

Exploiting Time and Frequency Diversity in IEEE 802.15.4 Industrial Networks for Enhanced Reliability and Throughput

Kristina Kunert, Magnus Jonsson, and Elisabeth Uhlemann
CERES – Centre for Research on Embedded Systems
Halmstad University, Box 823, SE-30118 Halmstad, Sweden
{kristina.kunert; magnus.jonsson; elisabeth.uhlemann}@hh.se

Abstract

Industrial networks based on IEEE 802.15.4 are spreading, even though the joint requirement on predictability and reliability from industrial applications is hard to fulfil in wireless networks, and the data rate of IEEE 802.15.4 is rather low. With the goal of providing real-time guarantees, with increased reliability and throughput, we propose two multichannel network architectures based on IEEE 802.15.4 with predictable medium access, real-time analysis admission control and transport layer retransmissions. We evaluate the architectures in terms of reliability, utilization, delay, complexity, scalability and energy efficiency. The evaluations show that throughput and reliability can be enhanced through redundancy and concurrency in the frequency domain.

1. Introduction

Industrial applications are characterized by concurrent demands on both reliability and predictability. Reliability is defined as the ability to provide a low packet error rate whereas predictability is ability to meet strict real-time deadlines. The choice of wireless networks as the underlying communication medium makes it especially challenging to fulfil these communication requirements. Due to advantageous energy saving techniques, IEEE 802.15.4 [1] has become an interesting choice for wireless sensor networks in industrial contexts, e.g., the industrial standards WirelessHART [2] and ISA100.11a [3]. WirelessHART enables predictability by placing a time-slotted medium access control (MAC) scheme on top of IEEE 802.15.4, but lacks flexibility preventing efficient use of retransmissions for increased reliability. In earlier work [4] we developed a theoretical framework for reliable real-time traffic where reliability was increased through retransmissions, while still guaranteeing real-time delay bounds. This framework was applied to an IEEE 802.15.4 based single-hop network in [5]. However, using retransmissions reduces the data rate, an already limited resource in IEEE 802.15.4.

Our goal in this paper is to increase the effective bandwidth with IEEE 802.15.4 by introducing diversity not only in the time domain through retransmissions, but also in the frequency domain using multichannel com-

munication. Aiming towards a multichannel master-slave network, two different network architectures with different single-hop star topologies are considered. The most straightforward design choice is to equip each node with several IEEE 802.15.4 transceivers fixed to different frequency channels. In order not to overlap with frequencies in IEEE 802.11b based networks, four frequency channels are available [1]. The number of slaves in the network is assumed to be higher than the number of frequency channels F , but does not have to be. The second network architecture uses one transceiver at each slave, but tuneable over all F frequencies. The effective bandwidth is increased through the use of multiple frequency channels, while still keeping the advantageous energy saving behaviour of IEEE 802.15.4, which includes a synchronized sleep phase for all nodes in the network. The first design choice is more costly in terms of hardware and energy consumption, but is likely to provide higher reliability and increased throughput as compared to a traditional single-channel IEEE 802.15.4 network. The second design choice is less costly in terms of hardware, but tuning to different frequencies reduces the network efficiency even if it provides increased robustness through frequency diversity. We investigate the performance of the proposed architectures in terms of reliability, predictability, bandwidth utilization, delay, scalability, hardware cost, software complexity and energy efficiency.

2. Related works

A network based on IEEE 802.15.4 has two different modes: beacon-enabled with slotted Carrier Sense Multiple Access, Collision Avoidance (CSMA/CA) MAC and nonbeacon-enabled with unslotted CSMA/CA. 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. The length of the beacon interval (T_{BI}) is defined as the interval from when one beacon transmission starts until it starts again. This interval can then be further divided into the superframe (T_{SF}) and the sleeping phase (T_{sleep}). The superframe contains a contention access period (CAP), which 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 IEEE 802.15.4.

Prior work in real-time communications typically provides only statistical performance guarantees, e.g. [6] which targets improvement of average delay. Other related work enables real-time traffic in IEEE 802.15.4 networks without altering the standard, e.g., [7]-[8]. However, even though improvements in terms of latency and reliability for time-critical messages have been achieved, average performance rather than worst case is typically evaluated. [9] supports deadline guarantees and uses retransmissions for increased reliability, but merely for traffic with exactly one transmission per superframe. Our approach has no such limitations, but instead lets the number of retransmissions allowed in the system be regulated by an off-line real-time schedulability analysis. We thereby increase reliability without jeopardizing deadline guarantees already given to ordinary transmissions. The bandwidth utilization penalty enabling the improvement is reasonable, but the low data rate is limiting the applicability. To compensate for this, we introduce a multichannel IEEE 802.15.4 network with predictable master-slave medium access and transport layer retransmissions. A predictable MAC method provides an upper bound on the channel access delay which is necessary for real-time communications so that scheduling methods like earliest deadline first (EDF) can be enforced. Note, however, that a predictable MAC method does not guarantee error free transmission. Earlier efforts of multichannel protocols include both distributed and centralized MAC protocols for wireless ad-hoc and sensor networks [10]-[12]. However, even if multiple frequencies exist, they are not used concurrently, but merely alternatively and none of the reviewed protocols can provide predictable medium access. [13] presents a multichannel MAC protocol for general wireless networks, but does not include retransmissions, while the scheduling mechanism presented in [14] does not consider communication errors. Examples of multichannel MAC protocols for IEEE 802.15.4, e.g. [15] and [16], do not consider real-time guarantees or reliability.

3. System Model

To model a typical wireless industrial network, the number of communicating nodes is chosen to be reasonably low, i.e., in the order of tenths of nodes. All slave nodes are placed within transmission range of a master node which acts as the central controller. Due to the limited size of the network, the propagation delay between the master and any of the slaves is approximately equal. Basing our framework on IEEE 802.15.4, we include a sleep cycle and assume a data rate of 250 kbps.

