Designing Reliable Home-Automation Networks Based on Unidirectional Nodes

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

***Abstract*—There has been an increasing interest in home automation over the last few years. In particular, applications such as intelligent illumination, heating, and ventilation, allow reducing the overall energy consumption and improve comfort in our everyday lives. To implement such applications, multiple sensors and actuators often need to be connected into networks typically communicating over radio signals. Since there may be a considerable amount of packet losses due to interference in the network, most available technologies are based on bidirectional devices or nodes with the capability of acknowledging packets, and performing retransmissions if necessary. On the other hand, bidirectional nodes are expensive and rapidly increase costs of a home-automation network, in particular, as the number of nodes increases. Alternatively, we can use unidirectional nodes, which can either send or receive data and, hence, are considerably less expensive. However, since unidirectional nodes are unable to perform carrier sensing or acknowledge packets, the resulting network is strongly unreliable. To overcome this predicament, we propose a design technique that allows guaranteeing a reliable communication based on unidirectional nodes. Our technique consists in making each transmit-only node in the system send a sequence of packets with constant inter-packet time or separation. We prove analytically that, with the proposed technique, it is always possible to guarantee that at least one packet of each sequence reaches its receiver on time. We further evaluate our approach with a simulation based on OMNeT++.**

Keywords*—*Home automation, cost efficiency, reliability, sensor/actuator networks, unidirectional nodes

I. Introduction

Nowadays, an increasing number of electronic devices populate our homes providing comfort in many areas of our lives. These devices need to be connected into networks to perform different automation tasks, viz., home automation. This way, simple applications such as automatically raising shutters in the morning as well as more elaborate ones can be easily created. Typical applications include HVAC (Heating, Ventilation, and Air Conditioning), surveillance, door/window control, etc.

There are different communication technologies that are commonly used for transmitting data in home-automation networks. The most used technologies are normally optical, wired, or radio-based. Optical, more specifically, infrared communication is mainly used for multimedia systems, e.g., a TV remote control, and is seldom used for automation tasks in homes, since it requires a direct line of sight and it typically has a very limited range.

On the other hand, both wired and radio-based communication are widely used for automation tasks. However, radio-based systems are the preferred transmission method, as they allow for high flexibility with respect to installation and have low entry costs. For wired Systems, such as EIB/KNX or BACnet [1], cables need to be installed leading to an inflexible design, particularly, if this is not planned from the beginning, and high entry costs. Solutions like Zigbee-KNX Gateway [2] or KNX-RF [3] aim to connect both wired and wireless systems, to combine the benefits of both technologies. Nevertheless, these mixed systems are usually higher priced and still not as flexible as purely radio-based systems.

The high popularity of wireless communication has led to a wide variety of different technologies, ranging from simple transmit- or receive-only devices (EnOcean[1], Intertechno[2]), over Internet-enabled solutions [4] to fully meshed networks with routing capabilities such as ZigBee and Z-Wave[3]. In general, we can classify such technologies either into unidirectional or bidirectional systems.

Unidirectional devices or nodes can either transmit or receive data. As a result, data packets cannot be acknowledged and the states of nodes cannot be *retrieved* making it difficult to implement reliable communication. This makes unidirectional devices only suitable for non-safety-critical applications such as environmental monitoring [5], etc., where packet losses can be tolerated. The advantages of unidirectional nodes are a low cost and a better power efficiency compared to bidirectional nodes [6], since they do not need to power a receiver to monitor the radio channel.

On the other hand, bidirectional devices or nodes can both send and receive data. This allows acknowledging packets and retrieving the states of nodes, and enables more complex communication mechanisms such as packet routing and automatic retransmission. However, this comes at a higher computational expense and, together with a more complex transceiver circuitry, such devices are more expensive than simple unidirectional nodes. Regarding the high number of devices that are often needed in home-automation networks, there can be a substantial cost saving, if unidirectional nodes are used.

Since unidirectional transmission may incur in packet loss, special protocols and algorithms have to be used to improve reliability of communication. Unfortunately, existing solutions like CSMA (Carrier Sense Multiple Access), TDMA (Time Division Multiple Access) or slotted Aloha [7] cannot be used, because transmit-only nodes cannot perform carrier sensing nor synchronization. Therefore, most commercial systems

[1]http://www.enocean.com/
[2]http://www.intertechno.at/
[3]http://www.z-wavealliance.org/

based on unidirectional nodes successively send a sequence of packets (either with constant or random inter-packet times) to increase the probability of a successful reception. Clearly, this technique decreases energy efficiency as often more packets are sent than really necessary. Moreover, even if they improve the probability of a successful communication, these systems are unable to provide any guarantee that at least one packet from a transmit-only node reaches its corresponding receiver.

A. Contributions

In this paper we are concerned with the design of reliable sensor/actuator networks for home automation using unidirectional radio-based nodes. We proposed a technique which consists in each transmit-only node sending a sequence of α_i packets with constant inter-packet time p_i. Our technique is based on the observations that (i) packet losses mostly originate from simultaneous transmissions by neighboring nodes, and (ii) interference from outside is almost negligible. Our contributions can be summarized as follows:

- We identify the worst-case situation leading to the greatest amount of *unavoidable* packet losses. Based on this, we determine the minimum necessary number of packets that need to be sent by any transmit-only node such that at least one packet reaches the receiver.
- We propose a method for obtaining suitable inter-packet times p_i guaranteeing that at least one of the α_i packets reaches the receiver in the worst case and within a maximum deadline d_{max}.

