

A Novel Medium Access Control Protocol for Real-Time Wireless Communications in Industrial Automation

Klaus Tittelbach-Helmrich, Najib Odhah, Zoran Stamenković

IHP, Im Technologiepark 25, D-15236 Frankfurt (Oder), Germany

Email: tittelbach@ihp-microelectronics.com, odhah@ihp-microelectronics.com, stamenkovic@ihp-microelectronics.com

How to cite this paper: Tittelbach-Helmrich, K., Odhah, N. and Stamenković, Z. (2017) A Novel Medium Access Control Protocol for Real-Time Wireless Communications in Industrial Automation. *Int. J. Communications, Network and System Sciences*, 10, 282-298.

<https://doi.org/10.4236/ijcns.2017.1011017>

Received: October 30, 2017

Accepted: November 26, 2017

Published: November 29, 2017

Copyright © 2017 by authors and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In this paper, a novel Medium Access Control (MAC) protocol for industrial Wireless Local Area Networks (WLANs) is proposed and studied. The main challenge in industry automation systems is the ultra-low network latency with a target upper bound in the order of 1 ms while maintaining high network reliability and availability. The novelty of the proposed wireless MAC protocol resides in its similar latency performance as its counterpart in wired industrial LAN. First, the functional design of the MAC protocol is introduced. Then its performance results gained from hardware implementation (SystemC and VHDL) on an FPGA platform are presented. Finally, a real-time communication module which achieves the ultra-low latency required in industrial automation is described.

Keywords

MAC, WLAN, Latency, Reliability, Real-Time Communications, Industry Automation, Hardware Implementation, SystemC, VHDL

1. Introduction

The recent advances in wireless networking have witnessed the emergence of new wireless applications in different areas of the society. One of the most important wireless applications is wireless industry automation which incorporates many wireless technologies to cope with the urgent requirements of the real-time control communications. The two most significant requirements of the wireless industrial automation are the ultra-low network latency of 1 ms or less and the ultra-high reliability of millisecond outage per day [1]. These requirements have been already met by wired industrial Local Area Networks (LAN). Examples of

established wired industrial LAN systems are the Industrial Ethernet protocols such as *EtherCAT* and *ProfinetIRT+* [2] [3] [4] [5] [6]. Introducing innovations in the higher protocol layers, the bus cycle-time has been reduced from 10 ms to 125 μ s. Wires are expensive to install and maintain and the possibility of replacing them with reconfigurable wireless links is a tempting opportunity for industry. Achieving this replacement requires that the wireless connection works with a similar latency, reliability and capacity as the wired counterpart and that its management is simplified.

Low latencies are provided by a number of enhanced wireless systems. For example, *Wireless Hart* guarantees a latency of 10 ms [7]. It is based on the physical layer of IEEE 802.15.4. Mayank *et al.* [8] report on a very innovative protocol for highly reliable wireless real-time traffic that achieves a latency of ca. 1 ms. It is also built on an existing PHY layer. Both protocols are based on the “CSMA/CA” (carrier-sense multiple access with collision avoidance) principle, which does not really allow fast channel access (*i.e.* low latency) due to the “Listen before Talk” requirement. Another novel “TDMA”-based (time division multiple access) MAC protocol that does not suffer from this restriction has been proposed in [9]. Also, systems based on 5 G networks have been proposed [10].

In this paper, a high-reliability and ultra-low-latency wireless MAC protocol for industry automation applications is proposed and studied. Our solution is a wireless extension of the wired Field Bus systems used in the industry automation to support easy device installation and mobility. It is named DEAL MAC and is the core part of a large collaborative research project called *DEAL*, which is a German acronym standing for “Drahtlose zuverlässige Echtzeitkommunikation für Automatisierung, Produktion und Logistik in der Industrie” (in English: “Wireless Reliable Real-time Communications for Automation, Production and Logistics in Industry”) [11] [12].

The rest of the paper will be organized as follows: Section 2 gives an overview over the system requirements and architecture. In Section 3, the functionality of the proposed MAC protocol is described, and simulation data for performance (throughput, round-trip time) are presented. Section 4 covers MAC implementation as well as design and implementation of the complete DEAL communication module including its physical (PHY) layer. Finally, the conclusions are drawn in Section 5.

2. System Requirements and Architecture

This section presents an analysis of the system requirements and, based on that, the definition of the basic system architecture, in particular the MAC protocol. We have adopted existing wireless standards and incorporated changes required by our concept.