With industrial applications in mind, we choose a data traffic model with short packets and periodic bidirectional traffic between the master node and its slaves. As in [4], we model the traffic as flows, defined by the parameters: sending node (S_i), receiving node (R_i), period

(P_i), message length (C_i), and deadline (D_i). These flows are denoted real-time channels (RTCs) $\tau_i = \{S_i, R_i, P_i, C_i, D_i\}$, where $i = 1, 2, \dots$. Each flow is independent of all other flows, enabling the existence of several flows between the master and any slave. To model traffic typical for industrial applications, which e.g. collect data from sensors or distribute control data to actuators, each flow is assumed to either originate or terminate at the master node. D_i refers to the transport layer deadline as our retransmission scheme provides end-to-end reliability at the transport layer. The link layer retransmissions in the standard are therefore not needed and hence switched off, as possible with e.g. the IEEE 802.15.4 compliant ChipCon CC2420 transceiver [17]. Retransmission real-time channels, ReRTCs, are introduced into the network, modelled as traffic flows and defined by the same parameters as ordinary traffic flows $\tau_{re,j} = \{S_{re,j}, R_{re,j}, P_{re,j}, C_{re,j}, D_{re,j}\}$. Their period is the minimum mandatory interval between two activations of a certain ReRTC. This interval is defined at the design stage of the network, thereby providing an upper bound on the bandwidth used for retransmissions. The deadline of the ReRTC denotes the maximum time that one retransmission is allowed to take. To cater for retransmissions, the transport layer deadline in each RTC is divided into an ordinary ($D_{ord,i}$) and a retransmission deadline ($D_{retr,i}$), where $D_{ord,i} + D_{retr,i} = D_i$. All packets contain a perfect CRC checksum to determine when a retransmission is needed.

4. Evaluation Criteria

To evaluate the proposed multichannel network architectures, the following metrics are used:

Reliability Since the admission control based on real-time analysis ensures that only RTCs for which a deadline guarantee can be provided are allowed into the network, the deadline miss ratio due to late packet arrival is always zero. Nevertheless, deadline misses due to message errors can still occur, making the message error rate (MER) an important performance measure. However, reliability is not only increased through the time-diversity and data redundancy provided by retransmissions but also through frequency and hardware redundancy introduced by multiple transceivers, all contributing to increased network robustness and reliability.

Bandwidth utilization capability The capability of utilizing as much of the available bandwidth as possible depends on the flexibility of channel allocation and the degree of concurrency that can be achieved, i.e., how well the frequency diversity is exploited. A higher number of concurrent channels available at a node may lead to increased bandwidth and lower delay if properly exploited.

Delay The worst case delay depends on the architecture and the resulting blocking behaviour of packets to and from a node. However, also average delay and the delay jitter are of interest. Delay also connects to concurrency.

Scalability The network can be limited by several factors. Here we consider the relation between the number of slave nodes, frequencies and transceivers per node.

Hardware and software complexity simply indicates the estimated hardware cost per node in terms of number of transceivers and the relative complexity of the channel allocation algorithm (at run-time), taking into account the number of concurrent frequency channels per node.

Energy usage is assuming a simple relation between the number of transceivers per node and its energy usage.

5. Network Architectures

To extend the single-channel framework presented in [5] to a multichannel wireless network, we equip, in the first architecture, each node with F transceivers fixed to F different frequency channels. The second multichannel network architecture equips the master node with F fixed transceivers, assuming it to be more powerful and less resource restricted than the slave nodes, but only one transceiver per slave, tuneable over all F frequencies.

5.1. Baseline: Single-channel network

As a starting point for the multichannel protocols we take the single-channel protocol introduced in [5]. The MAC mechanism in [5] is an extension of the IEEE 802.15.4 MAC protocol with a polling mechanism providing predictable channel access and the ability to schedule traffic according to EDF. The master has access to all traffic specifications at start-up and sorts all packets to slaves (or polling messages asking for packets from slaves) according to EDF. The master then either sends a polling packet to a slave and waits for a response (a data packet) or it sends itself a data packet to a selected slave and awaits an acknowledgement (ACK). If the master does not receive an ACK (or a data packet) before a time-out – corresponding to the transmission of the data and ACK packets (or poll and data packets) and some processing and propagation delays – the polling or data packet is saved until its ordinary or retransmission deadline has elapsed. When the (ordinary or retransmission) deadline of the whole message has expired, the master knows the exact number of packets in this message that have to be retransmitted. In case more retransmission attempts are allowed and a retransmission channel is available, the necessary retransmissions are scheduled. The number and frequency of retransmissions allowed in the system is regulated by the off-line real-time schedulability analysis.

5.2. Multichannel I: F fixed transceivers

The availability of F concurrent frequency channels to access all nodes at any given time enables different organizational approaches. Either we can allow information dissemination via all frequencies or have one dedicated control or retransmission frequency channel, while the remaining $F-1$ channels are used for data dissemination. Alternatively, we can use some of the channels for delay-constrained real-time traffic and the remaining for best effort data traffic. With appropriate scheduling, we can guarantee concurrent transmission of at most F packets, i.e. a pure EDF handling of the F packets with the earliest deadlines. All schemes enable diversion of traffic to another frequency, avoiding unnecessarily high

packet loss in the presence of high interference at one particular frequency channel.

When choosing to not allocate a special control frequency, the master can schedule which information that should be sent on each frequency at each time instance. The beacon, sent out periodically by the underlying IEEE 802.15.4 protocol, can be broadcasted on all frequencies so that all slaves receive it on all frequencies. Based on the address field in the packet header the slaves assess the relevance of each packet themselves. As the master polls each slave, and has global knowledge of the scheduling, no medium access collisions due to several nodes trying to use the same frequency occurs. In fact, the polling procedure happens on all frequencies concurrently. The absence of an expected data packet or ACK from a particular slave at the end of a message deadline leads the master to check the availability of retransmission channels – unless the missing packet has already reached the maximum number of allowed retransmissions – and schedules retransmissions in exactly the same manner as an ordinary transmission.