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

Reducing the cost of sensor networks by using transmit- and receive-only nodes, is an idea that has appeared in various scenarios in the past: RFID Systems [8] [9], long range outdoor networks [5], wireless body area networks [10] and indoor networks [11]. The challenge hereby is to find communication schemes to allow for reliability, energy efficiency, and a bounded delay. As already mentioned, transmit-only nodes cannot perform carrier sensing nor synchronization, therefore, existing algorithms such as CSMA and Aloha cannot be used.

This behavior poses a problem for communication reliability, since there is no feedback for lost or corrupt packets. In this context, Cardell-Oliver et al. [12] identified three *error-contention* strategies: temporal diversity, spatial diversity and code-based methods. In temporal diversity, data packets are, for example, transmitted repeatedly at different times, while spatial diversity aims to separate devices geographically such that they do not fall within each other's range and, hence, cannot interfere with one another. Code-based methods add redundant data, which can be used by the receiver to correct damaged packets. However, the increased packet length leads to less energy efficiency and higher collision probability due to longer transmission times. Cardell-Oliver et al. further concluded that temporal and spatial diversity yield better results for indoor scenarios than code-based strategies [12].

An approach based on time diversity, called Timing Channel Aloha (TC-Aloha), has been proposed by Galluccio et al. [13]. This scheme encodes parts of the information in the inter-packet separation of the different nodes. As a result, the packet length and transmission time can be reduced, thus, decreasing energy consumption and the probability of packet collision. In order to improve reliability, packets are transmitted multiple times. Clearly, to recover the information embedded in the inter-packet separation, at least two packets must be received. By an analytical framework and experimental results, Galluccio et al. [13] prove that TC-Aloha increases data throughput; however, no guarantees can be given on whether data always reaches the corresponding receivers on time.

Another approach presented in [6] consists in using two different types of transmitters. Simple unidirectional transmitters (forming clusters) for cost reduction and so called *cluster heads* with receiving capability. The cluster heads collect the packets from their corresponding unidirectional clusters and forward them to receivers. Since cluster heads can perform carrier sensing and acknowledge packets, more sophisticated communication schemes can be implemented upon them. However, if many cluster heads are necessary for a good coverage in a home-automation scenario, costs increase rapidly. Moreover, this method cannot guarantee a fully reliable communication and packets may potentially never reach their receivers due to collisions, in particular, within one cluster.

In the domain of wireless sensor networks, Andersson et al. [14], [15] presented a transmission scheme for transmit-only nodes, which can guarantee that data always reaches its corresponding receiver. To this end, each transmitter sends a sequence of redundant data packets in a periodic pattern, such that at least one data packet of each transmitter cannot be interfered. The periodic patterns were found by using an ILP (Integer Linear Programming) solver and optimized for the shortest total transmission duration; however, these patterns greatly differ in length, meaning that some transmitters are very fast at sending their packets whereas others have total transmission durations which are multiple times longer. This leads to big differences in terms of the average and total delay and hinders the system analysis. Furthermore, it is assumed that all data packets have the same length and there are no interferences or noise generated by the outside world.

In this paper, we propose a technique to design reliable communication for home automation, which is completely based on unidirectional nodes. In contrast to most of the above approaches, our technique provides a guarantee by construction that data always reaches the corresponding nodes. Similar to [14], [15], our method involves sending a sequence of α_i identical packets for each message, with a constant inter-packet separation time which is specific for each node. However, in contrast to [14], [15], we consider home automation where all nodes need to have similar (worst-case) transmission durations. As stated above, the length of the packet sequences in [14], [15] strongly changes from node to node. In addition, we further obtain analytical upper bounds on the inter-packet times

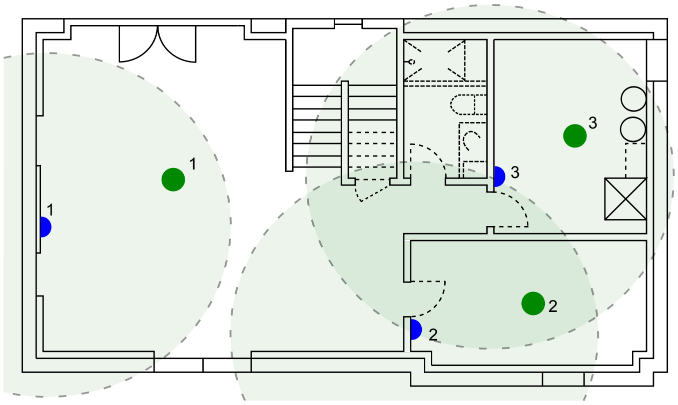

Fig. 1. Example of a home-automation network consisting of three transmit-only (blue semicircles) and three receive-only nodes (green circles). Which transmitter sends to which receiver is indicated by numbers, i.e., transmitter 1 communicates with receiver 1, etc.

of transmit-only nodes. Our technique results in deterministic behavior facilitating system analysis and improving the average delay.

In the next section we discuss the system model and assumptions used in the remainder of the paper.

