from single-hop WSNs where data needs to be conveyed in one direction and within a given time upper bound.

Finally, each transmit-only node is assumed to be activated only once within a time interval of length d_{max}. In our previous example, this means that the wireless light switch sends only one sequence of packets every 500ms. This is a logical design assumption, since multiple activations of the transmitters lead to unnecessary interference and do not help achieving the design goal of the system.

[1]This design decision simplifies our analysis in a considerable manner, while it does not affect the functionality of the unidirectional WSN.

4. PROPOSED MAC TECHNIQUE

In this section we introduce our reliability-aware MAC technique for unidirectional single-hop WSNs. As already mentioned, this consists in making each transmit-only node send a sequence of k consecutive, redundant packets upon activation. Thereby, inter-packet separations are chosen randomly. As a result, we can use probabilistic methods for assessing reliability of the resulting WSN. The reliability of a unidirectional single-hop WSN is defined as the probability that at least one packet of each transmit-only node reaches its destination in the worst case. The higher this probability is, the more reliable our WSN will be.

4.1 Reliability of a Unidirectional WSN

A packet can be lost as a consequence of interference in the communication channel. For ease of exposition, we first assume that interference originates from simultaneous transmissions by neighboring nodes with overlapping space and frequency ranges (see r_o in Fig. 1). Later in Section 5, we extend our analysis to consider external interference.

Definition: We define *reliability* of a unidirectional WSN as the probability that, in the worst case, at least one of a sequence of k packets of any node i reaches its destination within a specified deadline.

To compute the above probability, we need to consider the worst-case transmission conditions in the WSN: (i) all n nodes in the network are sending packets, and (ii) every time a packet is sent by a node, there exists a maximum fraction of the interval $[t_{min}, t_{max}]$, denoted by Δ_{coll}, for which any selected value of t_{ij} lead to collisions. While condition (i) is straight forward, condition (ii) requires more analysis.

Recall that each node in the network uniformly selects inter-packet times t_{ij} in $[t_{min}, t_{max}]$. Let us first 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 — note t_{min} is the minimum and t_{max} the maximum possible separation between two consecutive packets of a node:

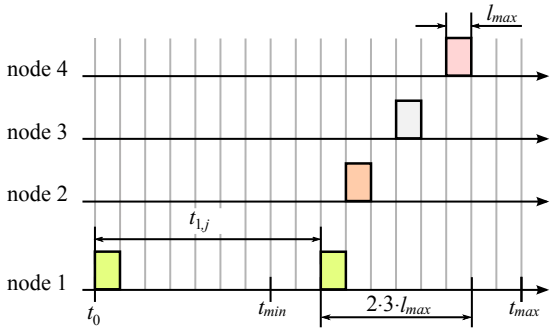

Figure 2: Computing the worst-case probability of packet loss: This results from the ratio between the maximum number of inter-packet times that potentially yield a collision to the total number of possible inter-packet times.

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

Given this case, let us analyze the example of Fig. 2 for four nodes. Node 1 sends packets at times t_0 and $t_0 + t_{1,j}$, where $1 \leq j \leq k$ denotes any of the k packets of node 1 — note that t_0 is the point in time at which node 1 sends one of its k packets and not node 1's activation time. Note that node 1's probability of loosing its second packet is given by the probability of choosing a $t_{1,j}$ (from $[t_{min}, t_{max}]$) that leads to a collision with a packet of any of the other nodes. This probability is maximum, when the fraction of $[t_{min}, t_{max}]$ leading to potential collisions, i.e., Δ_{coll}, is the greatest possible.

In Fig. 2, there is a period of time in $[t_{min}, t_{max}]$ equal to $2l_{max}$, for which any $t_{1,j}$ leads to a collision between node 1 and one of the other three nodes. This originates from the fact that even the smallest overlapping produces packet loss, i.e., a packet sent within $(t_{1,j} - l_{max}, t_{1,j} + l_{max}]$ will collide with node 1's packet. Now, in the worst case, the packets of the other three nodes are separated by at least a time equal to l_{max}. As a result, there will be three time intervals equal to $2l_{max}$ that lead to collisions with node 1. The sum of these time intervals results in the maximum value of Δ_{coll}, i.e., $2 \cdot 3 \cdot l_{max}$ in the example of Fig. 2. For n nodes, this leads to the following expression:

$$\Delta_{coll} = 2(n-1)l_{max}.$$

To generalize, let us remove the previous restriction allowing each node to send a maximum number of m packets in an interval of length $t_{max} - t_{min}$, where $1 \leq m \leq k$. In this latter case, the value of t_{min} has to fulfill the following condition:

$$\begin{aligned} m \cdot t_{min} &\geq t_{max} - t_{min}, \\ t_{min} &\geq \frac{t_{max}}{m+1}. \end{aligned} \quad (1)$$

Since there can be m packets of each node in an interval of length $t_{max} - t_{min}$, the generalized expression of Δ_{coll} is given by:

$$\Delta_{coll} = 2m(n-1)l_{max}. \quad (2)$$

As a consequence, we can compute the maximum possible probability of packet loss every time a packet is sent using the proposed MAC scheme by the ratio between Δ_{coll} and $t_{max} - t_{min}$:

$$q = \frac{2m(n-1)l_{max}}{t_{max} - t_{min}}. \quad (3)$$

Clearly, the probability of a successful packet transmission in the worst case is given by $1 - q$. Note that, for (3) to be valid the following condition must be satisfied (i.e., $q \leq 1$ must hold):

$$\begin{aligned} 2m(n-1)l_{max} &\leq t_{max} - t_{min}, \\ t_{min} &\leq t_{max} - 2m(n-1) \cdot l_{max}. \end{aligned} \quad (4)$$

In addition, since m, n, l_{max}, t_{max} and t_{min} are system parameters (i.e., common to all nodes in the WSN), q is independent of the node and of the packet being sent. As a result, we can model the transmission of packets in the network using a binomial distribution. This way, we compute the probability p that, in the worst case, at least one packet out of a sequence of k reaches its destination for any node in the network.

