Halmstad University Post-Print

Predictable real-time communications with improved reliability for IEEE 802.15.4 based industrial networks

Kristina Kunert, Elisabeth Uhlemann and Magnus Jonsson

N.B.: When citing this work, cite the original article.

©2010 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Kunert K, Uhlemann E, Jonsson M. Predictable real-time communications with improved reliability for IEEE 802.15.4 based industrial networks. In: 2010 8th IEEE International Workshop on Factory Communication Systems (WFCS). IEEE; 2010. p. 13-22.

DOI: [10.1109/WFCS.2010.5548639](https://doi.org/10.1109/WFCS.2010.5548639)

Copyright: IEEE

Post-Print available at: Halmstad University DiVA

<http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-5434>

Predictable Real-Time Communications with Improved Reliability for IEEE 802.15.4 Based Industrial Networks

Kristina Kunert, Elisabeth Uhlemann and Magnus Jonsson
Centre for Research on Embedded Systems (CERES)
Halmstad University, Halmstad, Sweden
{kristina.kunert, elisabeth.uhlemann, magnus.jonsson}@hh.se

Abstract

Emerging industrial applications requiring reliable wireless real-time communications are numerous. Using existing standards such as IEEE 802.15.4 is essential for reasons of interoperability and cost efficiency. However, since 802.15.4 is unable to provide predictable channel access, real-time guarantees cannot be given. Further, the noisy wireless channel makes reliable communications particularly challenging. By adding a deterministic medium access method and a transport protocol with a truncated retransmission scheme to 802.15.4, we jointly enforce reliability and predictability. We evaluate our solution analytically by real-time schedulability analysis including retransmissions, and by computer simulations. We show that the message error rate can be improved by several orders of magnitude while keeping the utilization penalty at reasonable levels.

1. Introduction

Most applications require a certain degree of predictability and reliability from the underlying communication protocol, but for some applications this requirement is stricter than for others. Industrial automation and monitoring are two such examples. When wireless technologies make their way into the industrial realm, meeting these requirements becomes increasingly challenging, due to the high and time varying noise levels inherent to the wireless medium. Still, the advantages of wireless connectivity (mobility, ease of installation, no cabling, low cost, etc.) are obvious. The use of commercial off the shelf (COTS) components and existing communication standards is close to mandatory in industrial communications as it reduces costs and facilitates interoperability. The IEEE 802.15.4 standard is intended for low-rate wireless personal area networks [1]. Due to its special focus on low-energy consumption, 802.15.4 is often used in wireless sensor networks. Even though many industrial applications involve sensors with limited energy supply, they still have certain quality of service (QoS) requirements, implying the need for predictable and reliable communications. Note that by predictability or real-time constraints we refer to timely treatment of data traffic with deadlines – not low delay communications.

In earlier work [2] we developed a theoretical framework for reliable real-time traffic over a wireless link, reducing the message error rate (MER) through retransmissions, while still being able to guarantee real-time delay bounds. This paper extends that framework to a single-hop network with a logical star topology using 802.15.4. The data traffic model is taken from a typical industrial setting, with short packets, in the order of tens of bytes payload, transmitted bidirectionally between a central master node and its surrounding slaves. As our framework assumes earliest deadline first (EDF) scheduling, a deterministic medium access control (MAC) enforcing this scheduling method is needed. By deterministic MAC we mean that access to the channel is granted in a predictable manner such that there is an upper bound on the delay that can occur before channel access is granted. Note, however, that a deterministic MAC method does not in any way guarantee that the transmission will be successful, in terms of data correctness, once channel access has been granted. In 802.15.4, time can be divided into superframes containing a contention access period (CAP), where medium access is granted according to slotted Carrier Sense Multiple Access, Collision Avoidance (CSMA/CA). The CAP can optionally be followed by a contention free period (CFP) where guaranteed time slots (GTS) can be dedicated to specific nodes in the network through reservations made during the CAP. Since channel access delays cannot be upper-bounded (due to random backoff times in the CAP or failure to reserve a GTS in the CFP), hard deadline guarantees cannot be enforced with plain 802.15.4. As it is vital to keep existing standards without alterations, we introduce a simple master-slave protocol with polling, which can be used on top of 802.15.4. To provide a suitable trade-off between flexibility and efficiency, targeting end-to-end QoS, retransmissions are administrated by a transport layer protocol, and thus no link layer retransmissions are used. The number of retransmissions allowed in the system is regulated by an off-line real-time schedulability analysis. Our approach considerably reduces the MER in a typical industrial network, without jeopardizing deadline guarantees already given to ordinary transmissions. The bandwidth utilization penalty enabling the improvement is reasonable.

The remainder of this paper is organized as follows. Section 2 gives a short overview of related work, whereas Section 3 describes our system model. Section 4

E. Uhlemann is partly funded by the Swedish Governmental Agency for Innovation Systems, Vinnova, through the VINNMER program, www.vinnova.se

provides details on the proposed framework with the deterministic MAC and the transport protocol with retransmissions. In Section 5, the timing and real-time analysis used by the admission control mechanism in the transport protocol is presented. An evaluation using computer simulations of our suggested approach is given in Section 6, before Section 7 concludes the paper.

2. Related works

The analytical framework derived in [2] by the authors, contains both a timing and a real-time analysis for admission control in a truncated retransmission protocol. Real-time analysis based on EDF theory for scheduling periodic tasks on a uniprocessor is well-known [3] and uses a two-step method to check the feasibility of a real-time task set: a utilization check and a workload check. This method has also been mapped to a networking context in [4]. However, the approach of letting the real-time analysis dictate the admission control not only of periodic messages, but also of potential retransmissions of failed packets within messages generated by accepted traffic flows makes [2] quite unique. Prior work in real-time communications often provides only statistical performance without deadline guarantees, e.g. [5, 6]. In [7], a truncated retransmission scheme, retransmitting packets only within the limits of the deadline, is presented, but the lack of queuing analysis implies that the delay cannot be upper-bounded. In [8], hard real-time guarantees are provided for traffic flows including retransmissions, but the approach relies on the existence of traffic regulators in each node. In addition, the analysis lacks timing details and uses flow analysis which in [9] was shown to give lower network utilization than the real-time analysis used in our framework.