III. System Model and Assumptions

We consider a sensor/actuator network of simple transmit- and receive-only nodes that are distributed within a room. Fig. 1 shows an example consisting of three transmitter (blue semicircles) and three receiver nodes (green circles). A transmitter node is activated by an event, e.g., a user turns on the light, or a motion detector gets triggered, etc. Upon activation, the corresponding transmitter broadcasts its data within a certain range of about $5\,\mathrm{m}$ to $10\,\mathrm{m}$ and, hence, every receiver within that range will receive this data – see shaded regions around transmitters in Fig. 1. If a receiver is *connected* to a transmitter[4], it will process data; otherwise this is discarded. Since transmit-only nodes are unable to detect whether data has reached the receiver, they transmit a sequence of α_i packets to increase the probability that at least one of them reaches the receiver – note that α_i is an integer number greater than one. In this paper, we consider a constant inter-packet time denoted by p_i for each node where $1 \leq i \leq n$ holds.

Communication is performed via $868\,\mathrm{MHz}$ radio, which ensures an interference-free coexistence with common devices using $2.4\,\mathrm{GHz}$, such as WLAN routers, microwaves, etc., and allows higher data rates compared to $433\,\mathrm{MHz}$. Moreover, the $868\,\mathrm{MHz}$-band is restricted to 1% duty cycle which means that, devices have to implement transmission-free intervals that are 99 times longer than their corresponding transmission times. This helps minimizing interference with other devices.

On the other hand, since nodes are deployed within a house or building, they are shielded from the outside world to a great extent. As a result, we assume that interference from outside the network is zero[5] and the only interference originates from simultaneous transmissions of neighbor nodes with overlapping ranges[6]

A packet transmitted by a node usually consists of an identifier field identifying the transmitter, a data field and a check sum or CRC (Cyclic Redundancy Check) field. In this paper, we do not analyze the application itself, i.e., how data is used to provide a desired functionality, but rather the reliable communication between such nodes.

Packets from the different nodes usually have a relatively constant length around three bytes (a one-byte identifier, a one-byte data, and a one-byte CRC). Transmitting a packet takes a given amount of time which 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} the maximum length of any packet in the system.

In order that our sensor/actuator network work as expected, for each transmitter node in the system, it must be guaranteed that one packet arrives within a deadline measured from the node's activation time. For example, in case of a light switch, light should be turned on within 0.5 seconds. A greater delay is unacceptable, since it will negatively impact the quality of the system. In this paper, we consider that this deadline is the same for all transmitter nodes and denote it by d_{max}.

Finally, each transmitter is assumed to be activated only once within a time interval t_{max} where $d_{max} < t_{max}$ 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 introduce our proposed approach for designing reliable home-automation networks using unidirectional nodes. Based on the assumption that interference from outside the network is negligible, our technique consists in identifying the worst case situation, where the most packets can be lost due to interference from between nodes.

As stated above, nodes with overlapping ranges may interfere with each other if they happen to transmit simultaneously. For a reliable communication, it has to be guaranteed that at least one packet reaches its corresponding receiver within a pre-specified d_{max}. Towards this, we first obtain the worst-case number of packets that need to be transmitted by any node in the system. This is stated in the following lemma.

Lemma 1: Let us consider a set of n independent transmit-only nodes, which are activated once within a time interval of length t_{max} and transmit a sequence of α_i packets within d_{max} where $d_{max} < t_{max}$. If $t_{max} \geq 2 \times d_{max}$ holds, it cannot be avoided that at least $n-1$ packets out of α_i be lost independent of the inter-packet time p_i of the different nodes.

Proof: Since events triggering the different nodes are independent of each other, e.g., a user turns on the light, or a motion detector gets activated, they can occur at arbitrary

[4] An identifier is sent in each packet by the transmitter, which is then recognized by its corresponding receiver.

[5] Note that other devices use $868\,\mathrm{MHz}$, e.g., garage doors, toys, etc. However, these are usually low-power and have a short range.

[6] If there are interferences which cannot be avoided, for example, from other networks in neighboring apartments, these interferences can be modeled by using additional nodes when planning the system.

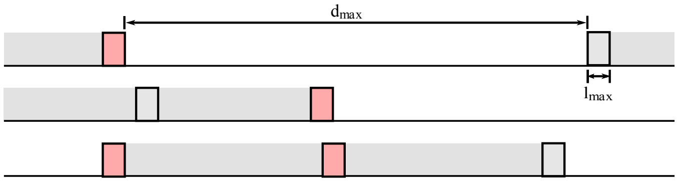

Fig. 2. Illustration of Lemma 1 for the case of three nodes. The last packet of the top node interferes with the first packet of bottom node (reddish shading). Given that the second packet of the bottom node might also be lost, its third (and last) packet can only be guaranteed to reach its receiver (grayish shading), if the top node is forced to wait for at least d_{max} time before transmitting anew.

points in time. It might be the case that every time a node tries to transmit a packet, this gets interfered by a packet of another node being activated at that time independent of the inter-packet time p_i of the different nodes. As a result, the corresponding packets are lost.

Let us assume that node j starts transmitting at time t_0. By assumption, this then sends α_j packets within $[t_0, d_{max}]$ with a constant inter-packet time p_j. In the worst case, this node sends its last packet 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 its α_i packets at time $t_0 + d_{max} - l_{max}$, the last packet of node j and the first packet of node i will be lost. If the remaining $n-2$ nodes in the system get activated at $t_0 + d_{max} - l_{max} + k_i \times p_i$ where k_i is an integer number and $1 \leq k_i \leq n-2$, all first $n-1$ packets of node i will be lost independent of any inter-packet time in the system.