2.1. System Requirements

Wireless and mobile communication systems are becoming increasingly important

for industrial automation, in which the most important system parameters to be fully optimized are the network latency and reliability. The latency should be minimized to a value below 1 ms and the reliability should be improved to maintain a packet error rate in the order of 10^{-9} . Also, a data rate equal to or greater than 20 Mbit/s must be satisfied. The system latency can be reduced by fast data processing technologies (hardware versus software), reducing overheads (like frame preambles), flexible scalable packet lengths, clever acknowledgement policies, and other advanced MAC protocol techniques. Potential industrial applications of such a system include switch-boxes (for example, interconnection with sensors), coupled-motor systems for automated production and logistics, and production robots. Moreover, for driverless industrial trucks and crash test vehicles, in addition to the real-time wireless communication, the real-time wireless positioning with high accuracy in both the outdoor and indoor scenarios is required. Therefore, innovative baseband and MAC protocol solutions have to be explored and designed to boost the industrial applications of the wireless communication technology. As there are already baseband chips on the market, which allow latencies well below 0.5 ms, the real bottlenecks are the traditional MAC protocols and data packet structures, which can raise the latency above 30 ms.

A MAC protocol is part of most wireless and some wired communication systems. It controls the access to the shared medium, *i.e.* it defines rules, methods and priorities to send data and management information. Normally, also frame formats, acknowledgement policies, security features and similar properties are defined. Many MAC protocols have been standardized in order to ensure interoperability of equipment from different manufacturers. Prominent examples are the standards IEEE 802.11 for wireless LAN [13] and IEEE 802.15 for wireless Personal Area Networks (PAN) [14]. Wireless communication networks for industry automation may be based on similar operation principles as conventional WLAN. However, there are also important differences in the requirements as shown in **Table 1**. For wireless industry automation, a novel proprietary MAC protocol must be developed because the established standards such as IEEE 802.11 and IEEE 802.15.4 cannot meet the latency (round trip delay) requirements and suffer from very large overhead for small packet sizes [15] [16] [17].

Reliable real-time wireless communication for industry automation, *e.g.* in order to simplify device installation and allow mobile nodes to be attached, is still an open issue. Many research activities have recently been conducted to tackle this challenge in the framework of the German so-called “Industrie 4.0” program, where the *DEAL* project is one part. The real challenge of the wireless automation is the traffic structure: packet size is quite low (few bytes per node in contrast to standard WLAN, where it is in the range up to 1500 bytes), packet loss rate shall be in the order of 10^{-9} (while re-transmissions are almost forbidden for latency reasons), and traffic is very regular (same number of bytes in each cycle, which has a length in the order of 1 ms).

Table 1. Basic network parameters of industrial and standard WLAN.

Property	Industrial Network	Standard WLAN
Network topology	Fixed	Flexible and varying in time
Traffic characteristics	Generally quite regular	Generally bursty
Typical packet size	1 - 32 byte	Up to 1500 byte
Packet latency	Below 1 ms	Generally above 10 ms
Packet error rate	Typically below 10^{-9}	Typically from 10^{-2} to 10^{-4}
ARQ = Automatic Repeat Request	ARQ hardly possible	Reliability improved by ARQ

2.2. System Architecture

Several wired communication standards have been established for industrial automation systems [2] [3] [4] [5] [6]. The most advanced ones are based on the Ethernet physical layer, but the MAC protocols are unique and specific for each standard. The goal of the *DEAL* project is to demonstrate a wireless extension of such a conventional Field Bus system (e.g. *EtherCAT* or *ProfiNet*). It should easily integrate into existing systems, connect wirelessly access points and mobile clients, and provide basically the same performance (with respect to data rate, reliability, and latency). The extension should also include a localization feature. A typical application scenario is sketched in **Figure 1**. The wired Field Bus system consists of a “Field Bus Controller”, where a number of sensor or actor devices are directly attached. Its wireless extension part is attached to the Field Bus via a few base stations (access points). They control the wireless network that may connect several additional sensors and actors wirelessly. The interface between a *DEAL* radio module (actually, its MAC layer) and a Field Bus device or controller is based on SPI. A commercial field bus converter converts this to one of the common Field Busses like *EtherCAT* or *ProfinetIRT+*. The *DEAL* system also contains a localization system, based on a time difference of arrival (TDoA) method [18]. Each node can determine its own position, based on ranging measurements to several fixed base stations. This is sketched in **Figure 1**, but not further discussed in this paper.

The physical (PHY) layer of the *DEAL* wireless extension network is the one of IEEE 802.11a (5-GHz OFDM Wi-Fi) [13]. It provides the required data rates (a few tens of Mbit/s) and is proven to be robust in multi-path environments. One drawback is the relatively large packet overhead due to preamble and signal field (20 μ s) when the data packets are small like in Field Bus systems (typically 4 - 16 bytes, which corresponds to around 3 - 12 μ s at 12 Mbit/s). However, this is a common bottleneck in conventional packet-based wireless systems.