Another area of related research deals with enabling real-time traffic in 802.15.4 networks without altering the standard, e.g., [10-13]. However, even though QoS improvements in terms of latency and reliability for time-critical messages have been achieved, average performance rather than the worst case is typically evaluated. The possibility of using GTSs that are part of the IEEE 802.15.4 standard has been considered in [14-18]. But since [14-16] do not provide any timing details, and [14] base their work on delay estimations, deadline guarantees cannot be provided. [17] supports deadline guarantees, but merely for traffic with exactly one transmission per superframe. [18] does include a mathematical schedulability analysis for delay-constraint traffic – but neither [18] nor [14-17] consider retransmissions. One promising improvement of 802.15.4 is the ANGEL Enhancement Layer [19] which adds priority queuing and stochastic service differentiation. However, retransmissions are not treated and so far proof is provided only through measurements. Without further timing analysis, deterministic deadline guarantees cannot be given. Due to the advantages of IEEE 802.15.4 in industrial settings, it has given rise to industrial communication standards such as WirelessHART [20], placing a time-slotted

scheme on top of 802.15.4. While WirelessHART, evaluated in e.g., [21], provides deterministic channel access by dedicated, pre-assigned timeslots, its lack of flexibility prevents efficient use of retransmissions.

In [22] link-layer retransmissions are considered in a centralized polling-based system similar to ours. Typically, moving the retransmission mechanism up or down in the layers of the protocol stack implies a trade-off between efficiency and flexibility. When lower layers handle the retransmissions, the flexibility is reduced, but the efficiency may be enhanced. However, in our case, when the constraint on network utilization only allows a limited number of retransmitted packets, some messages may not benefit from retransmissions (e.g., two packets in a message require retransmissions but only one packet can be allowed by the admission control mechanism). Hence, the network efficiency is higher if the bandwidth used for retransmissions is available to all nodes and all messages with a potential to reduce the MER.

3. System model

Considering some typical industrial applications, we assume a wireless network with a modest number of slave nodes connected to a central master node in a logical star. All slaves are within transmission range of the master with approximately equal propagation delay and all communication takes place via the master node using one single frequency channel. Dictated by the IEEE 802.15.4 standard, a sleep cycle is included and the chosen data rate is 250 kbps. The frequency channel should be chosen to minimize the interference between collocated 802.15.4 and 802.11 networks. There are four channels for 802.15.4 which do not overlap with the 802.11b (European) channel selection [1]. We further assume the availability of CRC checksums in all packets and a negligible probability of lost acknowledgements. Since no link layer retransmissions are needed these are assumed to be switched off, a feature available e.g., with the IEEE 802.15.4 compliant ChipCon CC2420 [23].

The data traffic model is taken from a typical industrial setting, with short packets, in the order of tens of bytes payload, transmitted both ways between a central master node and its surrounding slaves. All traffic is of a periodic nature and modelled as traffic flows, called real-time channels (RTCs). Each flow τ_i , $i = 1, 2, \dots$, is characterized by five parameters: sending node, receiving node, period, message length, and relative deadline: $\tau_i = \{S_i, R_i, P_i, C_i, D_i\}$. As our retransmission scheme is placed at the transport layer, the deadline indicates the transport layer deadline. The parameters defining each RTC are assumed to be known to the master. All traffic flows are independent of each other which implicates that several flows can coexist between two arbitrary nodes. We assume, however, that the master is involved in all transmissions, collecting sensor data from the slaves or distributing control data to the slaves.

Since only those RTCs for which channel access can be guaranteed before their deadlines expire, are admitted

Figure 1. Superframe structure without GTSS

into the system, no deadlines are missed due to late arrival. Instead our performance measures are deadline misses due to *MER* and *utilization*, i.e., collective bandwidth of accepted RTCs. Both measures are crucial for most industrial control processes.

4. The retransmission framework

To guarantee a maximum delay for real-time traffic over an 802.15.4 network, we need to extend the MAC layer to make its access time predictable. To increase reliability through retransmissions, we add a transport layer enabling admission control based on real-time analysis.

4.1. Deterministic medium access

A network based on IEEE 802.15.4 has two different modes: beacon enabled with slotted CSMA/CA medium access and non-beacon enabled with unslotted CSMA/CA medium access. While the former is available only for networks with a logical star topology, the latter is intended for logical mesh topologies. As our work assumes a logical star topology, only the beacon enabled mode is considered below.

The beacon is broadcasted at regular time intervals (beacon intervals) by a network coordinator. Every beacon interval can be divided into an active phase, called the superframe, and an optional inactive phase, designed to enable a power saving mode, Fig. 1. The maximum allowed length of the superframe is equivalent to the beacon interval length. The superframe is divided into 16 equally sized timeslots, where the beacon is the starting point of the first of these slots. The superframe contains a CAP where medium access is granted according to slotted CSMA/CA. The CAP can be optionally followed by a CFP where GTSSs can be dedicated to specific nodes in the network. A node has to send an explicit request during the CAP for allocation of one of the GTSSs. The maximum number of GTSSs is seven, and thus the rest of the timeslots are occupied by the CAP. The possibility of using GTSSs, which was added to the standard to enable applications with a need for guaranteed bandwidth, is only available for star topologies.

Since the CAP is CSMA/CA-based, the random back-off procedure makes channel access non-deterministic which means that it cannot be used for critical real-time traffic since channel access times and queuing delays cannot be upper-bounded. Further, as the reservation of a GTS is done during the CAP, the worst case delay for the allocation request cannot be upper-bounded. In addition, all GTS requests are treated according to a first-come, first-serve rule, not taking deadlines into consideration and deallocation of GTSSs has to be made explic-

itly; otherwise the same time slot stays allocated to the same node during all future superframes.