To compute p, we need to consider all possible combinations, i.e., only one packet arrives, two packets arrive, etc., which is a cumbersome procedure. It is much easier to compute $1 - p$, i.e., the probability that, in the worst case, no packet of a sequence of k reaches its destination. This is the probability that k consecutive packets be 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 (3), we obtain with $x = k$, i.e., k out of k packets are lost:

$$1 - p = \left(\frac{2m(n-1)l_{max}}{t_{max} - t_{min}} \right)^k. \quad (5)$$

In order that p corresponds to our definition of reliability, we need to make sure that nodes always send k packets within the specified deadline d_{max}. Towards this, recall again that every node i waits a random time t_{ij} chosen from $[t_{min}, t_{max}]$ before sending any packet. In the worst case, node i will have to wait t_{max} before sending each of its k packets as illustrated in Fig. 3. To guarantee that each node i sends its k packets within d_{max}, the following must hold:

$$t_{max} \leq \frac{d_{max} - l_{max}}{k}. \quad (6)$$

Given a value of t_{max} as per (6), we can reshape (5) to compute the value of t_{min} that satisfies a desired reliability p for the whole WSN:

$$t_{min} \leq t_{max} - \frac{2m(n-1)l_{max}}{\sqrt[k]{1-p}}. \quad (7)$$

Note from (5) that a $p = 1$, i.e., 100% reliability, can only be achieved for $n = 1$, i.e., for only one node in the

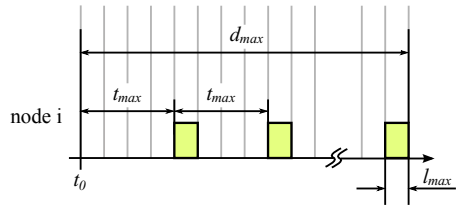

Figure 3: Relation between d_{max} and t_{max}: In the worst case, t_{max} should fit k (number of redundant packets sent) times in a time interval of length $d_{max} - l_{max}$.

network, independent of all other parameters. For $n > 1$, if p tends to 1, t_{min} tends to minus infinity as per (7). In other words, 100% reliability as with [1] or [8] 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 redundant packets sent and, hence, making considerably better use of energy.

5. PRACTICAL FACTORS

In this section, we extend our proposed analysis to consider practical factors, in particular, clock drift and external interference.

5.1 Clock Drift

All clocks used in electronic devices show a deviation in frequency with respect to each other, i.e., they *count* time at different rates. This deviation is known as clock drift and normally depends on a number of different factors such fabrication-induced variability, operating temperature, etc. As a result, since transmit-only nodes cannot be synchronized, they will unavoidably have different time scales.

Recall that a node generates random inter-packet times t_{ij} in the interval $[t_{min}, t_{max}]$ — with a uniform distribution — and waits for these t_{ij} before sending any packet. A clock drift does not affect the generation of random inter-packet times, since bounds t_{min} and t_{max} are computed off-line. In other words, the length of the interval $[t_{min}, t_{max}]$ remains constant and, hence, (3) is still valid.

However, since a node *counts* for t_{ij} before sending a packet, a clock drift leads to an *absolute* waiting time $\bar{t}_{ij}$ different than t_{ij}, i.e., the time without clock drift. As a result, this needs to be considered when selecting the bounds t_{min} and t_{max} for the random inter-packet times.

Towards this, recall that a node may send m packets in an interval of length $t_{max} - t_{min}$, where $1 \leq m \leq k$ and k is the total number of redundant packets sent by the node in one sequence. Again, the node waits for a random t_{ij} before sending each packet where t_{ij} is always greater than or equal to t_{min}. The node may need to wait for t_{min} time before each of the above mentioned m packets.

We need to consider clock drift so as to guarantee that at most m packets be in an interval of length $t_{max} - t_{min}$, otherwise (3) will stop being valid. To this end, let as denote by Δt_{min} the maximum possible clock deviation (with respect to an ideal, non-drifting clock) in an interval of length t_{min}. As a result, we proceed as follows to incorporate clock drift into (1):

$$\begin{aligned} m \cdot (t_{min} - \Delta t_{min}) &\geq t_{max} - t_{min}, \\ t_{min} &\geq \frac{t_{max} + m \cdot \Delta t_{min}}{m+1}. \end{aligned} \quad (8)$$

Similarly, t_{max} is selected such that we can guarantee that k packets be sent within the specified deadline d_{max}. To consider clock drift in this case, let us denote by Δt_{max} the maximum possible clock deviation in an interval of length t_{max}. In the worst case, a node may need to wait for t_{max} before sending each of the k packets. We proceed as follows to incorporate clock drift in (6):

$$\begin{aligned} k \cdot (t_{max} + \Delta t_{max}) &\leq d_{max} - l_{max} \\ t_{max} &\leq \frac{d_{max} - l_{max} - k \cdot \Delta t_{max}}{k}. \end{aligned} \quad (9)$$

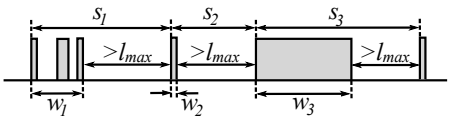

Figure 4: An exemplary sequence of external interference pulses at the communication channel. In this case, the maximum possible duty cycle σ is equal to $\frac{w_3}{s_3}$. Since no packet can be sent in a time interval that is less than l_{max}, external interference pulses that are separated by less than l_{max} are considered to be one single large pulse. This is the case of the first three pulses in the example, which are merged into on single pulse of width w_1.