Alternatively, one of the frequency channels could be allocated for control traffic only. The control channel would be carrying polling messages asking for data packets from the slaves, to be transmitted on another frequency. This leaves $F - 1$ frequencies for actual data traffic, both ordinary transmissions and retransmissions, allowing $F - 1$ concurrently transmitted data packets, to and from the master. Data packets transmitted by the master do not need to be announced on the control channel as slaves are listening on all frequencies anyway, and therefore are ready to receive. However, the following ACK needs to use a designated data channel, as slaves do not have access to the control channel. Network wide polling information can be broadcasted over the control frequency in parallel with the ordinary data traffic over the remaining frequencies. However, as the control channel is used exclusively for polling traffic, it may not be sufficiently utilized. The same holds true for a variant where retransmissions use a dedicated frequency channel. Further, system vulnerability is increased by grouping polling packets or retransmissions onto one single frequency. Consequently, the alternative with a dedicated control frequency is not treated further here.

5.3. Multichannel II: one tuneable slave transceiver

In this architecture, the slaves have one transceiver each, tuneable over F different frequencies, while the master has F transceivers, each one fixed to a single frequency. Since a slave only can be reached through one frequency at a time, the master has to consider traffic flow dependencies when allocating the frequencies and only one RTC having a particular node as its destination can be allocated concurrently. This also results in a maximum of one packet that can be guaranteed EDF treatment. The fewer concurrent frequencies per slave, the harder it is for the master to use the bandwidth efficiently and exploit the frequency diversity, as dependencies amongst the different real-time traffic channels come into play. Given the real-time constraint, the packet with the shortest deadline is allocated first, but

the remaining $F-1$ frequencies can only carry packets to or from slaves not involved in another transmission. A packet with the second earliest deadline involving the same slave cannot be scheduled concurrently. The master therefore goes through the EDF queue, identifying the $F-1$ packets with earliest deadlines, but pertaining to individual slaves. When having identified which (at most) F packets, including retransmission packets, are to be sent during the next time instance, the master broadcasts this information on all frequencies. Next, all slaves tune their transceivers to the respective allotted frequency channel and the data transfer takes place in parallel. We need to tune for each packet (rather than for each message or each flow) to cater for packets or retransmissions with earlier deadlines entering the queue. The reason for having tuneable transceivers, rather than one dedicated frequency, for each slave is that the network is likely to contain more slaves than available frequency channels. Tuning, however, introduces additional delay, firstly when the slaves, after having processed the control information, tune to the frequency on which they are supposed to transmit or receive next. Secondly, the ACK for packets sent from the master to slaves needs to be transmitted before the next control message can be broadcasted, prolonging the period of control messages. For slow devices the tuning time between two frequency channels can be as long as the transmission time for one packet [16], which means that the possible bandwidth utilization is decreased substantially. However, progress in hardware development will enable fast tuning devices, decreasing this disadvantage in the future, and thus this architecture represents an interesting alternative.

6. Timing details and analysis

A timing analysis for a single-channel IEEE 802.15.4 network was conducted in [5], but adaptations are necessary for it to be applicable also to multiple channels.

6.1. Timing parameters

Recall that the superframe includes both the actual beacon and the CAP. Beacon transmission time and CAP length are denoted as T_{beacon} and T_{CAP} , respectively. We assume that no GTSs are used as they are neither necessary nor sufficient to guarantee an upper delay bound. Not all of the CAP can be used for transmissions as each packet transmission has to be finished before the start of the sleeping phase. Hence, the actual (useable) length of the CAP is reduced by the maximum time it takes to send one packet ($T_{timeout}$). Therefore, T_{CAP} can be calculated as:

$$T_{CAP} = T_{BI} - T_{sleep} - T_{beacon} - T_{timeout} \quad (1)$$

As described below, the value of $T_{timeout}$ is different for different architectures, as the MAC protocol is adapted accordingly.

To account for the sleeping phase we introduce the concept of experienced bit rate (r^e), which is a scaled down bit rate as compared to the real rate r . The scaling factor depends on the length of time provided for communications, i.e. T_{CAP} , in comparison to the length of the whole beacon interval, as:

$$r^e = r \cdot \frac{T_{CAP}}{T_{BI}} \quad (2)$$

The experienced transmission time of data (T_{data}^e), polling (T_{poll}^e) and ACK (T_{ACK}^e) packets scales accordingly. For multichannel case I, r is the combined bit rate of all channels, i.e., F times the bit rate of one frequency.

6.1.1. Single-channel case

$T_{timeout}$ in the single-channel case is calculated by taking into consideration the following sequence of time intervals. The master starts by processing the packet before transmission (T_{procM}), followed by the actual transmission (T_{poll} or T_{data}) and propagation time (T_{prop}). Upon arrival the packet is processed by the slave, and in case it is a data packet, a CRC check is done (T_{procS} or T_{procS_CRC}). Thereafter, the slave transmits a response (T_{packet} or T_{ACK}), resulting in another propagation delay. The master processes the packet and, in case it is a data packet, does a CRC check (T_{procM} or T_{procM_CRC}). Taking into consideration the possibility of delay propagation variations or inexact synchronization between master and slaves, a short time margin (T_{margin}) is added as a safety margin between the time when the master expects a packet and its actual time of arrival. The equations from [5] are repeated here for convenience:

$$T_{timeout_poll,i} = T_{procM} + T_{poll} + T_{prop} + T_{procS} + T_{data,i} + T_{prop} + T_{procM_CRC} + T_{margin} \quad (3)$$

$$T_{timeout_data,i} = T_{procM} + T_{data,i} + T_{prop} + T_{procS_CRC} + T_{ACK} + T_{prop} + T_{procM} + T_{margin} \quad (4)$$

Eq.(3) matches the case of the master polling for a data packet from a slave. Eq.(4) assumes a data packet sent by the master which then awaits an ACK from a slave.

6.1.2. Multichannel I

Having F fixed transceivers in all nodes results in exactly the same chain of events as for the single channel case, so eq.(3) and (4) are directly applicable.