To provide an upper bound on the queuing delay, we extend the basic 802.15.4 MAC method with an EDF polling scheme, resulting in a deterministic MAC protocol upper-bounding the medium access delay. Note that a TDMA-based mechanism similar to WirelessHART could have been chosen, but the benefit of polling is that it can be based on instantaneous conditions, allowing retransmissions to enter the schedule faster, as they do not have to wait for the next dedicated time slot. Since our approach is placed on top of IEEE 802.15.4, the division of the superframe into CAP and CFP is not necessary and the entire beacon interval (minus the inactive interval) can be used for real-time communication. The functionality in the added MAC layer is limited since all scheduling is located in the transport layer, which hands down packets according to EDF. Due to the added polling protocol, the medium is always free when a packet is handed down to the 802.15.4 MAC layer. This means that no random back-off delays will occur, rendering deterministic medium access possible. Also acknowledgements and retransmissions are entirely handled by the transport layer and erroneous packets are simply identified by the MAC layer and reported to the transport layer where appropriate measures are taken.

4.2. Real-time transport layer retransmission scheme

The key functionalities of the transport layer are the real-time automatic repeat request (ARQ) scheme and the EDF scheduling of all traffic (including all retransmissions). Traditional transport layer tasks such as flow and congestion control can be omitted since both the traffic flow model and the schedulability test limit the amount of traffic that needs to be handled in a timely fashion. The choice of EDF for traffic scheduling was guided by its well-researched analytical framework, its high ability to find feasible schedules, and its ability to schedule a retransmission with an early deadline amongst ordinary transmissions with longer deadlines. Placing the retransmission scheme in the transport layer rather than in the link layer enables end-to-end reliability and increases the flexibility, since retransmissions can be granted on a message level and also shared between several nodes.

In order to accommodate retransmissions, the transport layer deadline (D_i) is divided into two parts, called ordinary and retransmission deadline, $D_{ord,i}$ and $D_{retr,i}$. In addition to the regular traffic flows, we introduce flows solely dedicated for retransmissions, the retransmission real-time channels (ReRTCs), defined by sender, receiver, period, message length, and relative deadline: $\tau_{re,j} = \{S_{re,j}, R_{re,j}, P_{re,j}, C_{re,j}, D_{re,j}\}$. The period of a ReRTC is merely a system parameter, indicating the minimum interval allowed between two consecutive retransmissions using this ReRTC. This is necessary to upper-bound the bandwidth used for retransmitted packets. The relative retransmission deadline indicates the maximum time allowed for any specific retransmission.

Transport layer acknowledgements are only used by

Figure 2. Flow chart for transport protocol in the master node

slaves when acknowledging the successful reception of a packet from the master. The retransmission of erroneous packets to the master is administered (polled for) by the master without the need for acknowledgements. Consequently, no link layer acknowledgements provided by 802.15.4 are used. To enable retransmissions, the slave nodes keep all packets until the deadline has passed.

Master node functionality (Fig. 2): At start up, the master node receives (e.g., by the system manager) the necessary traffic specifications and the retransmission parameters. For all the accepted RTCs (having passed the EDF feasibility check described in Section 5), deadline guarantees can be given. Based on the parameters of the RTCs, the master creates an EDF sorted queue of polling packets (to collect data from slaves) and data packets (to be transmitted from the master to slaves). The master will treat the packets one by one, determining for each one, their type and waiting time, accounting either for the reception of a requested packet from a slave ($T_{timeout_poll,i}$) or for the acknowledgement for its own transmitted data packet ($T_{timeout_data,i}$). If and only if this time is as long as, or shorter than, the time left until the end of the contention based phase, the master will send its polling or data packet. Not until this timeout has ended is the master allowed to treat the next packet in the EDF queue. After the timeout, the master node transport layer checks for newly arrived packets from its underlying MAC layer, i.e., a packet from the polled slave or simply an acknowledgement for a correctly received

Figure 3. Time-sequence diagram for slave-to-master communication

packet. Before scheduling a retransmission of an absent or erroneous packet, the master checks if the maximum number of allowed retransmission attempts ($N_{attempt}$) is reached and if a retransmission channel that has not been used during the length of one retransmission period (P_{re}) is available. Only if both requirements are fulfilled, a retransmission is scheduled. Otherwise, the packet and all other packets belonging to the same message are discarded. Checking retransmission channel availability is crucial to assure that no previously guaranteed deadlines are violated. The timeout values are calculated as:

$$T_{timeout_poll,i} = T_{procM} + T_{poll} + T_{prop} + T_{procS} + T_{data,i} + T_{prop} + T_{procM_CRC} + T_{margin} \quad (1)$$

$$T_{timeout_data,i} = T_{procM} + T_{data,i} + T_{prop} + T_{procS_CRC} + T_{ACK} + T_{prop} + T_{procM} + T_{margin} \quad (2)$$

T_{procM} denotes processing time needed by the master before the packet can leave the node, while T_{procS} is the time needed by the slave to identify a requested packet. In addition, the processing times include delays introduced by the underlying 802.15.4 protocol, i.e., the necessary interframe space and time for carrier sensing. Additional processing time T_{proc_CRC} is needed for error checking when the received packet is a data packet. T_{ACK} , T_{poll} and $T_{data,i}$ are the transmission time of an acknowledgement, a polling and a data packet, respectively. T_{prop} denotes a common propagation delay, while a time margin (T_{margin}) is introduced to compensate for inexact synchronization or propagation variations and serves as a safety margin between the expected reception of a data or acknowledgement packet and the actual timeout. For a graphical illustration of equations (1) and (2), Fig. 3 and 4 provide time-sequence diagrams for slave-to-master and master-to-slave communication respectively.

Slave node functionality (Fig. 5): Slave nodes are assumed to be simpler and lack logic for retransmission administration and schedulability analysis. All ordinary transmissions are stored in a software queue which is continuously populated by periodic application data packets. Furthermore, a small random access memory is present for sent packets awaiting possible retransmission. If no retransmission is requested before a packet's deadline, it will be discarded. As acknowledgements are sent in direct response to the reception of a data packet,

Figure 4. Time-sequence diagram for master-to-slave communication

no memory for storing those is necessary. The transport layer at a slave node distinguishes between three types of packets, data, retransmission requests and polling packets. Arriving packets have already been error checked by the MAC layer. Data packets trigger the generation of acknowledgements, while polling packets trigger the selection of a particular data packet from the local random access memory (retransmission) or the software queue (ordinary transmission).