Note that this new bound for t_{max} can be used instead of (6) to compute the t_{min} that satisfies the reliability requirement p in (7). The value of t_{max} in (9) needs to be considered in computing the other bounds on t_{min} such as (4) and (8) taking clock drift into account.

Clearly, (8) and (9) requires us to know Δt_{min} and Δt_{max}, which depend on t_{min} and t_{max} respectively. However, we know that clock deviation due to clock drift will increase with the length of the considered time interval. Since $t_{min} < t_{max} < \hat{t}_{max}$ holds for $\hat{t}_{max} = \frac{d_{max}}{k}$, we have that $\Delta t_{min} < \Delta t_{max} < \Delta \hat{t}_{max}$ also holds where $\Delta \hat{t}_{max}$ is the maximum clock deviation in an interval of length $\hat{t}_{max}$. As a result, to resolve the above dependency, we can safely replace Δt_{min} and Δt_{max} by $\Delta \hat{t}_{max}$ in (8) and (9).

5.2 External Interference

In order to consider external interference, we first need a model to characterize its behavior. Clearly, without such a model, it is impossible to perform any analysis. We hence assume that the maximum *duty cycle* by external interference — denoted by σ — can be determined, i.e., the greatest possible ratio between pulse width w_i to inter-pulse separation s_i of external interference — see Fig. 4:

$$\sigma = \max_{\forall i} \left(\frac{w_i}{s_i} \right), \quad (10)$$

where $i \in \mathbb{N}_{>0}$ is an index identifying the particular pulse. This σ can be obtained, for example, by measuring at the communication channel for a sufficiently large time window; or this may also be known from previous experience. Note that external interference pulses separated by less than l_{max} are considered to be one single large pulse. This is because the minimum overlapping with an external interference pulse may already yield packet loss. Hence, no data packet can be sent between such pulses as illustrated in Fig. 4.

This σ gives also the greatest probability of encountering an external interference pulse at the communication channel [14] — note that $\sigma \leq 1$ holds. This probability is clearly independent of q in (3), i.e., the probability of packet loss due to internal interference, since external interference is independent of any of the (internal) nodes in the network. As a result, when considering external interference, the probability of packet loss is given by:

$$\hat{q} = q + \sigma - q \cdot \sigma = q + (1 - q) \cdot \sigma, \quad (11)$$

i.e., the probability that a packet is lost at the communication channel either by internal or by external interference.

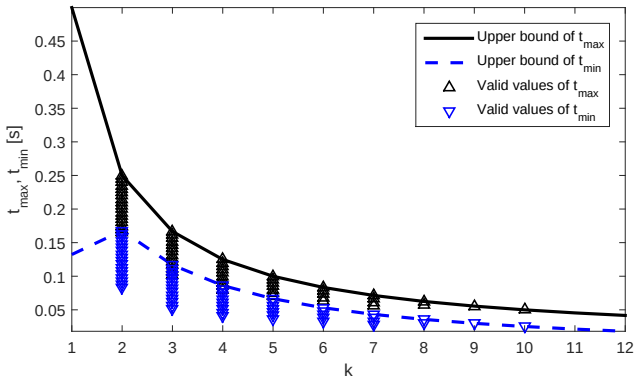

Figure 5: Valid values of t_{max} and t_{min} for $n = 50$, $p = 0.95$, $m = 1$, $d_{max} = 500$ ms and $l_{max} = 187.5$ μs.

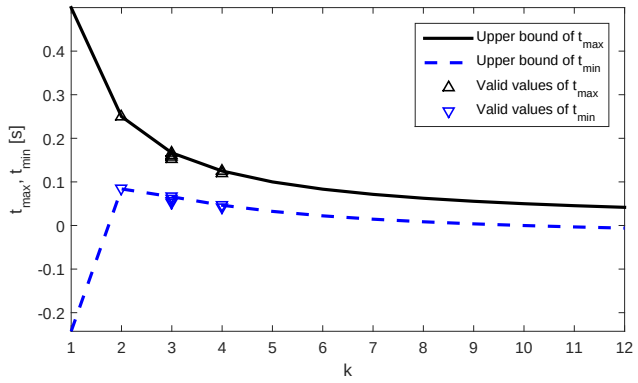

Figure 6: Valid values of t_{max} and t_{min} for $n = 50$, $p = 0.95$, $m = 2$, $d_{max} = 500$ ms, and $l_{max} = 187.5$ μs.

Note that we can still apply the binomial distribution, since $\hat{q}$ in (11) is independent of the node and the packet being sent. As a result, proceeding as for the case with no external interference, we obtain the probability that k consecutive packets are lost:

$$1 - p = (q + \sigma - q \cdot \sigma)^k , \tag{12}$$

and, hence, replacing q as per (3) we can solve for t_{min} such that p, the desired reliability requirement, is satisfied under external interference:

$$t_{min} \leq t_{max} - \frac{2m(n-1)(1-\sigma)l_{max}}{\sqrt[k]{1-p} - \sigma}. \tag{13}$$

If t_{min} becomes negative or greater than t_{max}, it will not be possible to fulfill the reliability requirement p for the given external interference σ. The other constraints on t_{min}, i.e., (1) — or (8) when considering clock drift — and (4) still have to be satisfied. The value of t_{max} is again given by (6) — or (9) when accounting for clock drift.