6.1.3. Multichannel II

With a tuneable transceiver at the slaves and F fixed transceivers at the master, further delays have to be considered. The master starts by processing the packets to be sent and then broadcasts a control packet on all frequencies. Considering the frequency limitations, the control packet concerns a maximum of four frequencies, upper-bounding the packet length. After the transmission time and propagation delay of the control packet, the slaves process the received packet and then tune their transceivers to the respective frequency indicated in the control packet, introducing tuning delay (T_{tune}). We then have to differentiate between two cases: traffic flows from the slave to the master and vice versa. For the first case, the slave simply sends the requested data packet to the master on the indicated channel, resulting in transmission and propagation delay, processing and CRC checking delay at the master, and an additional time margin as stated in 6.1.1. For the second case of traffic flows originating at the master, the master waits for the slave to tune and then sends a data packet. The slave processes the received packet and conducts a CRC

check, before it answers with an ACK, which then is processed by the master. Even here the time margin is added. In this architecture, the extra polling packet introduced also for data packets sent from the master to the slave results in additional transmission and propagation delay, processing time, plus tuning time:

$$T_{timeout_poll,i} = T_{procM} + T_{poll} + T_{prop} + T_{procS} + T_{tune} + T_{data,i} + T_{prop} + T_{procM_CRC} + T_{margin}. \quad (5)$$

$$T_{timeout_data,i} = T_{procM} + T_{poll} + T_{prop} + T_{procS} + T_{tune} + T_{data,i} + T_{prop} + T_{procS_CRC} + T_{ACK} + T_{prop} + T_{procM} + T_{margin}. \quad (6)$$

Here (5) defines the timeout value for traffic flows from slave to master and (6) vice versa.

6.2. Timing analysis

The deadline D_i of an RTC τ_i is divided into an ordinary deadline $D_{ord,i}$ and a retransmission deadline $D_{retr,i}$. This holds true for all considered architectures. To be able to use the deadline parameter in the real-time analysis provided below, the deadline has to be shortened further, such that it corresponds to the latest possible point in time when the packet must have left the sender in order to arrive in time at its destination. This parameter is called maximum queuing delay ($d_{ord,i}$, $d_{retr,i}$) and can be calculated by subtracting from the given deadline a number of constants constituting a worst case situation. The worst case waiting time for a packet with the highest deadline occurs when it arrives to the EDF queue just as a packet with a longer deadline has started sending, and thereafter, when that packet's transmission process has ended, the remaining time until the network enters the sleep mode is not sufficient to accommodate a packet transmission. As a result, a high priority packet has to wait two timeouts, one for each case stated above, plus the following sleep phase and beacon transmission. Therefore $d_{ord,i}$ for traffic flows from master to slave (M→S) and (S→M) are:

$$d_{ordM \rightarrow S} = D_{ord,i} - T_{sleep} - T_{beacon} - T_{timeout_data,i} - \max\{T_{timeout_poll,i}, T_{timeout_data,i}\}, \quad (7)$$

$$d_{ordS \rightarrow M} = D_{ord,i} - T_{sleep} - T_{beacon} - T_{timeout_poll,i} - \max\{T_{timeout_poll,i}, T_{timeout_data,i}\}. \quad (8)$$

As ReRTCs are scheduled as regular RTCs, the same adaptation of deadline to maximum queuing delay has to be made. The deadline of the ReRTC is a system parameter and is related to D_{retr} as follows. The system specifies a maximum number of retransmission attempts ($N_{attempt}$) necessary to meet the reliability demands of the application. These $N_{attempt}$ attempts have to be finished before D_{retr} expires, implying a per-retransmission deadline (D_{re}). Assuming equal deadline partitioning we have

$$D_{re} = \frac{D_{retr}}{N_{attempt}} \quad (9)$$

To guarantee the deadlines for each of those retransmission attempts, we set the deadline of each ReRTC to D_{re} . Isolating the queuing delay for retransmissions is again done by subtracting the worst case situation parameters and differs between M→S and S→M packets as:

$$d_{retrM \rightarrow S} = D_{re} - T_{sleep} - T_{beacon} - T_{timeout_data,i} - \max\{T_{timeout_poll,i}, T_{timeout_data,i}\} \quad (10)$$

$$d_{retrS \rightarrow M} = D_{re} - T_{sleep} - T_{beacon} - T_{timeout_poll,i} - \max\{T_{timeout_poll,i}, T_{timeout_data,i}\} \quad (11)$$

The queuing deadline is calculated in this way for both the single-channel and multichannel case II, using the appropriate timeout intervals. However, for the multichannel architecture I, we have to consider the fact that the total bit rate is F times the bit rate of one frequency. This can lead to a fault in the calculation where the transmission time of a packet suddenly is only $1/F$ times as long as in reality. Even if the network can send F times as many packets as with one frequency, each packet itself is never sent faster than the actual bit rate per frequency. Therefore we have to subtract a further constant from the deadline which is the actual timeout interval of one packet minus the wrongly perceived timeout interval of $1/F$ times the actual timeout. This results in a specific queuing delay for the multichannel case I of:

$$d_{ordM \rightarrow S} = D_{ord,i} - T_{sleep} - T_{beacon} - T_{timeout_data,i} - \max\{T_{timeout_poll,i}, T_{timeout_data,i}\} - \left(1 - \frac{1}{F}\right) \cdot T_{timeout_data,i} \quad (12)$$

$$d_{ordS \rightarrow M} = D_{ord,i} - T_{sleep} - T_{beacon} - T_{timeout_poll,i} - \max\{T_{timeout_poll,i}, T_{timeout_data,i}\} - \left(1 - \frac{1}{F}\right) \cdot T_{timeout_poll,i} \quad (13)$$

6.3. Real-time scheduling analysis

The real-time analysis used in our earlier work is based on EDF scheduling of periodic real-time tasks on a uniprocessor [18], and includes both a check of the utilization and of the workload imposed on the processor by the scheduled tasks. This analysis originally assumed fully preemptive tasks, but has been adapted for communication [19] where preemption is only possible after the complete transmission of one packet, i.e., once a packet has started to be sent, it cannot be stopped in order to send a higher priority packet. Therefore the blocking time of one packet has been introduced into the calculation of the queuing deadline of a packet. The real-time analysis is used each time a new RTC requests access to the network, to decide if its inclusion is possible without jeopardizing any deadline guarantees already given, while also guaranteeing the deadline of the new RTC.