5. Timing and real-time scheduling analysis

In this section, both timing analysis and scheduling analysis are described. Compared to the framework in [2], the introduction of the 802.15.4 network makes certain adaptations to the analysis necessary.

5.1. Timing parameters

In accordance with the standard, the beacon interval time, T_{BI} , denotes the amount of time from the start of the transmission of the beacon until the next beacon starts, Fig. 6. T_{BI} has three main parts (not taking into consideration the optional phase with GTS which is not used in our solution), the transmission time of the beacon, T_{beacon} , the CAP, T_{CAP} , and an inactive phase, T_{sleep} .

Figure 5. Flow chart for transport protocol in the slave node

Figure 6. Superframe timing parameter

The interval described by the beacon interval minus the inactive phase, is also called the superframe time, T_{SF} . The actual length of the CAP usable for transmitting packets is further shortened by the equivalent of the time it would take to transmit one packet which would not be able to finish completely before the end of the CAP. Due to the nature of the polling protocol, this time consists of: the transmission time of one data packet (T_{data}), either a polling (T_{poll}) or an acknowledgement (T_{ACK}) packet, and two propagation delays (T_{prop}). Further, processing delays in the master and in the slave (T_{procM} , T_{procS}), one of them also including error checking (T_{procM_CRC} , T_{procS_CRC}) and the time constant (T_{margin}) compensating for, e.g., small differences in time synchronization, all have to be taken into account. Therefore, T_{CAP} is :

$$T_{CAP} = T_{BI} - T_{sleep} - T_{beacon} - T_{data} - \max\{T_{ACK}, T_{poll}\} - 2 \cdot T_{prop} - T_{margin} - \max\{T_{procM} + T_{procS_CRC}, T_{procS} + T_{procM_CRC}\} \quad (3)$$

To compensate for the periods of time when no packet transmission is possible, i.e., during the sleeping phase and during the time just before the sleeping phase when it is too late to initiate a new packet transmission, we introduce the concept of experienced bit rate, r^e . Since the actual bit rate r can only be used for a limited time during each beacon interval, r^e represents the average bit rate experienced by the data traffic as if the medium would have been available during the entire beacon interval. It is calculated as:

$$r^e = r \cdot \frac{T_{CAP}}{T_{BI}} \quad (4)$$

Due to this scaling of the bit rate, the experienced transmission time of data ($T_{data,i}^e$), acknowledgement ($T_{ACK,i}^e$), and polling ($T_{poll,i}^e$) packets is given by the length of a packet divided by the experienced bit rate as:

$$T_{data,i}^e = \frac{L_{data,i}}{r^e}, \quad T_{ACK,i}^e = \frac{L_{ACK,i}}{r^e}, \quad T_{poll,i}^e = \frac{L_{poll,i}}{r^e} \quad (5)$$

where L_x indicates the length of the respective type of packet. For the sake of simplicity, all data packets are assumed to be of equal length, even if the framework supports packets of arbitrary lengths. The new transmission times will influence the timeout values calculated earlier in (1) and (2) which now become:

$$T_{timeout_poll,i}^e = T_{procM} + T_{poll,i}^e + T_{prop} + T_{procS} + T_{data,i}^e + T_{prop} + T_{procM_CRC} + T_{margin} \quad (6)$$

$$T_{timeout_data,i}^e = T_{procM} + T_{data,i}^e + T_{prop} + T_{procS_CRC} + T_{ACK,i}^e + T_{prop} + T_{procM} + T_{margin} \quad (7)$$

5.2. Timing analysis

Since each RTC is defined as $\tau_i = \{S_i, R_i, P_i, C_i, D_i\}$, where $i = 1, 2, \dots, Q$, this yields Q RTCs existing simultaneously in the system. Correspondingly, the ReRTCs, defined as $\tau_{re,j} = \{S_{re,j}, R_{re,j}, P_{re,j}, C_{re,j}, D_{re,j}\}$ with $j = 1, 2, \dots, M$, gives M concurrently existing retransmission channels. Section 4.2 described the division of the deadline D_i into one deadline for ordinary transmissions $D_{ord,i}$ and one for retransmissions $D_{retr,i}$. The deadlines specified up until now give the point in time by which the packet has to have arrived at the receiver transport layer. However, the real-time analysis described below requires as input the maximum queuing delay, i.e., the delay bound until the packet has to have left the sender. In order to determine the maximum queuing delay for an ordinary transmission $d_{ord,i}$, several parameters have to be subtracted from the corresponding deadline $D_{ord,i}$. The worst-case situation is when, firstly, the transmission of a packet has just started, while secondly, a packet with higher priority (shorter deadline) arrives to the EDF queue, and thereafter, thirdly, the sleep cycle starts. This means that the newly arrived packet will first have to wait during the entire transmission time of this previous packet, i.e., $\max\{T_{timeout_poll,i}, T_{timeout_data,i}\}$. Further, these parameters have to be subtracted twice, since all transmissions have to have ended before the end of the active phase according to the 802.15.4 standard. Therefore, the high priority packet cannot be transmitted if the time remaining in the current phase is too short for its entire transmission to finish. Finally, we also have to include the length of the sleep phase (T_{sleep}) and the subsequent beacon transmission (T_{beacon}). This results in the following maximum queuing delay for slave-to-master ($d_{ordM \rightarrow S}$) or master-to-slave ($d_{ordS \rightarrow M}$) packets:

$$d_{ordM \rightarrow S} = D_{ord,i} - (2 \cdot T_{prop} + T_{data} + T_{ACK} + T_{procM} + T_{procS_CRC} + T_{margin}) - T_x - T_{sleep} - T_{beacon} \quad (8)$$

$$d_{ordS \rightarrow M,i} = D_{ord,i} - (2 \cdot T_{prop} + T_{poll} + T_{data} + T_{procM_CRC} + T_{procS} + T_{margin}) - T_x - T_{sleep} - T_{beacon}, \quad (9)$$

where the delay introduced by the previous packet, for reasons of readability is denoted T_x :

$$T_x = 2 \cdot T_{prop} + \max\{T_{data} + T_{ACK} + T_{procM} + T_{procS_CRC} + T_{poll} + T_{data} + T_{procM_CRC} + T_{procS}\}. \quad (10)$$

The max function is used as we cannot know if the previous packet was slave-to-master or master-to-slave communication, and we have to assume the worst case.