Now let us further assume that the n-th packet of node i is sent at $t_0 + 2\times d_{max} - 2\times l_{max}$ such that it finishes being sent by $t_0 + 2\times d_{max} - l_{max}$ (i.e., within d_{max} from the activation time of node i). This packet of node i will only be able to reach its receiver, if node j is not activated anew until time $t_0 + 2\times d_{max} - l_{max}$. Since the minimum overlapping between any two packets yields packet loss, in order that node i's n-th packet is not interfered by node j, this latter should not be activated anew until $t_0 + 2\times d_{max}$ – see illustration in Fig. 2. The lemma follows. □

As a result of the above lemma, each node needs to transmit a minimum of n packets in order that at least one reaches the receiver within d_{max}, i.e., $\alpha_i = n$ for $1 \leq i \leq n$. In addition, each node should only be activated once with $t_{max} \geq 2\times d_{max}$. Lemma 1 only considers the first packet being sent by each node in the system. However, each node sends a sequence of n packets. Any of these subsequent packets may collide with other packets of the different nodes leading to interference. As a result, Lemma 1 states necessary, but not sufficient conditions for a reliable communication. In other words, to guarantee that at least one packet reaches the receiver in the worst case, we need to perform a more detailed analysis.

In principle, from the above discussion, we know that a packet from one node can be interfered by a packet of another node sending at the same time. In addition, the minimum overlapping between any two packets leads to packet loss, since data can be corrupted. The following theorem states a necessary and sufficient condition for guaranteeing that, among the n packets sent by a node, at least one of them reaches its receiver.

Theorem 1: Let us consider a set of n independent transmit-only nodes. Each of them sends a sequence of n packets within d_{max} and is activated at most once within $t_{max} = 2\times d_{max}$. At least one packet of each node can be guaranteed to reach its receiver, if the following condition holds for $1 \leq i \leq n$, $1 \leq j \leq n$, $1 \leq k_i \leq n-1$, and $i \neq j$:

$$\bmod\left(\frac{k_i \times p_i}{p_j}\right) \geq 2\times l_{max}, \quad (1)$$

where $\bmod(\cdot)$ is the modulo operation, l_{max} is the maximum length of a packet, while p_i and p_j are the (constant) inter-packet times of the i-th and the j-th node respectively.

Proof: Let us assume that a node starts sending its sequence of n packets at time t_0 with a constant inter-packet time p_i. According to Lemma 1, the first $n-1$ packets can be intercepted by packets of the remaining nodes, if all nodes send their corresponding n packets within d_{max} and are only activated once in a time interval $t_{max} = 2\times d_{max}$. In order that at least one packet of an i-th node safely reaches its receiver, it should be guaranteed that no other packet from any other node affects the n-th packet of the i-th node.

If the j-th node affects the first of packets from the i-th node, none of the subsequent packets from node j should affect the n-th packet of node i. That is, none of node j's packets should have any overlapping with the n-th packet of node i. Clearly, the n-th packet of node i is sent at time $t_0 + (n-1)\times p_i$. Let us now assume that a packet of node j is also sent at time t_0 interfering with the first packet of node i. The remainder of $\frac{(n-1)\times p_i}{p_j}$ should allow for enough *space* to send a node j's packet before the n-th packet of node i start being sent, i.e., before $t_0 + (n-1)\times p_i$: $\bmod\left(\frac{(n-1)\times p_i}{p_j}\right) \geq l_{max}$. However, a packet of node j can still affect the first packet of node i, if this is sent at time $t_0 - l_{max} + \varepsilon$ or $t_0 + l_{max} - \varepsilon$ where ε is an infinitesimally small amount of time. This is because, in the latter case, there will be some amount of packet overlapping (given by ε) between node i and j. As a result of this, the remainder of $\frac{(n-1)\times p_i}{p_j}$ should allow for enough space to send a node j's packet with whatever initial overlapping between the first packet of node i and a packet of node j: $\bmod\left(\frac{(n-1)\times p_i}{p_j}\right) \geq 2\times l_{max}$. In a similar manner, if a packet of node j affects the second packet of node i that is sent at time $t_0 + p_i$, it should be guaranteed that none of the subsequent packets of node j can affect node i's n-th packet: $\bmod\left(\frac{(n-2)\times p_i}{p_j}\right) \geq 2\times l_{max}$. For any two nodes i and j where $1 \leq i \leq n$, $1 \leq j \leq n$, and $i \neq j$, this translates in that $\bmod\left(\frac{k_i\times p_i}{p_j}\right) \geq 2\times l_{max}$ has to hold for $1 \leq k_i \leq n-1$. The theorem follows. □

Corollary 1: Let us assume that, according to Theorem 1, (1) holds for $1 \leq i \leq n$, $1 \leq j \leq n$, $1 \leq k_i \leq n-1$, and $i \neq j$. If all n transmit-only nodes are activated simultaneously at time t_0, the first packet of each such nodes will be lost; however, there will be no more packet losses in $[t_0, t_0 + (n-1)\times p_i]$ for $1 \leq i \leq n$.