The utilization check simply tests if the admitted RTCs and ReRTCs are not using more bandwidth than available. For a single-channel network the condition is simply that the utilization U must not be greater than 1. In multichannel case I, the utilization must not be greater than F due to the concurrency of the F frequencies. Because of the sleeping phase, these constraints only hold true if we calculate U using the experienced bit rate:

$$U = \sum_{i=1}^V \left(\frac{T_{timeout_poll,i}^e \cdot N_i}{P_i} \right) + \sum_{i=1}^W \left(\frac{T_{timeout_data,i}^e \cdot N_i}{P_i} \right) + \sum_{i=1}^X \left(\frac{T_{timeout_retr_poll,i}^e}{P_{re,i}} \right) + \sum_{i=1}^Y \left(\frac{T_{timeout_retr_data,i}^e}{P_{re,i}} \right), \quad (14)$$

where

$$T_{timeout_poll,i}^e = T_{procM} + T_{poll}^e + T_{prop} + T_{procS} + T_{data,i}^e + T_{prop} + T_{procM_CRC} + T_{margin}, \quad (15)$$

$$T_{timeout_data,i}^e = T_{procM} + T_{data,i}^e + T_{prop} + T_{procS_CRC} + T_{ACK}^e + T_{prop} + T_{procM} + T_{margin}. \quad (16)$$

The timeout parameters for the retransmissions are solely introduced to increase clarity and are defined as $T_{timeout_retr_poll,i}^e = T_{timeout_poll,i}^e$ and $T_{timeout_retr_data,i}^e = T_{timeout_data,i}^e$. N_i indicates the number of packets per message for RTC i . As retransmissions are done per packet, this parameter is assumed to be equal to one for all ReRTCs and is therefore omitted from the equation. V and W indicate the number of S→M and M→S RTCs, respectively, and X and Y the number of S→M and M→S ReRTCs, respectively. As the timeout calculation differed for the multichannel network with one tuneable transceiver at each slave, the proper experienced timeout interval is:

$$T_{timeout_poll,i}^e = T_{procM} + T_{poll}^e + T_{prop} + T_{procS} + T_{tune} + T_{data,i}^e + T_{prop} + T_{procM_CRC} + T_{margin} \quad (17)$$

$$T_{timeout_data,i}^e = T_{procM} + T_{poll}^e + T_{prop} + T_{procS} + T_{tune} + T_{data,i}^e + T_{prop} + T_{procS_CRC} + T_{ACK}^e + T_{prop} + T_{procM} + T_{margin} \quad (18)$$

Part two of the real-time analysis checks the workload imposed on the network by the traffic channels. It is defined by the workload function $h(t)$ which is the sum of all transmission times of all messages by all RTCs and ReRTCs, including those message instances with an absolute deadline before time t . Time t starts with zero at the beginning of the hyperperiod (hp), i.e., the instance in time when all RTC periods start simultaneously. The hp ends when all periods begin simultaneously again. The reason for choosing this starting point is that to provide real-time guarantees, we have to analyse the worst case workload, since the data traffic then will experience the worst delay. The simultaneous start of all periods has been proven to be the worst case in [20]-[22], where also the workload function is used to test the feasibility of task sets. The $h(t)$ function is given as

$$h(t) = \sum_{\substack{i \in [1,V], \\ d_{ordS \rightarrow M,i} \leq t}} \left(1 + \left\lfloor \frac{t - d_{ordS \rightarrow M,i}}{P_{T,i}} \right\rfloor \right) \cdot (T_{timeout_poll,i}^e \cdot N_i) + \sum_{\substack{i \in [1,W], \\ d_{ordM \rightarrow S,i} \leq t}} \left(1 + \left\lfloor \frac{t - d_{ordM \rightarrow S,i}}{P_{T,i}} \right\rfloor \right) \cdot (T_{timeout_data,i}^e \cdot N_i) + \sum_{\substack{i \in [1,X], \\ d_{retrS \rightarrow M,i} \leq t}} \left(1 + \left\lfloor \frac{t - d_{retrS \rightarrow M,i}}{P_{re,i}} \right\rfloor \right) \cdot T_{timeout_retr_poll,i}^e + \sum_{\substack{i \in [1,Y], \\ d_{retrM \rightarrow S,i} \leq t}} \left(1 + \left\lfloor \frac{t - d_{retrM \rightarrow S,i}}{P_{re,i}} \right\rfloor \right) \cdot T_{timeout_retr_data,i}^e \quad (19)$$

where the first two terms give the workload of the ordinary traffic flows (S→M and M→S), and the last two terms the workload of the retransmission channels (S→M and M→S). Depending on if we calculate the workload of the single-channel case or the multichannel

cases, only the timeout parameter will vary. Only multichannel case II uses a different timeout interval, while multichannel case I and the single-channel case have an identical timeout value. Further, the queuing delay calculations differ and will thus also influence the result of the workload function. When testing feasibility, the constraint $h(t) \leq t, \forall t$, has to be met. In multichannel case I, however, the concurrency of frequencies leads to the constraint of $h(t) \leq F \cdot t, \forall t$. In [23], the computational complexity of these tests was reduced and a sufficient subset of points in time at which the function has to be checked was provided.