A corresponding calculation of maximum queuing delay has to be made for retransmission deadlines. The division of the application deadline into $D_{ord,i}$ and $D_{retr,i}$ is guided by a system parameter limiting the amount of time spent on retransmissions. It is identical for all channels, so without loss of generality we set $D_{retr,i} = D_{retr}$. The number of allowed retransmission attempts per packet ($N_{attempt}$) is a system parameter that depends on the degree of reliability requested by the application. As the retransmission deadline D_{retr} has to harbour all retransmission attempts, the deadline per retransmission

for any channel, $D_{retrM \rightarrow S}$ or $D_{retrS \rightarrow M}$, denoting master-to-slave and slave-to-master RTCs respectively, is

$$D_{retrM \rightarrow S,i} = D_{retrS \rightarrow M,i} = \frac{D_{retr}}{N_{attempt}} \quad (11)$$

So to guarantee the deadlines of individual retransmission attempts, we further define the deadline of all retransmission channels to be the length of one retransmission attempt, i.e.,

$$D_{re} = D_{re,j} = D_{retrM \rightarrow S,i}. \quad (12)$$

As for ordinary RTCs, we also have to isolate the maximum queuing delay for packets sent on a retransmission channel. Taking the whole deadline per retransmission attempt, and thereby the deadline of the retransmission channel, as a starting point we have to subtract the same parameters as for ordinary packets. Again the maximum queuing delay will be experienced when a packet arrives in the queue just as another packet with a longer deadline has started to transmit, and after that the sleeping phase will commence. In accordance with (8) and (9), we calculate the maximum queuing delay for a packet of channel i in the master-to-slave communication case, $d_{retrM \rightarrow S,i}$ as:

$$d_{retrM \rightarrow S,i} = D_{re} - 2 \cdot T_{prop} - T_{data} - T_{ACK} - T_{procM} - T_{procS_CRC} - T_{margin} - T_x - T_{sleep} - T_{beacon}. \quad (13)$$

In slave-to-master communication, i.e., when the master is polling the slave for the retransmission of data, the maximum queuing delay of this packet ($d_{retrS \rightarrow M,i}$) is:

$$d_{retrS \rightarrow M,i} = D_{re} - 2 \cdot T_{prop} - T_{poll} - T_{data} - T_{procM_CRC} - T_{procS} - T_{margin} - T_x - T_{sleep} - T_{beacon}. \quad (14)$$

5.3. Real-time schedulability analysis

EDF theory for scheduling periodic real-time tasks on a uniprocessor in a real-time system [3] is well-known and uses a well-proven two-step method to check the feasibility of a real-time task set: a utilization check and a workload check. Excluding a small number of special cases, both tests are necessary, and none of them is in itself sufficient to prove that a given task set (or in our case a channel set) is schedulable, i.e., will miss none of its deadlines. Thus, only a positive response from both checks will provide us with the deadline guarantees we are targeting. The EDF scheduling theory has been mapped to a networking context in [4]. Note that EDF scheduling in a uniprocessor context is assumed to be preemptive, while in a communication context preemption can only happen on a message basis, not on a packet basis, i.e., once the transmission of a packet has begun, it cannot be stopped. This is why the packet blocking time was introduced when calculating the maximum queuing delay above. The utilization and workload checks only have to be executed whenever a new RTC is requesting network allocation; otherwise, as soon as the tests have returned positive answers, all deadlines for both ordinary transmissions and retransmissions can be guaranteed as long as system or network parameters are maintained. Assuming a single channel network, the utilization con-

dition is basic: the utilization must not be higher than 1 at any time. This condition holds true even in the presence of a sleeping phase and packet blocking times due to the introduction of the experienced bit rate. According to EDF scheduling theory, adapted to our context, the utilization is given by:

$$U = \sum_{i=1}^V \left(\frac{T_{\text{timeout_poll},i}^e \cdot N_i}{P_i} \right) + \sum_{i=1}^W \left(\frac{T_{\text{timeout_data},i}^e \cdot N_i}{P_i} \right) + \sum_{i=1}^X \left(\frac{T_{\text{timeout_retr_poll},i}^e}{P_{re,i}} \right) + \sum_{i=1}^Y \left(\frac{T_{\text{timeout_retr_data},i}^e}{P_{re,i}} \right). \quad (15)$$

V and W denotes the number of RTCs from master to slave and slave to master, respectively, and $V+W=Q$, which is the total number of RTCs in the network. Accordingly, X and Y denotes the number of ReRTCs from master to slave and slave to master, respectively, and $X+Y=M$, which is the total number of ReRTCs in the network. N_i indicates the number of packets that RTC i is sending each instance of its period P_i . This parameter is absent in sum three and four, as retransmission channels always only send one packet during their period $P_{re,i}$. $T_{\text{timeout_poll},i}^e$ and $T_{\text{timeout_data},i}^e$ are used according to their definitions in (6) and (7), while the timeout parameters for the retransmissions are introduced solely for ease of understanding and are defined as $T_{\text{timeout_retr_poll},i}^e = T_{\text{timeout_poll},i}^e$ and $T_{\text{timeout_retr_data},i}^e = T_{\text{timeout_data},i}^e$.