Proof: Let us consider that any pair of nodes i and j where $i \neq j$ are activated together at time t_0. If (1) holds for $1 \leq k_i \leq n-1$, there is at least $2\times l_{max}$ space between any packets of i and j in $[p_i, (n-1)\times p_i]$. This means that, with

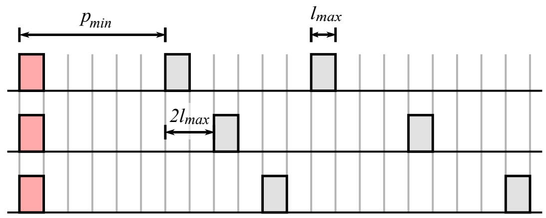

Fig. 3. Illustration of Corollary 1 for the case of three nodes, i.e., $n = 3$. If (1) holds according to Theorem 1, in the case of a simultaneous activation, the first packet of each node will get lost (reddish shading). However, there will be no further packet losses for the next two packets (i.e., $n-1$) of each node (grayish shading).

exception of the first packets sent at t_0, there is no overlapping between packets of node i and j in $[t_0, (n-1) \times p_i]$ – see illustration in Fig. 3. Further, if (1) holds for all i and j where $i \neq j$, no packets are lost in $[t_0, (n-1) \times p_i]$ and $1 \leq i \leq n$ besides the first one sent at t_0 by each node. The corollary follows. □

Theorem 1 allows us to guarantee that at least one packet of each node reaches its corresponding receiver within d_{max}, provided that each node send n packets within d_{max}. However, it does not help select the values of p_i for each of the nodes. To this end, let us consider the following analysis.

Lemma 2: Let us consider a set of n independent transmit-only nodes. Each of them sends a sequence of n packets within d_{max} and is activated at most once within $t_{max} = 2 \times d_{max}$. In order to guarantee that at least one packet of each such nodes reaches its corresponding receiver, the following condition must hold for any p_i and p_{i-1} where $1 < i \leq n$:

$$p_i - p_{i-1} \geq 2 \times l_{max}. \tag{2}$$

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

Proof: According to Theorem 1, if (1) holds for all i and j where $i \neq j$ and $1 \leq k_i \leq n-1$, it can be guaranteed that at least one packet of each node reaches its corresponding receiver.

Now, for a any i, $j = i-1$, and $k_i = 1$, from (1) we have $\mathrm{mod}\left(\frac{p_i}{p_{i-1}}\right) \geq 2 \times l_{max}$, 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 $p_i - p_{i-1} \geq 2 \times l_{max}$ and the lemma follows. □

Theorem 2: Let us consider a set of n independent transmit-only nodes. Each of them sends a sequence of n packets within d_{max} and is activated at most once within $t_{max} = 2 \times d_{max}$. 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 $p_{min} = \min_{1 \leq i \leq n}(p_i)$:

$$p_{min} \geq 2 \times n \times l_{max}, \tag{3}$$

where $p_{min} < p_i < 2 \times p_{min}$, i.e., $\lfloor \frac{p_i}{p_{min}} \rfloor = 1$, for $1 \leq i \leq n$.

Proof: Let us assume that one packet of a node i is interfered at time t_0 by a packet of node j. In order that the next packet sent by node i is not interfered again by node j, according to Theorem 1, (1) needs to hold for all i and j where $i \neq j$ and $k_i = 1$.

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 $p_{min} = p_1$ holds). Since $\lfloor \frac{p_i}{p_{min}} \rfloor = 1$ is assumed to hold, note that $\lfloor \frac{p_i}{p_j} \rfloor = 1$ also holds for all i and j.

Now, if $i = n$, $j = n-1$, and $k_n = 1$ hold, from (1) we have $\mathrm{mod}\left(\frac{p_n}{p_{n-1}}\right) = p_n - p_{n-1} \geq 2 \times l_{max}$, which leads to $p_n \geq p_{n-1} + 2 \times l_{max}$. Similarly, if $i = n-1$, $j = n-2$, and $k_{n-1} = 1$, we obtain $p_{n-1} \geq p_{n-2} + 2 \times l_{max}$. Proceeding as before for $1 < i \leq n$ and $1 \leq j \leq i$, we obtain that $p_2 \geq p_1 + 2 \times l_{max}$ and, hence, $p_n \geq p_1 + 2 \times (n-1) \times l_{max}$.

Since $p_{min} = p_1$, $p_1 < p_n < 2 \times p_1$ holds by assumption. In addition, (1) is assumed to hold for all i and j where $i \neq j$ and $1 \leq k_i \leq n-1$. Hence, for $i = 1$, $j = n$, and $k_1 = 2$, (1) becomes $\mathrm{mod}\left(\frac{2 \times p_1}{p_n}\right) = 2 \times p_1 - p_n = 2 \times p_1 - p_1 - 2 \times (n-1) \times l_{max} \geq 2 \times l_{max}$, which leads to $p_1 \geq 2 \times n \times l_{max}$. The theorem follows. □

Theorem 2 provides a lower bound for p_{min} for the case that $p_{min} < p_i < 2 \times p_{min}$ where $1 \leq i \leq n$, i.e., all p_i have similar values. This is a meaningful choice for p_i since all nodes have the same deadline d_{max}. Note that inter-packet times p_i that strongly differ from each other make it difficult to meet d_{max} with all nodes.

Theorem 3: Let us consider a set of n independent transmit-only nodes. Each of them sends a sequence of n packets within d_{max} and is activated at most once within $t_{max} = 2 \times d_{max}$. If the first packet of a node i is interfered by a packet of node j, in order to guarantee that the next $(n-1)$ packets sent by node i are not interfered again by node j, the following condition must hold for the minimum inter-packet time $p_{min} = \min_{1 \leq i \leq n}(p_i)$:

$$p_{min} \geq 2 \times (n-2) \times (n-1) \times l_{max} + 2 \times l_{max}, \tag{4}$$

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

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 $(n-1)$ packets sent by node i are not interfered again by node j, (1) needs to hold for all i and j where $i \neq j$ and $1 \leq k_i \leq n-1$ 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 $p_{min} = \min_{1 \leq i \leq n}(p_i) = p_1$ and $p_{max} = \max_{1 \leq i \leq n}(p_i) = p_n$ hold.