7. Performance evaluation

We evaluated the two proposed multichannel network architectures and compared them to the results for the single-channel network in [5]. To evaluate the reliability in terms of MER and the bandwidth utilization capability, we have implemented a simulator in MatLab. The reference network has a logical star topology with one master node and 10 slave nodes. A maximum transmission range of 100 m was assumed, leading to a propagation delay of 0.3 μ s. The underlying IEEE 802.15.4 protocol is in beacon mode using a bit rate of 250 kbps. A beacon frame length of 122.88 ms was assumed to keep the same parameters as in our earlier work for the sake of comparability. Sleeping phases corresponding to 50% and 75% of beacon interval time were simulated, i.e., the superframe length was 61.88 and 30.72 ms, respectively. For tuneable transceivers a tuning delay of 131 μ s was chosen based on results from [24]. The number of concurrent frequencies, F , was set to four.

We assume a bursty error behaviour modelled by a typical two-state Markov model (Gilbert-Elliot). The two states have a bit error rate of 10^{-2} and 10^{-4} respectively, the state changing probabilities are 0.5 and 0.99, and the state hold times equal one timeout time. The packet error rate is then given by $P_e = 1 - (1 - P_b)^C$, where P_b is the bit error probability and C the packet length in bits. For simplicity, we assume an error-free feedback channel. To ensure statistical significance, simulations have been run until at least 10 message errors were experienced, given the lowest error probability.

Considering a typical industrial application, the packet length is 120 bits for data, ACK, and polling packets, guided by the minimum packet size possible in the standard. The simulated traffic flows, with a known period, deadline and message length, belong to one of two traffic classes specified in Table I. The choice of traffic class is randomized with a uniform distribution, as is the choice of source and destination nodes for each flow, but always including the master node as one of these. Table I also shows the parameters of the retransmissions channels. Simulations have been run for 2, 4, and 8 retransmission channels, but due to space limitations only results for 8 retransmission channels are shown in the figures. For all packets the maximum number of retransmission attempts per packet ($N_{attempt}$) was fixed to two.

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 | | | |

Reliability The MER is determined through simulations, but the total reliability also includes hardware redundancy and the amount of redundancy in the frequency domain. When evaluating the MER through simulations only accepted RTCs are included. As shown in Fig. 1, the MER for the case without any retransmissions is 10^{-1} for all architectures. This is expected as the MER only depends on the Gilbert-Elliot model and the message length. For all architectures the highest improvement in MER was achieved with 8 ReRTCs (the results for 2 and 4 ReRTCs were between those for 0 and 8 ReRTCs). The highest MER improvement by retransmissions was experienced for low numbers of requested RTCs, quite independent of architecture. This can be explained by the fact that for a relatively small number of RTCs, the demand for ReRTCs is lower and therefore a higher percentage of requested retransmissions can actually be granted. The decrease in MER between one and 20 RTCs can be attributed to the fact that before all the available ReRTCs can be used, a certain number of requests for retransmissions is needed, which in turn requires a certain number of RTCs. If the ReRTCs are not used, the MER cannot be improved. The reason why the curves differ between architectures for large numbers of requested RTCs is that the multichannel architecture I could accept considerably more RTCs, while the number of retransmission channels was kept constant, leading to a lower percentage of granted retransmissions. Note also that multichannel architecture II requires an extra control packet to schedule concurrent transmissions. Each extra packet may increase the MER and make the protocol more vulnerable to channel noise. Increasing the sleeping phase to 75% of the beacon interval did not affect the MER and therefore the figures are omitted.

Considering reliability in terms of hardware and frequency redundancy, multichannel architecture I is obviously the superior solution. It has several redundant transceivers and is concurrently using all available frequencies. Considering the hardware aspect, the single-channel and multichannel architecture II are equivalent, but the use of tuneable transceivers provides higher frequency redundancy for the multichannel approach. Even if the possible traffic channel interdependencies make it hard to use the frequency domain efficiently, the ability to tune the transceiver still provides frequency diversity.

Bandwidth utilization capability Utilizing as much as possible of the available bandwidth demands a high

degree of concurrency. As mentioned above, multichannel network I has the highest degree of concurrency and can therefore easily utilize the complete bandwidth offered by each available frequency. Obviously, single-channel networks do not have the possibility of concurrency, but may utilize all of the available bandwidth on the one available frequency. Multichannel network II can be found in between these two cases. It utilizes one of the frequencies well, but whether the remaining frequencies are used efficiently depends on traffic channel interdependencies. In our analysis, we considered the network utilization for the three architectures both with and without retransmissions. The graphs shown in Fig. 2 show the utilization of the active part of the beacon interval and include only the utilization by ordinary traffic flows in order to identify the utilization penalty introduced by the retransmission channels. The calculation of utilization also includes the acknowledgement and polling packets required to transmit the data packets. Considering Fig. 2 it becomes obvious that the multiple transceiver case is superior to the other two cases in terms of bandwidth utilization capability. Without retransmissions, multichannel architecture I saturates at a utilization of about 3.9 using four concurrent transceivers. The other two networks saturate at a utilization just below 1, where multichannel case II falls even below the single-channel case. The reason for this is the penalty introduced by the tuning delay, leading to a lower number of RTCs that can be scheduled. Furthermore, the existence of the additional control packet in the MAC protocol for multichannel case II leads to an earlier saturation of the network as compared to the single-channel or the other multichannel case. When introducing retransmissions into the system, the utilization by ordinary traffic channels drops for all architectures. Multichannel network I drops to around 2.5, while the other two cases drop to about 0.6, the tuneable case still being lower than the single-channel one. The main reason for this considerable reduction is not the bandwidth utilization by the retransmission channels, which is rather small compared to the magnitude of the penalty. Instead all traffic flows become more difficult to schedule due to the shortening of the deadline (from D_i to $D_{ord,i}$) to cater for retransmissions. The same penalty was also visible for 2 and 4 retransmission channels (not shown in this paper) with hardly any difference between the 2, 4 and 8 ReRTCs cases. When considering a longer sleep phase, the utilization for the multiple transceiver case I with retransmissions fell to around 2.25, while the two other cases decreased to about 0.5. The figures are omitted due to space limitations.