The workload function $h(t)$ is summing all transmission times of all messages by all RTCs in all periods, only including message instances that have their absolute deadline before time t . This point in time t is measured from the start of the hyperperiod of all RTCs, which is defined as the length of time from when all periods of all RTCs and ReRTCs start at the same time, to when they do so again. The hyperperiod can easily be calculated by the least common multiple of all the channels' periods. Finding the start of the hyperperiod and measuring t from that point in time is essential as the concurrent start of all periods has been proven to be the worst case in terms of workload on the network, and therefore results in the worst case delay [24-26]. Only when taking into consideration worst case situations will we be able to upper bound the delay experienced by the packets. We calculate the workload function as follows:

$$h(t) = \sum_{\substack{i \in [1,V] \\ d_{ordS \rightarrow M,i} \leq t}} \left(1 + \left\lfloor \frac{t - d_{ordS \rightarrow M,i}}{P_{T,i}} \right\rfloor \right) \cdot (T_{\text{timeout_poll},i}^e \cdot N_i) + \sum_{\substack{i \in [1,W] \\ d_{ordM \rightarrow S,i} \leq t}} \left(1 + \left\lfloor \frac{t - d_{ordM \rightarrow S,i}}{P_{T,i}} \right\rfloor \right) \cdot (T_{\text{timeout_data},i}^e \cdot N_i) + \sum_{\substack{i \in [1,X] \\ d_{retrS \rightarrow M,i} \leq t}} \left(1 + \left\lfloor \frac{t - d_{retrS \rightarrow M,i}}{P_{re,i}} \right\rfloor \right) \cdot T_{\text{timeout_retr_poll},i}^e + \sum_{\substack{i \in [1,Y] \\ d_{retrM \rightarrow S,i} \leq t}} \left(1 + \left\lfloor \frac{t - d_{retrM \rightarrow S,i}}{P_{re,i}} \right\rfloor \right) \cdot T_{\text{timeout_retr_data},i}^e. \quad (16)$$

These four summation terms correspond to the terms in the utilization calculation in (15). Term one and two calculate the workload imposed by master-to-slave and slave-to-master RTCs, respectively, while term three and four add the maximum workload introduced by the retransmission channels. This workload function is used in the feasibility test introduced in [24-26]. Although the test is formally basic, and simply tests that $h(t) \leq t, \forall t$, its computational complexity is high compared to the utilization check due to the introduction of continuous time. In [27] the complexity was lowered by reducing continuous time to discrete numbers at which the workload function had to be calculated and further only to the instances of time when a packet deadline occurs. These checks can be limited to a bounded interval of time between the start of the first hyperperiod and the end of the first so-called busyperiod (P_{busy}), where a busyperiod is any period within a hyperperiod during which the link is not idle. This can formally be specified as:

$$t \in \bigcup_{i=1}^V \{m \cdot P_i + d_{ordS \rightarrow M,i} : m = 0, 1, 2, \dots\} + \bigcup_{i=1}^W \{n \cdot P_i + d_{ordM \rightarrow S,i} : n = 0, 1, 2, \dots\} + \bigcup_{i=1}^X \{a \cdot P_{re,i} + d_{retrS \rightarrow M,i} : a = 0, 1, 2, \dots\} + \bigcup_{i=1}^Y \{b \cdot P_{re,i} + d_{retrM \rightarrow S,i} : b = 0, 1, 2, \dots\}, \quad (17)$$

where $t \in [1; P_{\text{busy}}]$.

6. Simulation study

In order to evaluate the improvement in MER, a computer simulator was implemented in Matlab. The simulator includes the 802.15.4 MAC layer, our MAC layer extension with polling, and the transport layer admission control and retransmission protocol. We are assuming a logical star topology with single-hop traffic, a data rate of 250 kbps going both ways between one central master and nine connected slave devices. It should be noted that the number of requested real-time channels is the limiting factor, not the number of nodes. We further assume a packet length of 120 bits for data, acknowledgement, and polling packets, which is the minimum size allowed in the 802.15.4 standard. Since the ACK packets used by the standard are not able to carry any additional data (as e.g. the identity of a real-time channel) it is necessary to use small data packets instead. Assuming a propagation speed through air of $3 \cdot 10^8$ m/s, we set the propagation delay to 0.3 μ s, corresponding to a distance of approximately 100 m. The beacon frame length is set to 122.88 ms. Two different lengths of superframes are tested, 61.88 ms and 30.72 ms. This choice of values means that during approximately 50% or 75% of the beacon interval, respectively, the nodes will be in an energy saving sleeping mode to prolong battery lifetime.

TABLE I: Traffic specification

Ordinary traffic			
Traffic class	P_i (ms)	D_i (ms)	C_i (bits)
No 1	600	600	480
No 2	1000	1000	600
Retransmissions			
ReRTC	P_{re} (ms)	D_{re} (ms)	C_{re} (bits)
No 1	600	200	120
$N_{attempt} = 2$ $M = 2$ or 4 or 8			

As the standard is designed with low-rate traffic in mind, the simulated traffic classes are chosen trying to emulate the traffic properties of industrial applications. The traffic class definitions are found in Table I and for all simulations, the traffic channels are randomly chosen from these different traffic classes. The source and destination nodes are randomized amongst all nodes, however, the master node has to be either source or destination for each channel. For both random choices we use a uniform distribution. For the retransmissions, we assume a set of identical retransmission channels with the number of retransmission attempts, $N_{attempt}$, fixed to 2, while the number of ReRTCs, M , is varied from 2 to 4 and 8.

The channel model adopted is Gilbert-Elliot with two states corresponding to two bit error rates, 10^{-2} and 10^{-4} . The probability of state changes are set to 1% for leaving the state with low error probability, and 50% for leaving the high error probability state at each time step in the simulation. One time step corresponds to one packet exchange between the master and one of the slaves. The channel model has been chosen to emulate a situation where most of the time, the BER is relatively low, but during short periods of time, the medium suffers from a higher noise leading to bursts of errors due to, e.g., a mobile obstacle interfering with the radio propagation, or a transmission on a collocated network using an interfering frequency channel. Knowing the BER P_b and the message length in bits (C), the average message error rate can be calculated by $MER = 1 - (1 - P_b)^C$. Due to the fact that our messages consist of several packets, which in their turn consist of several bits, even a relatively low BER will result in a fairly high MER.