The *DEAL* MAC has been developed from the scratch and tailored to the demands in industry automation. To meet the extremely challenging latency and jitter requirements, the complete MAC layer has been implemented in hardware (on an FPGA). This comprises sending and receiving data packets with low latency, frame timing within a super-frame, time synchronization between

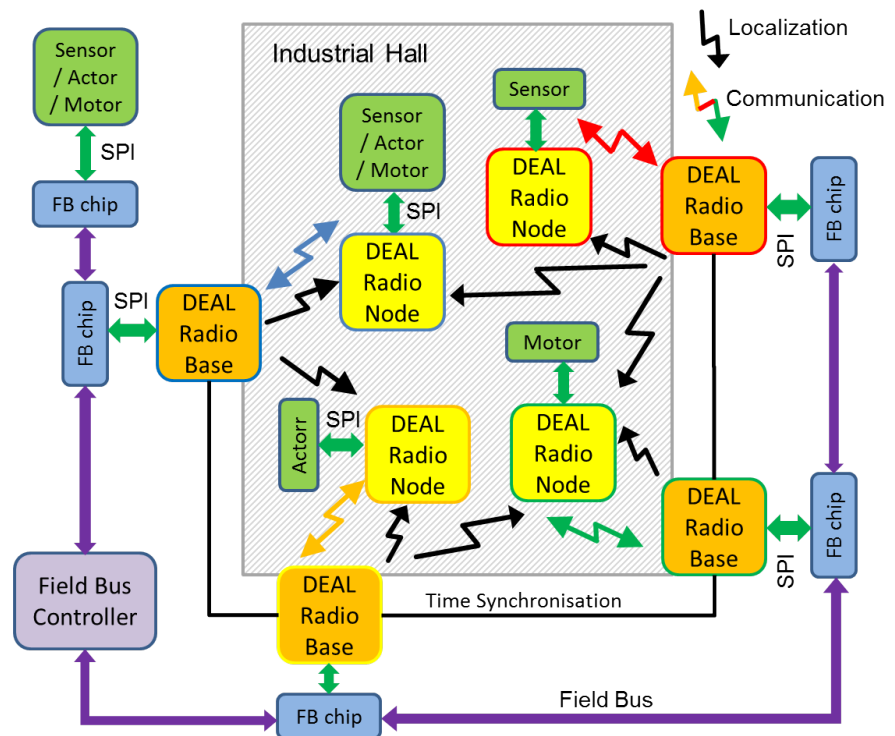


Figure 1. DEAL system architecture.

stations, and beamforming. For system setup and configuration, there is a software tool that runs externally, e.g. in the Field Bus controller.

3. MAC Layer

Because of the hard MAC latency requirements and the regular traffic structures we have decided to implement the MAC layer fully in hardware. In this case, we do not have to expect performance bottlenecks as in software solutions. On the other hand, the flexibility and extendibility of the hardware implementation will be restricted [19] [20].

The MAC processor core has been primarily designed and simulated in *SystemC*. This is a hardware-oriented extension of the well-known C programming language. The MAC behaviour and performance can be well simulated in *SystemC*. Then, the commercial software tool “Stratus” (from Cadence, formerly “CtoS”) was used to generate synthesizable RTL-level Verilog code from *SystemC*. The Verilog code was supplemented by a few components (e.g. interface blocks) directly written in VHDL. Finally, the whole MAC design was synthesized and implemented for a Xilinx Artix7 FPGA platform.

3.1. MAC Functionality

The MAC network architecture consists of a central controller (access point/base station) and several stations (nodes). Data transfer is performed between controller and nodes only (as opposite to peer-to-peer communication). The station (node) provides the following functionality:

- Frame transmission and reception (cyclic and acyclic data) with $\approx 2 \mu\text{s}$ precision in time,
- Error checking (CRC on top of the Viterbi decoder in the PHY layer),
- Frame re-transmission if required (missing acknowledgement), duplicate rejection,
- Management of tables for beam forming parameters (including reception and forwarding of localization information in frames from nodes to base stations),
- Evaluation of time synchronization signals and adjustment of the time base.

The base station (access point) provides the following additional functionality:

- Maintenance and administration of the MAC super-frame structure,
- Transmission of time synchronization signals to synchronize all nodes in the network (precision $\approx 2 \mu\text{s}$),
- Association/disassociation of nodes at the access point, optionally remote configuration of nodes from the Field Bus controller.

Figure 2 shows a simplified but typical super-frame structure. The base station sends a beacon, broadcasting some general system parameters like network ID and system time. To save preamble overhead, it directly concatenates to all downlink data packets (downlink data fragments). The nodes send their uplink data in individual data frames, each needing its own preamble and other frame overhead like inter-frame spacing. The super-frame period must be long enough to cover all frames that must be transferred to serve all sensors/actors/motors in the network. In a real radio network, one has to expect frame errors, where re-transmissions must be carried out to meet the reliability requirements. In real Field bus systems, also so-called “acyclic data” are transferred (normally much less than once per Field Bus cycle), e.g. for configuration purposes. For this, we have reserved some time at the end of the super-frame period in the DEAL MAC as shown in **Figure 3**. This time period is called On-Demand Period (ODP) and may also be used for acknowledgements (ACKs) for uplink data.