In summary, (6) and (7) can be used in the absence and (6) and (13) in the presence of external interference. However, dynamically adjusting t_{max} and t_{min} to the interference level is not possible, since unidirectional nodes cannot sense the channel and are therefore not able to detect any changes. The configuration must consequently be performed in the deployment phase or manually changed later.

6. NUMERICAL EVALUATION

In this section, we perform a detailed evaluation of the worst-case behavior of the proposed reliability-aware MAC for single-hop WSNs. Unless otherwise stated, we consider a data rate of 256 kbit/s and a packet size of 6 bytes (8 bits identifier, 32 bits data, and 8 bits checksum). As a result, the packet length l_{max}, i.e., the time necessary for transmitting one packet, is equal to 187.5 μs. We further assume a deadline d_{max} of 500 ms, which corresponds to typical values in home automation and body area networks, and a default number of nodes $n = 50$. Note that the results shown in this section are applicable to other possible WSN configurations as well. For simplicity, we do not take external interference and no clock drift into account in our experiments. However, the validity of observations and conclusions is not affected.

6.1 Selecting t_{max} and t_{min}

Given a single-hop WSN with parameters n, l_{max}, and d_{max}, we first need to determine the number of redundant packets k that guarantees the desired reliability p. This implies finding valid values of t_{max} and t_{min} for a given k that not only satisfy (6) and (7), but also (1) and (4). Fig. 5 and Fig. 6 show plots of (6) (solid line) and (7) (dashed line) for $p = 0.95$ and different values of k. The value of m has been set to 1 in Fig. 5 and to 2 in Fig. 6. Recall that this parameter determines the number of packets from the same node in an interval of length $t_{max} - t_{min}$. Upward- and downward-pointing triangles identify values of t_{max} and t_{min} that, apart from (6) and (7) respectively, allow meeting (1) and (4) and hence the desired $p = 0.95$.

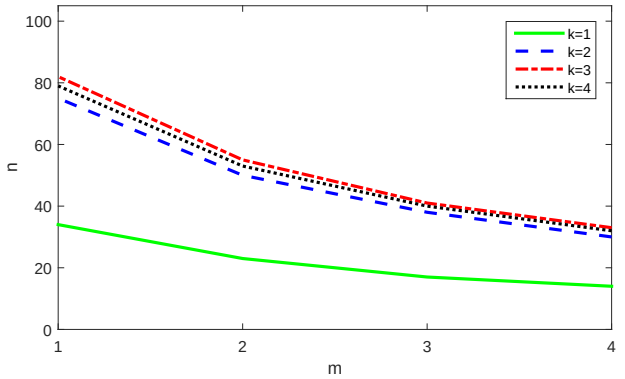

Figure 7: Number of nodes n for $1 \leq m \leq 4$, $1 \leq k \leq 4$, $l_{max} = 187.5$ μs, $p = 0.95$, and $d_{max} = 500$ ms.

From Fig. 5 and Fig. 6, we can see that $p = 0.95$ cannot be guaranteed for every k. In the case of $m = 1$ in Fig. 5, $p = 0.95$ is only possible if k is in $[2, 10]$. It should be noted that the set of valid t_{max} and t_{min} is greater for $k = 2$ than for $k = 10$. For $k = 10$, there is only one possible value for t_{max} and t_{min}. A greater k implies more redundant packets and, hence, also more interference and energy consumption at the communication channel.

Now, in the case of $m = 2$, $p = 0.95$ is only feasible for k in $[2, 4]$ as shown in Fig. 6 and, in general, there are less valid values of t_{max} and t_{min}. This is analyzed next in detail.

6.2 Effect of m

Let us now study the effect of m on the maximum number of nodes n that can be reached for a specified p. As already observed from Fig. 5 and Fig. 6, m has a negative impact on the feasibility of the WSN. This is also reflected in Fig. 7,

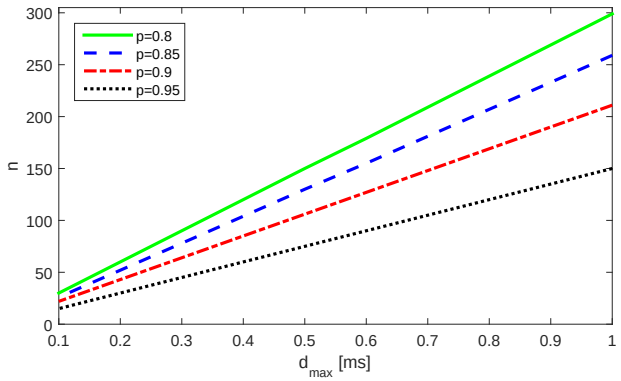

Figure 8: Number of nodes n for 100 ms $\leq d_{max} \leq$ 1000 ms, $0.8 \leq p \leq 0.95$, $k = 2$, $m = 1$, and $l_{max} = 187.5\,\mu$s.