Let us first consider $i = 1$ and $j = n$. If $k_i = 1$ holds, from (1) we have that $\mathrm{mod}\left(\frac{p_1}{p_n}\right) \geq 2 \times l_{max}$ is equal to $p_1 \geq 2 \times l_{max}$ since $p_1 < p_n$. For $k_i = 2$, from (1) we have that $\mathrm{mod}\left(\frac{2 \times p_1}{p_n}\right) \geq 2 \times l_{max}$ is equal to $2 \times p_1 - p_n = p_1 - 2 \times (n-1) \times l_{max}$, as $\lfloor \frac{2 \times p_1}{p_n} \rfloor = 1$ holds – see again proof of Theorem 2. Similarly, for $k_i = 3$, we have that

Algorithm 1 Searching for the optimum p_{min}

Require: n, l_{max}

1: $p_{min} = 2 \times n \times l_{max}$
2: $p_1 = p_{min}$
3: **for** $i = 2$ to n **do**
4:   $p_i = p_{i-1} + 2 \times l_{max}$
5: **end for**
6: **while** $p_{min} < 2 \times (n-2) \times (n-1) \times l_{max} + 2 \times l_{max}$ **do**
7:   **for** $i = 1$ to n **do**
8:     **for** $k_i = 1$ to $n-1$ **do**
9:       **for** $j = 1$ to n, $j \neq i$ **do**
10:         **if** $\mod\left(\frac{k_i \times p_i}{p_j}\right) < 2 \times l_{max}$ **then**
11:           $p_{min} = p_{min} + l_{max}$
12:           **goto** line 6:
13:         **end if**
14:       **end for**
15:     **end for**
16:   **end for**
17: **end while**
18: return (p_{min})

$\mod\left(\frac{3 \times p_1}{p_n}\right) \geq 2 \times l_{max}$ is equal to $3 \times p_1 - 2 \times p_n = p_1 - 2 \times 2 \times (n-1) \times l_{max} \geq 2 \times l_{max}$, as $\lfloor \frac{3 \times p_1}{2 \times p_n} \rfloor = 1$ holds. For $k_i = n-1$, we have that $\mod\left(\frac{(n-1) \times p_1}{p_n}\right) \geq 2 \times l_{max}$ is equal to $(n-1) \times p_1 - (n-2) \times p_n = p_1 - (n-2) \times 2 \times (n-1) \times l_{max} \geq 2 \times l_{max}$, as $\lfloor \frac{(n-1) \times p_1}{(n-2) \times p_n} \rfloor = 1$ also holds. As a result, we have that $p_1 \geq 2 \times (n-2) \times (n-1) \times l_{max} + 2 \times l_{max}$ which is lower bound for $p_{min} = p_1$ stated in (4). Since $p_n = p_{max}$, note that choosing another j where $1 < j < n$ yields a lower bound that is closer to that of Theorem 2. In other words, the lower bound of (4) is the greatest necessary value of p_{min}. The theorem follows. □

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

Theorem 3 assumes that $(k-1) \times p_{min} < (k-1) \times p_{max} < k \times p_{min}$, for $1 < k \leq n-1$. This assumption simplifies the analysis; however, it leads to the lower bound of (4) which is not necessary the *optimum*, in the sense that there might be a feasible p_{min} that is less than the bound of (4). Note that a shorter p_{min} allows choosing shorter p_i, hence, meeting shorter d_{max}.

On the other hand, we can design an algorithm that iteratively searches for the optimum p_{min} as shown in Alg.1. This algorithm basically starts assuming a p_{min} as per (3) – see line 1 – and increments it by steps of l_{max} – see line 11. For each value of p_{min}, it checks whether Theorem 1 holds or not – see lines 7 to 10. Note that all p_i can be found for the given p_{min} using Lemma 2 – see lines 2 to 4. If Theorem 1 holds, the algorithm stops and returns the current value of p_{min} – see line 18. If not, it increments p_{min} until it reaches the value of (4) which is proven to be safe – see line 6.

Fig. 4 shows the minimum inter-packet times calculated using (4) and found using Alg. 1. Both the calculated and optimized inter-packet times increase quadratically with the number of transmitter nodes; however, Alg. 1 allows reducing the separation between two packets for $n > 3$.

Once we have found p_{min} either by using directly Theorem 3 or Alg. 1, we can use Lemma 2 to obtain each p_i. Note that this choice of p_i allows meeting d_{max}, if and only if the following corollary holds for it.

Corollary 2: Let us consider a set of n independent transmit-only nodes. Each of them will be able to send a sequence of n packets within d_{max} if the following condition holds for $1 \leq i \leq n$:

$$d_{max} \geq (n-1) \times p_i + l_{max}, \tag{5}$$

where p_i is the (constant) inter-packet time of the i-th node.