The performance measures evaluated in the simulator are network utilization and MER, with and without packet retransmissions, as a function of the number of requested RTCs for different lengths of the sleeping phase. Note, however, that one traffic flow, i.e., one RTC, is not equivalent to one node, as several traffic flows can be allocated between two nodes. When evaluating the utilization and the MER, only accepted RTCs are included and RTC i is accepted only if the real-time analysis can find a feasible schedule for the case when RTC i is included. Otherwise RTC i is rejected. The actual utilization value generated by one RTC also includes the polling or ACK packet and the propagation delay of both packets. When we use retransmissions, it still holds true that only accepted original RTCs are included in the

TABLE II: Maximum network utilization by retransmission channels

Sleep \ M	2	4	8
Appr. 50% of BI	0.0066	0.0132	0.0264
Appr. 75% of BI	0.0136	0.0272	0.0544

utilization calculation. However, the introduction of ReRTCs into the network reduces the maximum utilization possible by original RTCs. Knowing the experienced bit rate (r^e), the packet length (C_{re}) of a ReRTC, its period (P_{re}), and the total number of ReRTCs (M), the maximum bandwidth utilization by ReRTCs (U_{re}) is:

$$U_{re} = \frac{\left(\frac{C_{re} + L_{ACK/poll}}{r^e} + 2 \cdot T_{prop} \right)}{P_{re}} \cdot M. \quad (18)$$

The results for the different cases are shown in Table II.

The results of the evaluations are presented in Fig. 7 and 8 for sleeping phases of 50% and 75% of the beacon interval length, respectively, where subfigure (a) shows the network utilization during the active phase and subfigure (b) the MER. In all figures, the solid line depicts the polling scheme without retransmissions, while the dashed lines show the results when using our transport layer retransmission scheme. In Fig. 7(a) the utilization curves coincide for low numbers of RTCs when the schedulability is not influenced by the introduction of retransmission channels due to the system being lightly loaded. The utilization curve for the case of no retransmissions indicates that saturation of the network with maximum utilization is reached at 88 requested channels. The utilization penalty introduced by retransmission channels is about 40 percentage points for all simulated numbers of ReRTCs. Consequently, with the introduction of retransmission channels, the utilization by ordinary traffic is reduced considerably. The reason for this is that the limiting factor for channel admission is not the utilization constraint, but the workload constraint. Due to the shortening of the deadline when dividing it into ordinary and retransmission deadline, the traffic becomes more difficult to schedule as soon as retransmissions are introduced. The increasing MER with increasing number of RTCs in subfigures (b) when using retransmissions can be explained by the fact that fewer RTCs will require fewer retransmissions, which means that the probability of having a ReRTC available is higher. When the number of requested RTCs is very small, the different message lengths of the traffic classes and the bursty error model lead to a fluctuation in the MER which can be seen in the slight sinking of the curve before starting to rise along with the number of requested RTCs. Without retransmissions, the bandwidth can be used to a higher degree, but the MER lies at almost 10^{-1} , which means that one in ten messages will be erroneous and has to be discarded. The MER is reduced with a higher value of M . With two ReRTCs, the MER improves to about 10^{-2} , while four ReRTCs result in a MER of 10^{-3} for few RTCs and 10^{-2} for higher numbers of requested RTCs. The MER could be pushed down all the way to almost

Figure 7. Simulation results for sleeping phase of 50% of the beacon interval

10^{-4} for few requested RTCs with eight Re RTCs, but settles between 10^{-3} and 10^{-2} when the number of RTC requests is high.

When studying the amount of real-time traffic flows schedulable in the network at a sleeping phase of about 50% of the beacon interval, approximately 88 RTCs is a very high number for industrial networks. If the application does not require such a large amount of traffic flows, it is possible to instead increase the length of the sleeping phase in order to prolong the battery lifetime of the sensor network. Therefore, in the second simulation setting, the length of the sleeping phase was increased to approximately 75% of the beacon interval, depicted in Fig. 8(a) and (b). Again we can see that the utilization curves in (a) follow each other when the number of RTCs is small. The saturation point without retransmissions is reached at about 42 requested RTCs which still allows for a reasonable large network. The initial MER value without retransmissions is again at 10^{-1} , with a bandwidth utilization of approximately 0.98. The intro-

duction of retransmission channels introduces a utilization penalty of about 45 percentage units, relatively independent of the number of Re RTCs. The reason for this is again that as soon as retransmissions are introduced the traffic becomes more difficult to schedule since the deadlines are shortened by dividing them into ordinary and retransmission deadlines. However, introducing more Re RTC does not imply much more penalty. The utilization curves for 2 and 4 Re RTCs coincide in Fig. 8(a). However, the improvement in MER varies with the number of Re RTCs. Two retransmission channels offer improvements of around one order of magnitude, while four retransmission channels for low requested RTC numbers can reach an improvement of between one and two orders of magnitude. Eight Re RTCs decrease the MER to between 10^{-4} and 10^{-3} . Again the curve for 8 Re RTCs experiences the already described fluctuation for a low number of requested RTCs due to the varying message lengths of the RTCs and the bursty error model.

Figure 8. Simulation results for sleeping phase of 75% of the beacon interval

7. Conclusion

We present an approach for increasing predictability and reliability in a single-hop wireless industrial network using IEEE 802.15.4. Our framework provides deadline guarantees for both ordinary transmissions and retransmissions and is able to decrease the MER by several orders of magnitude while keeping the utilization penalty at reasonable levels. Our solution adds a deterministic MAC mechanism and a transport level protocol with a truncated retransmission scheme, which both can be used on top of the IEEE 802.15.4 standard. Placing retransmissions at the transport level enables a high degree of flexibility, since retransmissions of individual packets can be granted based on the overall success possibility of the entire message. These additions enable the use of the existing standard for industrial applications with special demands on predictability and reliability.