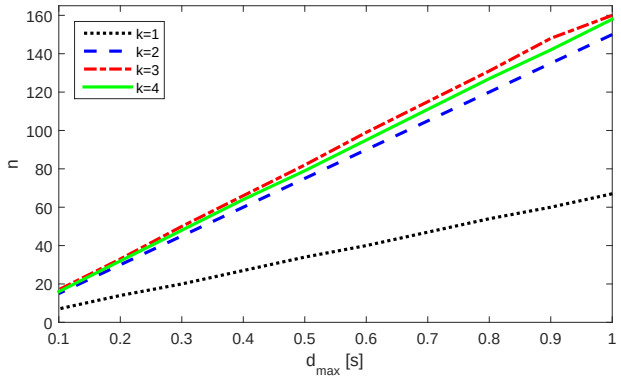

Figure 9: Number of nodes n for 100 ms $\leq d_{max} \leq$ 1000 ms, $1 \leq k \leq 4$, $p = 0.95$, $m = 1$ and $l_{max} = 187.5\,\mu$s.

which shows how n varies with m and different values of k. For $k = 2$ and $p = 0.95$, while around for 70 nodes are possible with $m = 1$, roughly above 30 can be reached with $m = 4$. An identical behavior can be observed for other k.

In other words, m allows for more flexibility in selecting values of t_{min}. The greater m is chosen, the closer t_{min} may be to zero — see (1). As a result, the interval $[t_{min}, t_{max}]$ becomes longer decreasing the probability of packet collision. On the other hand, m also increases the number of packets in $[t_{min}, t_{max}]$ from the same node, which again increases the probability of packet collision as per (3). In general, the second effect dominates such that a greater m negatively impacts the feasibility of the WSN.

6.3 Effect of d_{max}

Fig. 8 and Fig. 9 illustrate how the number of possible nodes n changes with d_{max}. In general, the longer d_{max} is, the more nodes can be accommodated for a desired p. Different values of p are evaluated for a constant $k = 2$ in Fig. 8. As expected, n decreases with increasing p, i.e., the higher the desired reliability, the less nodes can be accommodated. For a $d_{max} = 500$ ms, around 70 nodes are possible with $p = 0.95$, whereas 150 nodes can be reached, if $p = 0.8$ is chosen instead.

Fig. 9 shows the variation of n with d_{max} for different values of k and a constant $p = 0.95$. Overall, for a desired p, a greater k allows reaching more nodes. That is, sending more redundant packets increases the reliability of the WSN.

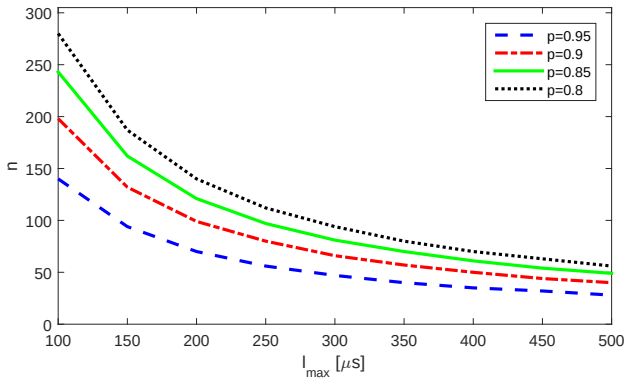

Figure 10: Number of nodes n for 100 μs $\leq l_{max} \leq$ 500 μs, $0.8 \leq p \leq 0.95$, $k = 2$, $m = 1$, and $d_{max} = 500$ ms.

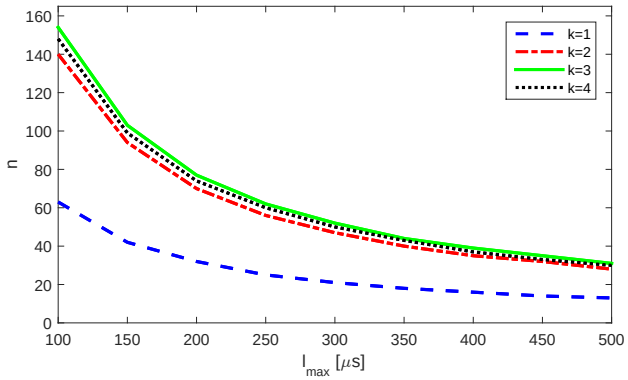

Figure 11: Number of nodes n for 100 μs $\leq l_{max} \leq$ 500 μs, $1 \leq k \leq 4$, $p = 0.95$, $m = 1$, and $d_{max} = 500$ ms.

For $d_{max} = 500$ ms and a $k = 1$, around 30 nodes can be reached. In this case, if $k = 2$ is used, we can go up to around 70 nodes. However, from $k = 4$ onwards, the situation reverses. That is, a greater k reduces the number of possible nodes, since sending more redundant packets from this point onwards increases the interference between nodes. For $d_{max} = 500$ ms and $k = 4$, we can reach less nodes (around 75) than for $k = 3$ (around 80).

6.4 Effect of l_{max}

Fig. 10 and Fig. 11 show the dependency of n on the packet length l_{max}. Here, the number of possible nodes for a desired reliability decreases as l_{max} grows. Fig. 10 shows this effect for different values of p. While 70 nodes are feasible for $p = 0.95$ and $l_{max} \approx 200\,\mu$s, double as many are possible with $p = 0.8$. Fig. 11 illustrates how n varies with l_{max} and different values of k. For a desired p, a greater k allows reaching more nodes as in the previous section. For $l_{max} = 200\,\mu$s and a $k = 1$, around 30 nodes can be reached. If $k = 2$ is used instead, 70 nodes become possible. From $k = 4$ onwards, the situation reverses as previously discussed. For $l_{max} \approx 200\,\mu$s and $k = 4$, we can reach approximately 75, whereas 80 nodes are possible with $k = 3$. It can be observed — as expected from (2) — that a smaller l_{max} reduces the probability of collision and, hence, allows for more nodes in a WSN while meeting the reliability requirement.