Proof: A node i sends n packets with constant inter-packet time p_i. If it starts sending its first packet at time t_0, it will send its subsequent $n-1$ packets at times $t_0 + k_i \times p_i$ where $1 \leq k_i \leq n-1$. Node i finishes sending its n-th packet at time $t_0 + (n-1) \times p_i + l_{max}$. The corollary follows. □

V. Experimental Results

In this section, we present our experimental results comparing the proposed technique with other possible methods. To this end, we have performed a simulation based on the OMNeT++ network simulation framework [16] and an extension for mobile and wireless networks named MiXiM [17]. 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 set up correctly in a way that physical effects, such as fading, shading and reflection of radio waves do not cause packet loss and can therefore be neglected. The data rate of transmission has been fixed to 128 kbit/s and the packet size is 3 bytes (8 bits for identifier, 8 bits for data, and 8 bits for check sum). The transmission time of a single packet takes consequently 187.5 µs, i.e., this is the value of l_{max}.

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 algorithms as explained below.

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

- The *single* algorithm is a simple method, in which a single packet is transmitted followed by a transmission

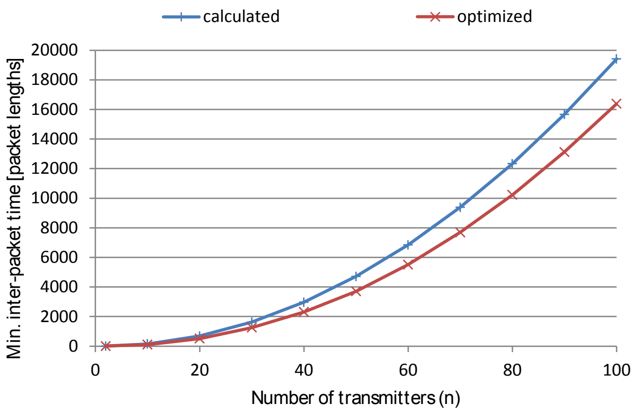

Fig. 4. Minimum inter-packet times in packet lengths (i.e., l_{max}) calculated according to (4) and optimized with Alg. 1. The inter-packet times increase quadratically with the number of transmitter nodes.

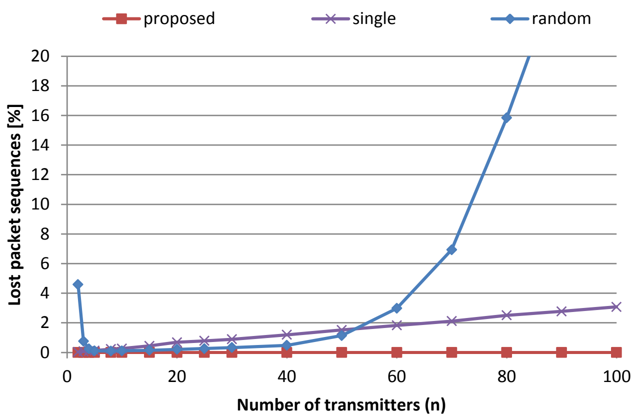

Fig. 5. The amount of lost packet sequences of the three compared algorithms is depicted. Each algorithm was executed for different numbers of transmitter nodes (n); for each value of n, 100,000 different packet sequences have been simulated.

pause of $0.5\,\mathrm{s}$ to $0.6\,\mathrm{s}$. This algorithm is easy to implement and the most resource-efficient on a given node.

- The *random* algorithm transmits n packets with random inter-packet times that are uniformly distributed between $1\,\mathrm{ms}$ and $15\,\mathrm{ms}$. After n packets have been sent, this algorithm implements a transmission pause with a random length between $1\,\mathrm{ms}$ and $15\,\mathrm{ms}$.
- The *proposed* algorithm is our algorithm as presented in Section IV, where inter-packet times are chosen according to (4) and (2).

These three algorithms were evaluated with respect to different numbers of transmitter nodes in our OMNeT++ simulation network. Fig. 5 shows how many packet sequences are lost with the different algorithms as n increases. For the *simple* algorithm, the amount of packet sequences lost increases with a rising number of transmitters. This is due the fact that the capacity of the communication channel is limited because of the fixed transmission interval with a maximum of $0.6\,\mathrm{s}$. Consequently, a higher load leads to a higher collision probability.

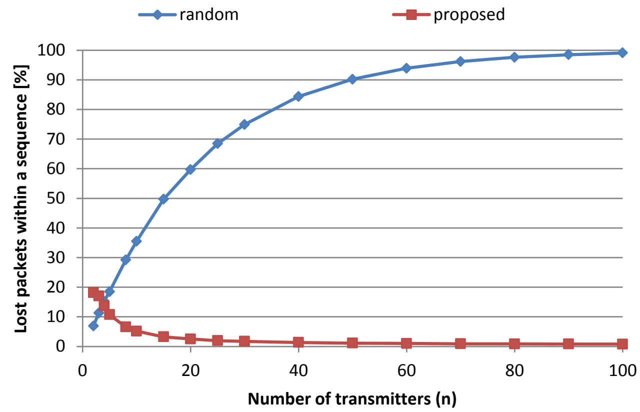

Fig. 6. The amount of lost packets per sequence is depicted for the *random* and our *proposed* algorithm. Each algorithm was executed for different numbers of transmitters (n); each time 100,000 packet sequences have been simulated.

For the *random* algorithm, a different behavior is observed: It first starts with a relatively high collision probability, which decreases until its minimum at $n = 10$ and rises again for higher n. For $n > 53$, the packet loss is higher than for the *single* algorithm. This big amount of packet sequence losses for low n can be explained by the low packet numbers per sequence. As n increases, the collision probability first decreases due to a greater number of packets sent; however, it rises again for $n > 10$. This is because the communication channel starts being flooded with packets and hence a random choice of inter-packet times stops being effective.