Although these results look very favourable towards the multiple transceiver case, one has to take into consideration certain practical constraints. We assume that multiple transceivers located on one node are placed far enough apart to function properly without interfering with each other. However, on small nodes this might not always be possible, in which case sending and receiving at the same time is problematic. This implies that traffic on the multiple transceiver architecture case I needs to be scheduled such that either all transceivers on a slave

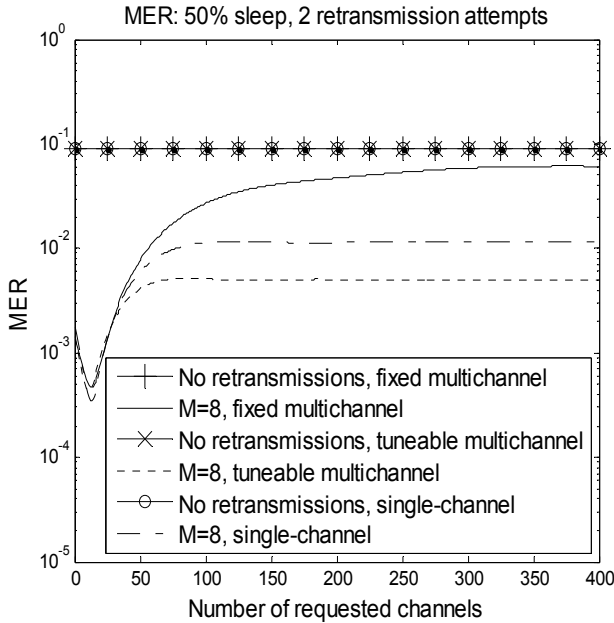


Figure 1. Message error rate

receive concurrently or all transmit concurrently. This introduces similar traffic dependencies as present in the tuneable architecture. From the scheduling point of view this does not pose a problem, however, the analysis would suffer from the same pessimism as in the tuneable case, decreasing the utilization by guaranteed real-time traffic to merely one. Bandwidth not used by guaranteed real-time traffic can be used by soft real-time and best-effort traffic. Introducing this kind of traffic into the network would certainly increase the utilization, especially for the tuneable transceiver case, where the pessimism in the analysis is the greatest and therefore the unused bandwidth the highest.

Delay The worst case delay is highest in the tuneable network due to the introduction of the tuning delay and the additional control packet, making the timeout value longer. However, when studying the average delay, our simulations have confirmed that it is independent of the architecture, as it is more dependent on the maximum number of retransmissions per packet and the number of retransmissions necessary for a successful transmission, i.e., it is connected to the MER. Both for the single-channel network and multichannel network I, traffic interdependencies are irrelevant, but, when introducing tuning, two packets can only be scheduled for concurrent transmissions if they pertain to mutually independent traffic channels, i.e., if neither source nor destination contain the same slave. The traffic flow interdependencies render the worst case delay for multichannel network II identical to that of a single-channel network.

Scalability The amount of hardware necessary for multichannel case I implies that it scales poorly, as for each frequency, one additional transceiver is necessary at each node. The tuneable case has no such issues with scalability and the more nodes there are in the network, the less likely they might be to block each other. So, even if the guaranteed utilization is not increased, the utilization by nonguarantee-seeking traffic is likely to

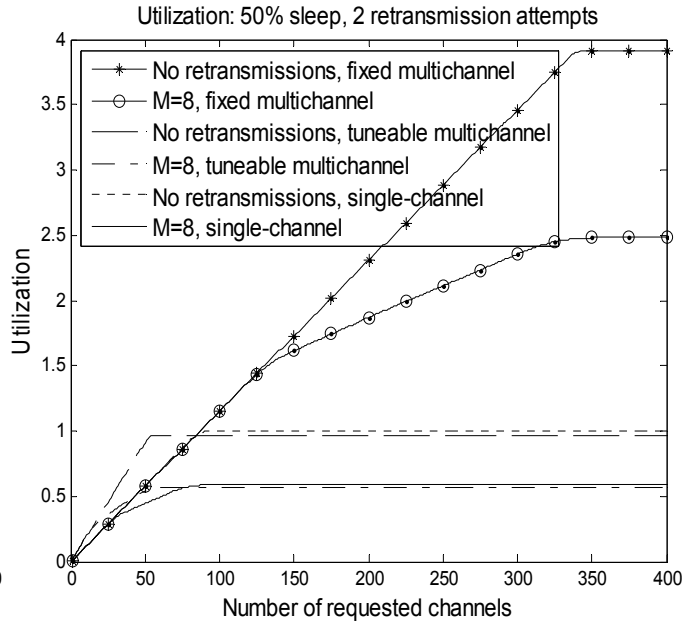


Figure 2. Bandwidth utilization

increase. The single-channel case has no such issues. The scalability is also connected to the utilization through the number of accepted RTCs, in which case the multiple transceiver architecture case I is superior.

Hardware and Software Complexity Obviously, the multichannel architecture I suffers the hardest from hardware costs, while the tuneable and single-channel case are equally good solutions from that perspective. The single-channel architecture has the channel allocation algorithm with the lowest complexity. Multiple transceivers at each node can be scheduled in parallel without the need to consider which packet to schedule over which frequency, so this implementation is also quite simple. The most complex case occurs when traffic interdependencies can interfere with pure EDF scheduling and therefore the tuneable architecture results in the most complex channel allocation algorithm.

Energy usage Although the detailed analysis of energy consumption is out of the scope of this paper, we argue that a higher number of transceivers, as in multichannel case I, leads to a higher energy usage, while a single fixed transceiver is the best solution from an energy perspective. The tuning of transceivers implies additional energy consumption in multichannel network II.

Summary We can conclude that none of the proposed architectures is best for all considered performance metrics. An appropriate architecture should thus be selected based on the data traffic model, the constraints on hardware, software and energy, as well as the required utilization, throughput and reliability. While multichannel I is superior in bandwidth utilization capability and reliability, it suffers with respect to hardware costs, scalability and energy usage. On the other hand, these are areas in which the tuneable architecture implementation has its advantages. Hardware costs and energy usage are kept low and scalability is not a problem. However, the low bandwidth utilization capability is the result of low concurrency at each node, since frequency diversity only exists on a network wide level, not a per-node level.