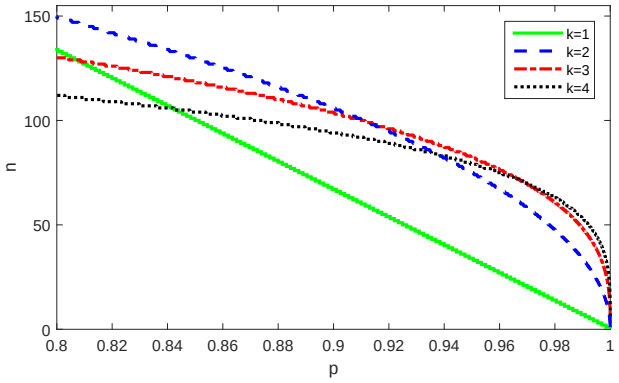

Figure 12: Number of nodes n for $0.8 \leq p \leq 1$, $1 \leq k \leq 4$, $l_{max} = 187.5\,\mu$s, $m = 1$, and $d_{max} = 500$ ms.

6.5 Reliability vs. number of nodes

Fig. 12 shows the dependency of the maximum possible number of nodes n on the desired reliability p for different values of k. We can observe that sending no redundant packets, i.e., $k = 1$, allows an acceptably big n for low values of p. With $k = 1$ and $p = 0.8$, it is possible to achieve $n = 125$. This rapidly reduces to $n = 30$ for $p = 0.95$. A sequence of two or more redundant packets is necessary to achieve high reliability at big values of n, i.e., for $p > 0.85$. For example, for $k = 2$, $n = 140$ is reachable with $p = 0.8$ and around $n = 70$ with $p = 0.95$. A higher k allows a greater n if $p > 0.95$. For example, for $k = 4$ and $p = 0.98$, around 60 nodes are possible, whereas a $k = 2$ only leads to 45 nodes. Although not shown in Fig. 12, for $k > 4$, this effect reverses at $d_{max} = 500$ ms, i.e., less nodes can be accommodated as k grows beyond 4. Note that $k > 4$ may still be beneficial in settings with longer deadlines, i.e., $d_{max} > 500$ ms.

As mentioned above, the only way of guaranteeing a reliability of 100% with our scheme, i.e., $p = 1$, is with only one node, i.e., $n = 1$. This can also be observed in Fig. 12 where all curves — for all k — tend to 1 as p tends to 1.

7. SIMULATION AND COMPARISON

In this section we compare the proposed MAC technique with other possible approaches from the literature. To this end, we have performed a simulation based on the OMNeT++ framework [13] and on its extension for mobile and wireless networks called MiXiM [6]. This allows us to effectively simulate our network with different physical parameters and to record statistical values for very large numbers of transmissions.

We assume that the network has been configured correctly such that each transmitter has a transmission power, which is high enough to ensure good link quality to its connected receiver. Here, the transmission power is set high enough to counteract the slowly depleting battery during the nodes lifetime. Just as before, we assume a data rate of 256 kbit/s and the packet size of 6 bytes, which results in a packet length l_{max} of 187.5 μs.

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 sending packets with a certain pattern according to the algorithms under comparison.

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 taken into account. The resulting simulation data is processed by the OMNeT++ framework at runtime. In particular, the time stamps of simulated packets transmissions are compared to determine whether packets overlap and, hence, interfere with each other.

We consider the following four MAC schemes for the sake of comparison in our simulation:

- The *proposed* scheme is the one presented in Section 4, for which we set $k = 2$, i.e., two redundant packets per sequence, and a reliability requirement $p = 0.95$.
- The *simplistic* scheme transmits a sequence of 4 consecutive packets separated by transmission pauses. This method is easy to implement and used in commercial devices based on PT2262 [9], for example, Intertechno[2] products, which are used in home automation settings.
- The *periodic* algorithm is the transmission scheme as presented in [8]. It transmits n packets with constant inter-packet times that are configured to avoid collisions. After n packets have been sent, this algorithm also implements a transmission pause.
- The *optimized* algorithm is the transmission scheme as presented by Anderson et al. [2] [1]. It is also based on transmitting n packets with constant inter-packet times that are configured to avoid collisions. In contrast to the *periodic* algorithm, inter-packet times here have been optimized, i.e., they are shorter.

The above algorithms are simulated and compared for different numbers of nodes n. For each value of n, 100,000 different packet sequences have been simulated to obtain a meaningful comparison. All nodes in the simulation were triggered as frequently as possible to attain a high network load and, hence, yield results comparable to the worst case. This allows us to check if the reliability requirements can be met for the *proposed* scheme, which would be not possible for low network loads. Again, for the sake of simplicity, we do not consider external interference or clock drift in the simulation, which also goes in line with the assumptions made by the *periodic* and the *optimized* algorithms.

7.1 Packet sequence loss vs. number of nodes

Fig. 13 shows the average number of packet sequences that are lost by the different algorithms as n increases. For both the *proposed* and the *simplistic* algorithm, the amount of packet sequences lost increases with a rising number of transmitters. This is due to the fact that the capacity of the communication channel is limited. As a consequence, clearly, more transmitters result in a higher collision probability on the communication channel.