Although packet sequences are rarely lost in the shown experiments, both the *single* and the *random* algorithm cannot prevent these losses and, thus, cannot guarantee a fully reliable communication. In contrast to this, our *proposed* algorithm guarantees that at least one packet is always successfully received independent of the number of transmitters. Of course, this comes at the cost of longer inter-packet times as discussed before. The more transmitters are considered, the longer the communication delay, i.e., the longer it takes until a packet reaches its receiver in the worst case. However, on average, not always the last packet of a sequence reaches the receiver and delays should be better than the worst case.

The previous Fig. 5 regarded the loss of packet sequences by the different algorithms. Next we compare how many packets are lost on average within a packet sequence. Fig. 6 shows our results for the *random* and our *proposed* algorithm. Note that the *single* algorithm is not included in this comparison, since it does not send a sequence of packets but just only one.

The *proposed* algorithm starts at a relatively high number of packets lost in a sequence and decreases for greater n. This can be explained by the low number of packets sent at the beginning: loosing one of two packets is equal to 50% packet loss, whereas loosing one out of 100 is equal to only 1% loss. The amount of packet losses in a sequence then decreases for higher n due to the longer inter-packet times.

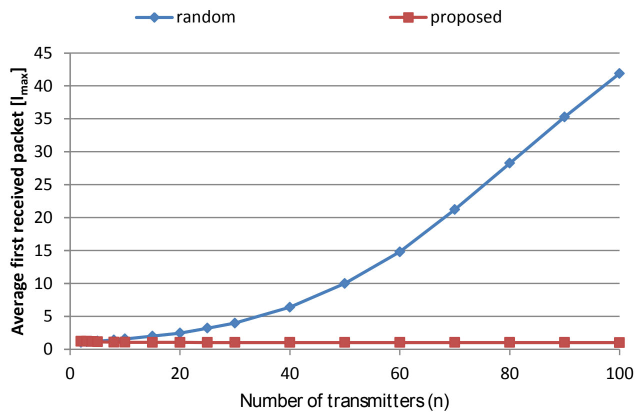

Fig. 7. The average delay for receiving a packet in numbers of l_{max} is depicted for the *random* and our *proposed* algorithm. Each algorithm was executed for different numbers of transmitters (n); each time 100,000 packet sequences have been simulated.

The *random* algorithm starts at a low number of lost packets in a sequence; however, it rapidly increases until it stabilizes at over 95% for $n > 65$ – see Fig. 6. In the case of the *random* algorithm, the communication channel saturates for high n and, hence, the collision probability increases. In other words, choosing random inter-packet times is not efficient for high numbers of transmitters.

Fig. 7 depicts the average message delay for the *random* and *proposed* algorithm, measured by the time from activation of transmission (first packet) until the end of the first received packet. As expected, the delay of the *random* algorithm increases with n due to its inefficient use of the limited channel capacity. In contrast to this, our *proposed* algorithm has a very low delay, which slightly decreases for higher n. The behavior matches with our previous conclusion from Fig. 6. Overall, this means that, to achieve reliability, our *proposed* algorithm transmits a high number of redundant packets.

VI. Concluding Remarks

In this paper, we proposed a technique for designing reliable home-automation networks based on unidirectional nodes, i.e., nodes with either the capability of transmitting or receiving data packets. In contrast to approaches from the literature, where complex communication mechanisms need to be implemented, the proposed technique allows for reliable communication by construction.

More specifically, our technique consists in each transmit-only node sending a sequence of n packets – proven to be the worst case – with constant inter-packet time p_i, being n the number of transmit-only nodes that can interfere with each other. Considering that interference from outside the network is negligible, we proved that it is possible to select inter-packet times such that it can always be guaranteed that at least one packet of each node reaches its corresponding receiver on time.

In addition, we presented a set of experimental results based on an OMNeT++ simulation. Our experiments show that the proposed technique never leads to packet losses and validate the presented analysis.

References

[1] H. Merz, J. Backer, V. Moser, T. Hansemann, L. Greefe, and C. Huebner, *Building Automation: Communication Systems with EIB/KNX, LON and BACnet*. Springer, 2009.

[2] W. S. Lee and S. H. Hong, "KNX – ZigBee Gateway for Home Automation," in *Proceedings of the IEEE Conference on Automation Science and Engineering (CASE)*, 2008.

[3] C. Reinisch, W. Kastner, G. Neugschwandtner, and W. Granzer, "Wireless Technologies in Home and Building Automation," in *Proceedings of the IEEE International Conference on Industrial Informatics (INDIN)*, 2007.

[4] M. Kovatsch, M. Weiss, and D. Guinard, "Embedding Internet Technology for Home Automation," in *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, 2010.

[5] 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.

[6] 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.

[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] 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.

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

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

[11] 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.

[12] R. Cardell-Oliver, A. Willig, C. Huebner, T. Buehring, and A. Monsalve, "Error Control Strategies for Transmit-only Sensor Networks: a Case Study," in *Proceedings of the IEEE International Conference on Networks (ICON)*, 2012.

[13] L. Galluccio, G. Morabito, and S. Palazzo, "TC-Aloha: A Novel Access Scheme for Wireless Networks with Transmit-only Nodes," *IEEE Transactions on Wireless Communications*, vol. 12, pp. 3696–3709, 2013.

[14] 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.

[15] "Delay-Bounded Medium Access for Unidirectional Wireless Links," CISTER - Research Centre in Real-Time and Embedded Computing Systems, Tech. Rep., 2007.

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

[17] 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.