8. Conclusion

We presented and evaluated two multichannel network architectures based on IEEE 802.15.4 for industrial networks. Using a single-channel network as a baseline, we added diversity in the time and frequency domains to increase reliability and guaranteed throughput in industrial networks while keeping the advantageous energy saving properties of the underlying IEEE 802.15.4 standard. The two described network architectures are using either F fixed transceivers on all nodes or tuneable slave transceivers. Through EDF polling and real-time analysis admission control, we added predictability to the network, needed to support hard real-time traffic in industrial applications. The architectures were evaluated in terms of reliability, utilization, delay, complexity, scalability and energy efficiency. The results show that neither of the described multichannel alternatives ranks high for all evaluated parameters. An appropriate architecture should thus be selected based on the industrial application in question. The option with several slave transceivers provides a high level of service through concurrency in the frequency domain and hardware redundancy, whereas the option with tuneable slave transceivers may increase guaranteed throughput for certain data traffic models while having low hardware costs, good scalability and low energy usage.

References

- [1] IEEE 802.15.4 Standard (2006) Part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Standard for Information Techn., IEEE-SA Standards Board, 2006.
- [2] <http://www.hartcomm.org/>
- [3] <http://www.isa.org/>
- [4] M. Jonsson and K. Kunert, "Towards Reliable Wireless Industrial Communication with Real-Time Guarantees," *IEEE Transactions on Industrial Informatics*, vol. 5, no. 4, pp. 429-442, Nov. 2009.
- [5] K. Kunert, E. Uhlemann, and M. Jonsson, "Predictable real-time communications with improved reliability for IEEE 802.15.4 based industrial networks," in Proc. *8th IEEE International Workshop on Factory Communication Systems (WFCS '10)*, Nancy, France, May 2010.
- [6] S. Pejhan, M. Schwartz, and D. Anastassiou, "Error control using retransmission schemes in multicast transport protocols for real-time media," *IEEE/ACM Transactions on Networking*, vol. 4, no. 3, pp. 413-427, 1996.
- [7] 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," in Proc. *Workshop on Real-Time Networks (RTN '06)*, Dresden, Germany, July 2006.
- [8] F. Shu, "Performance evaluation of the IEEE 802.15.4 CSMA-CA protocol with QoS differentiation," in Proc. *International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP '08)*, Sydney, Australia, Dec. 2008, pp. 475-480.
- [9] J. Francomme, G. Mercier, and T. Val, "A simple method for guaranteed deadline of periodic messages in 802.15.4 cluster cells for control automation applications," in Proc. *IEEE Conference on Emerging Technologies and Factory Automation (ETFA '06)*, Prague, Czech Republic, Sept. 2006, pp. 270-277.
- [10] J. Mo, H.-S.W. So, and J. Walrand, "Comparison of Multichannel MAC Protocols," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, Jan 2008, pp. 50-65.
- [11] M. Wang, L. Ci, P. Zhan, and Y. Xu, "Multi-channel MAC protocols in wireless ad hoc and sensor networks," in Proc. *ISECS International Colloquium on Computing, Communication, Control, and Management (CCCM '08)*, Guangzhou, China, Aug. 2008, pp. 562-566.
- [12] S.-C. Lo, "Design of multichannel MAC protocols for wireless ad hoc networks," *International Journal of Network Management*, vol. 19, no. 5, Sept. 2009, pp. 399-413.
- [13] B. Andersson, N. Pereira, and E. Tovar "Preliminary Discussion on Globally Prioritized Medium Access for Multi-Channel Wireless Systems", in Proc. *6th International Workshop on Real-Time Networks (RTN '07)*, Pisa, Italy, July 2007.
- [14] A. Rangnekar and K.M. Sivalingam, "QoS Aware Multi-Channel Scheduling for IEEE 802.15.3 Networks", *Mobile Networks and Applications*, vol. 11, no. 1, pp. 47-62, Feb. 2006.
- [15] P.G. Namboothiri and K.M. Sivalingam, "Performance of a multi-channel MAC protocol based on IEEE 802.15.4 Radio", *IEEE 34th Conference on Local Computer Networks (LCN '09)*, Zurich, Switzerland, Oct. 2009.
- [16] H.K. Le, D. Henriksson, and T. Abdelzaher, "A Practical Multi-Channel Media Access Control Protocol for Wireless Sensor Networks", *International Conference on Information Processing in Sensor Networks (IPSN '08)*, St. Louis, MO, USA, Apr. 2008, pp. 70-81.
- [17] Chipcon, *2.4 GHz IEEE 802.15.4/ZigBee-ready RF Transceiver*, Chipcon Products Texas Instruments 2004.
- [18] C.L. Liu and J.W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of the ACM*, vol. 20, no. 1, pp. 46-61, Jan. 1973.
- [19] H. Hoang and M. Jonsson, "Switched real-time Ethernet in industrial applications - deadline partitioning," in Proc. *The 9th Asia-Pacific Conference on Communications (APCC '03)*, Penang, Malaysia, Sept. 2003, vol. 1, pp. 76-81.
- [20] M. Spuri, "Analysis of deadline scheduled real-time systems," *Techn. Rep. No. 2772, INRIA*, France, 1996.
- [21] S.K. Baruah, A.K. Mok, and L.E. Rosier, "Preemptively scheduling hard-real-time sporadic tasks on one processor," in Proc. *11th Real-Time Systems Symposium (RTSS '90)*, Lake Buena Vista, FL, USA, Dec. 1990, pp. 182-190.
- [22] 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 Systems*, vol. 2, no. 4, pp. 301-324, Nov. 1990.
- [23] 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.
- [24] T. Voigt, F. Österlind, and A. Dunkels, "Improving Sensor Network Robustness with Multi-Channel Convergecast", in Proc. *2nd ERCIM Workshop on e-Mobility*, Tampere, Finland, May 2008.