Since the *simplistic* algorithm introduces no randomness in the selection of inter-packet separation, there is a high probability that multiple packets are destroyed in case of a collision. Its collision probability increases more steeply for higher n than for the *proposed* algorithm. For $n = 100$, it

[2]www.intertechno.at

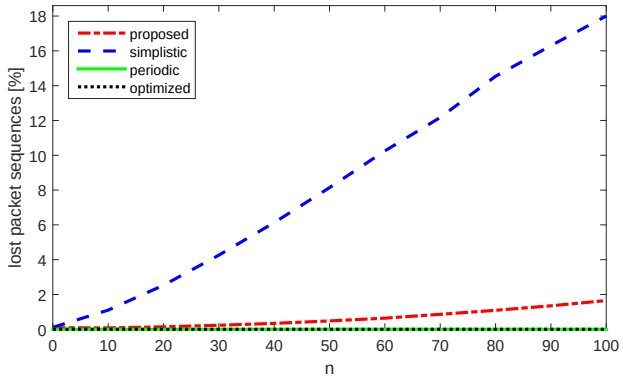

Figure 13: The average number of lost packet sequences is depicted for the four compared algorithms as the number of nodes n increases.

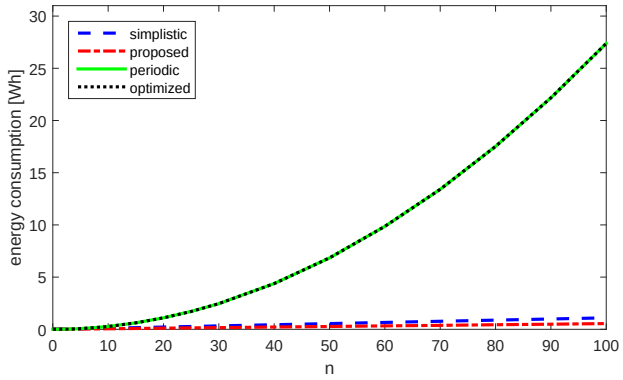

Figure 14: The yearly energy consumption used for packet transmission by the different algorithms is shown. On average, each node is triggered a number from 6 to 10 times per day, which corresponds to the common activation rates in home automation networks. The energy consumption during transmission is 5 mW.

loses an average of 18% of its total packet sequences. For low n, both algorithms have a very low collision probability, since their inter-packet separations are large enough compared to n and the number of packets sent.

On the other hand, it can be observed that the *proposed* algorithm's average loss rate of packet sequences is always below $1 - p = 1 - 0.95 = 5\%$. That is, the reliability requirement can always be fulfilled. Both the *periodic* and *optimized* algorithm lose zero packet sequences, since they are designed to guarantee that at least one packet of a sequence reaches its receiver in the worst case.

7.2 Energy consumption vs. number of nodes

Fig. 14 depicts the yearly consumed transmission energy by the different algorithms. Each node is triggered from 6 to 10 times per day — which is a typical activation rate of an appliance switch, light switch or a presence sensor in a home automation setting — consuming 5 mW for the duration of a packet transmission l_{max}. Due to the fact that both the *periodic* and the *optimized* algorithm send n packets per node, their energy consumption rises quadratically leading to a very high energy consumption for increasing values of n. For $n = 100$, they require a yearly total energy consumption of around 28 Wh, whereas the *proposed* algorithm only requires approximately 0.55 Wh.

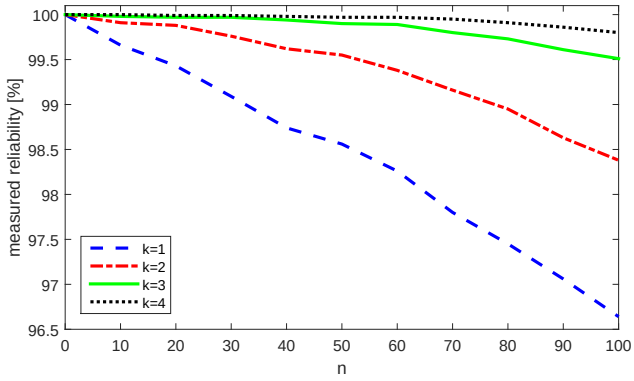

Figure 15: The measured reliability of the *proposed* algorithm with $p = 0.95$ is illustrated for a varying n and different values of k.

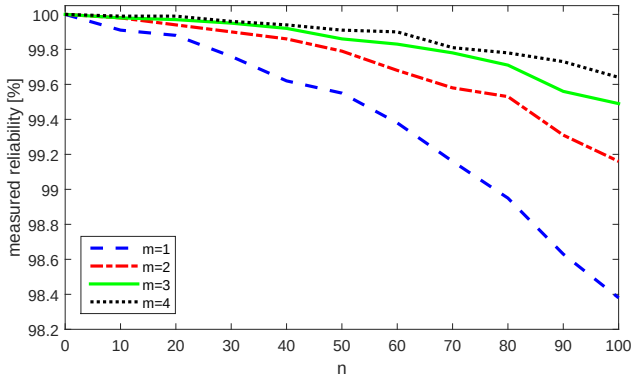

Figure 16: The measured reliability of the *proposed* algorithm with $p = 0.95$ is illustrated for a varying n and different values of m.

In contrast, the energy consumption of the *proposed* and the *simplistic* algorithm increases linearly, since the number of packets sent upon activation of a node does not depend on the number of nodes n. For our *proposed* algorithm, we have shown that a $k = 2$ suffices to meet necessary reliability requirements, where the *simplistic* one always sends a sequence of 4 packets. Clearly, if the *proposed* algorithm is configured for $k = 4$, it will consume the same energy as the *simplistic* one.