References

- [1] LAN/MAN Standards Committee of the IEEE Computer Society, "Part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low rate wireless personal area networks (WPANs)," *IEEE Computer Society*, 2006.
- [2] M. Jonsson and K. Kunert, "Towards reliable wireless industrial communication with real-time guarantees," *IEEE Trans. on Ind. Inf.*, vol. 5, no. 4, pp. 429-442, Nov. 2009.
- [3] C.L. Liu and J.W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *J. of the ACM*, vol. 20, no. 1, pp. 46-61, 1973.
- [4] H. Hoang and M. Jonsson, "Switched real-time Ethernet in industrial applications - deadline partitioning," *Proc. Asia-Pacific Conf. Comm. (APCC '03)*, Penang, Malaysia, Sept. 2003, vol. 1, pp. 76-81.
- [5] S. Pejhan, M. Schwartz and D. Anastassiou, "Error control using retransmission schemes in multicast transport protocols for real-time media," *IEEE/ACM Trans. on Networking*, vol. 4, no. 3, pp. 413-427, 1996.
- [6] E. Uhlemann and L.K. Rasmussen, "Incremental redundancy deadline dependent coding for efficient wireless real-time communications," *Proc. IEEE Conf. Emerging Techn. & Factory Automation (ETFA '05)*, Catania, Italy, Sept. 2005, pp. 417-424.
- [7] M.M. Butt, "Provision of guaranteed QoS with hybrid automatic repeat request in interleaved division multiple access systems," *Proc. IEEE Singapore Int. Conf. Comm. Syst.*, Singapore, Oct. 2006, pp. 1-5.
- [8] G. Giancola, S. Falco and M.G. Di Benedetto, "A novel approach to error protection in medium access control design," *Proc. IEEE Conf. Mobile & Wirel. Comm. Networks (MWCN '02)*, Stockholm, Sweden, Sept. 2002, pp. 337-341.
- [9] X. Fan, M. Jonsson and J. Jonsson, "Guaranteed real-time communication in packet-switched networks with FCFS queuing," *Computer Networks*, vol. 53, no. 3, pp. 400-417, Feb. 2009.
- [10] A. Koubâa, M. Alves, B. Nefzi and Y.-Q. Song, "Improving the IEEE 802.15.4 slotted CSMA/CA MAC for time-critical events in wireless sensor networks," *Proc. Workshop Real-Time Networks (RTN '06)*, Dresden, Germany, July 2006.
- [11] T. Kim, D. Lee, J. Ahn and S. Choi, "Priority toning strategy for fast emergency notification in IEEE 802.15.4 LR-WPAN," *Proc. Joint Conf. Comm. & Informations (JCCI '05)*, Daegu, Korea, Apr. 2005.
- [12] T. Kim and S. Choi, "Priority-based delay mitigation for event-monitoring IEEE 802.15.4 LR-WPANs," *IEEE Comm. Letters*, vol. 10, no. 3, pp. 213-215, Mar. 2006.
- [13] F. Shu, "Performance evaluation of the IEEE 802.15.4 CSMA-CA protocol with QoS differentiation," *Proc. Int. Conf. Intelligent Sensors, Sensor Networks and Inform. Processing (ISSNIP '08)*, Sydney, Australia, Dec. 2008, pp. 475-480.
- [14] A. Koubâa, M. Alves, E. Tovar and A. Cunha, "An implicit GTS allocation mechanism in IEEE 802.15.4 for time-sensitive wireless sensor networks: theory and practice," *Real-Time Systems*, vol. 39, 2008, pp. 169-204.
- [15] A. van den Bossche, T. Val and E. Campo, "Proposition of a full deterministic medium access method for wireless network in a robotic application," *Proc. IEEE Vehicular Techn. Conf. (VTC-Spring '06)*, Melbourne, Australia, May 2006, vol. 1, pp. 279-283.
- [16] M. Hameed, H. Trsek, O. Graeser and J. Jasperneite, "Performance investigation and optimization of IEEE 802.15.4 for industrial wireless sensor networks," *Proc. IEEE Int. Conf. Emerging Techn. & Factory Automation (ETFA '08)*, Hamburg, Germany, Sept. 2008, pp. 1016-1022.
- [17] J. Franconne, G. Mercier and T. Val, "A simple method for guaranteed deadline of periodic messages in 802.15.4 cluster cells for control automation applications," *Proc. IEEE Conf. Emerging Techn. & Factory Automation (ETFA '06)*, Sept. 2006, Prague, Czech Rep., pp. 270-277.
- [18] C. Na, Y. Yang and A. Mishra, "An optimal GTS scheduling algorithm for time-sensitive transactions in IEEE 802.15.4 networks," *Computer Networks*, vol. 52, 2008, pp. 2543-2557.
- [19] D. Kipnis, A. Willig, J.-H. Hauer and N. Karowski, "The ANGEL IEEE 802.15.4 enhancement layer: Coupling priority queueing and service differentiation," *Proc. Europ. Wirel. Conf. (EW '08)*, June 2008, Prague, Czech Rep.
- [20] HART Communication Foundation, "HART Field Communication Protocol Specification, Rev. 7.0," Sept. 2007.
- [21] S. Petersen and S. Carlsen, "Performance evaluation of WirelessHART for factory automation," *Proc. IEEE Conf. Emerging Techn. & Factory Automation (ETFA '09)*, Sept. 2009, Palma de Mallorca, Spain.
- [22] G. Gamba, F. Tamarin and A. Willig, "Retransmission strategies for cyclic polling over wireless channels in the presence of interference," *Proc. IEEE Conf. Emerging Techn. & Factory Automation (ETFA '09)*, Palma de Mallorca, Spain, Sept. 2009.
- [23] Chipcon, *2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*, Chipcon Products Texas Instruments 2004.
- [24] M. Spuri, "Analysis of deadline scheduled real-time systems," *Techn. Rep. No. 2772, INRIA*, France, 1996.
- [25] S.K. Baruah, A.K. Mok and L.E. Rosier, "Preemptively scheduling hard-real-time sporadic tasks on one processor," *Proc. IEEE Real-Time Syst. Symp. (RTSS '90)*, Lake Buena Vista, FL, USA, Dec. 1990, pp. 182-190.
- [26] S.K. Baruah, L.E. Rosier and R.R. Howell, "Algorithms and complexity concerning the preemptive scheduling of periodic, real-time tasks on one processor," *Real-Time Syst.*, vol. 2, no. 4, pp. 301-324, 1990.
- [27] J.A. Stankovic, M. Spuri, K. Ramamritham and G.C. Buttazzo, *Deadline scheduling for real-time systems - EDF and related algorithms*, Kluwer Academic Publishers, Boston, MA, USA, 1998.