7.3 Measured reliability for different k

Complementary to Section 6, where the worst-case behavior is illustrated in a numerical evaluation, in this and the next section, we analyze the average-case behavior of the proposed approach. Fig. 15 shows the measured reliability, i.e., the one observed in repeated simulations based on OMNet++, for a varying number of nodes and different values of k. Just as before, the *proposed* algorithm has been configured for meeting a reliability requirement of $p = 0.95$.

It can be observed that the average reliability is much higher than $p = 0.95$, i.e., on average, much less packets are lost. In addition, in contrast to the worst-case behavior, the average reliability increases with a growing k, as it is less likely that all packets be lost within a sequence. For

example, $k = 4$ has a better average behavior for $p = 0.95$ than $k = 3$, while we know that the worst-case behavior of $k = 4$ is worse than that of $k = 3$ — see Fig. 9 and Fig. 11. This then improves for higher values of p as shown in Fig. 12.

7.4 Measured reliability for different m

Fig. 16 shows the measured reliability for a varying number of nodes and different values of m. Again, the *proposed* algorithm has been configured for meeting a reliability requirement of $p = 0.95$. The average reliability observed in this case is also much higher than $p = 0.95$. For $m = 1$, at most 3% of the packet are lost for $n = 100$. It should also be noted that $m = 4$ has a better average behavior for $p = 0.95$ than lower values of m. This is the opposite of the worst-case behavior in Section 6, where $m > 1$ has been shown to negatively impact reliability — see Fig. 7.

8. CONCLUDING REMARKS

In this paper, we proposed a MAC technique that allows assessing reliability of single-hop WSNs based on unidirectional nodes, i.e., nodes that can either transmit or receive packets. With our approach, it is possible to design such a WSN to fulfill a pre-specified reliability requirement denoted by p, where $0 < p < 1$ holds and $p = 1$ stands for a fully reliable network.

The proposed technique consists in making every transmit-only node send a sequence of k redundant packets with random inter-packet times that are generated between t_{min} and t_{max}, where t_{max} and t_{min} are design parameters. We computed the probability that one packet is lost on the communication channel due to interference in the worst possible case. This probability was shown to be independent of the node sending and the packet being sent.

On this basis, we modeled the transmission of a sequence of redundant packets using a binomial distribution, which allows us to compute the probability that, in the worst case, at least one of a sequence of redundant packets reaches its destination within a given deadline. This latter probability was defined as reliability of a unidirectional single-hop WSN.

Finally, we extended our analysis to consider clock drift and external interference and presented detailed experimental results based on a numerical evaluation and on an OMNeT++ simulation. Our experiments show that the proposed technique leads to a maximum loss rate of packet sequences that is always below $1-p$. In contrast to similar approaches from the literature, our technique is less conservative and allows for a more cost-effective and energy-efficient design.

9. REFERENCES

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

[2] B. Andersson, N. Pereira, and E. Tovar. Delay-Bounded Medium Access for Unidirectional Wireless Links. Technical report, CISTER - Research Centre in Real-Time and Embedded Computing Systems, 2007.

[3] B. Blaszczyszyn and B. Radunovic. Using Transmit-only Sensors to Reduce Deployment Cost of Wireless Sensor Networks. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2008.

[4] C. Huebner, S. Hanelt, T. Wagenknecht, R. Cardell-Oliver, and A. Monsalve. Long Range Wireless Sensor Networks Using Transmit-only Nodes. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2010.

[5] H. Keong, K. Thotahewa, and M. Yuce. Transmit-only Ultra Wide Band Body Sensors and Collision Analysis. *IEEE Sensors Journal*, 13:1949–1958, 2013.

[6] 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.

[7] G. Mazurek. Collision-Resistant Transmission Scheme for Active RFID Systems. In *Proceedings of the International Conference on Computer as a Tool (EUROCON)*, 2007.

[8] 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.

[9] Princeton Technology Corp. PT2262 Datasheet. URL: http://www.princeton.com.tw.

[10] B. Radunovic, H. L. Truong, and M. Weisenhorn. Receiver Architectures for UWB-Based Transmit-Only Sensor Networks. In *Proceedings of the IEEE International Conference on Ultra-Wideband (ICU)*, pages 379–384. IEEE, 2005.

[11] L. G. Roberts. Aloha Packet System With and Without Slots and Capture. *Computer Communication Review (SIGCOMM)*, 5:28–42, 1975.

[12] B. Tas and A. Tosun. Data Collection Using Transmit-only Sensors and a Mobile Robot in Wireless Sensor Networks. In *Proceedings of the International Conference on Computer Communications and Networks (ICCCN)*, 2012.

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

[14] M. Weisenhorn and W. Hirt. Uncoordinated Rate-Division Multiple-Access Scheme for Pulsed UWB Signals. *IEEE Transactions on Vehicular Technology*, 54:1646–1662, 2005.

[15] J. Zhao, C. Qiao, R. S. Sudhaakar, and S. Yoon. Improve Efficiency and Reliability in Single-Hop WSNs with Transmit-Only Nodes. *IEEE Transactions on Parallel and Distributed Systems*, 24(3):520–534, 2013.

[16] B. Zhen, M. Kobayashi, and M. Shimizu. To Read Transmitter-only RFID Tags with Confidence. In *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2004.