The interface between Field Bus and MAC processor core is a 16-bit Dual-Port Memory (DPM) interface as usual in Field Bus systems. Our MAC processor incorporates an SPI slave, that is connected to a converter chip (see **Figure 1**), which converts the data to the format used in the Field Bus network. Such chips are commercially available for different Ethernet-based flavours of Field Bus. Logically, the MAC-Field Bus interface provides a number of addressable registers (or memory cells) that can be read and written by the SPI and MAC processor core to interchange control data (parameters) and cyclic or acyclic Field Bus payload data (wireless frames).

3.2. MAC Performance

In this section, we simulate the performance of our novel MAC protocol implementation in order to study its latency (round-trip time, minimum super-frame period) as a function of the network size (number of the wireless nodes), the

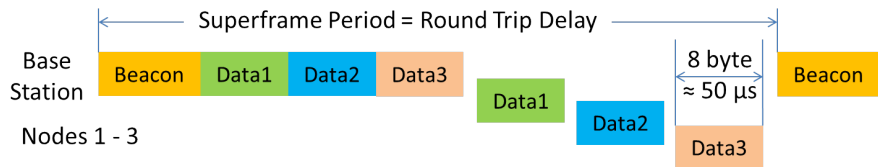


Figure 2. Basic DEAL MAC super-frame structure with typical timing data.

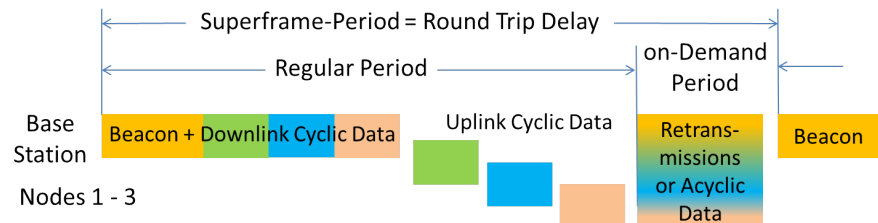


Figure 3. DEAL MAC super-frame structure with On-Demand Period (ODP).

data packet length and the data rate of the underlying PHY layer. Firstly, error-free transmission is assumed, and then the influence of packet errors is roughly estimated.

We calculate the (minimum) length of the super-frame period, which is required to cover all frames needed to transfer a certain number of payload bytes downlink and/or uplink to/from each node in one Field Bus cycle, which is presumed to be equal to the super-frame period. We vary the number of nodes, payload sizes, PHY data rate, etc.

For each PHY frame (IEEE 802.11a compliant), the following parameters are taken into account:

- 16 μ s Preamble of the Packet.
- 4 μ s (= 1 OFDM symbol) Signal field.
- $n \times 4 \mu$ s (= n OFDM symbols) PHY Payload (with variable data rate)
- 8 μ s (= 2 OFDM symbols) nominal spacing between two packets

Within the PHY payload, the following overhead is taken into account:

- 2 bytes for scrambler initialization and Viterbi tail bits.
- 3 bytes for fragment 0 (protocol version, MAC source address, emergency stop)
- 5 bytes fragment header + CRC for each fragment in the packet.

The packet length in bytes is converted to packet duration depending on the data rate. It is taken into account that the packet duration is always an integer multiple of the OFDM symbol duration of 4 μ s.

From this, the packet durations are calculated and aggregated for three data rates 6, 12 and 24 Mbit/s. This results in the super-frame period, which is at least required to transfer all the data.

3.2.1. Data Transfer without ACKs and Retransmissions

First, the performance is calculated assuming that all packets can be delivered without errors. Therefore, no (time) reserves are necessary for re-transmissions.

Figure 4 shows the minimum required super-frame period for a network in which four bytes are to be transmitted in the downlink *and* uplink for each node in each cycle. For this constellation, approximately 100 stations can be supplied with a round-trip delay of ca. 5 ms at a data rate of 12 Mbit/s. The latency increases linearly with the number of nodes in the network, *i.e.* a network consisting for example of 10 nodes could be served within 0.5 ms super-frame period (or round-trip delay).

Figure 5 shows the situation if, for each node, 4 bytes are to be transmitted *either* in the downlink *or* the uplink only. In the downlink, the data is directly “attached” to the beacon without a large overhead, whereas a preamble, a signal field and an inter-frame spacing are required for each packet in the uplink. The latter is much less performant, *i.e.* the minimum super-frame duration is significantly greater. Since only one packet with a relatively large PHY payload and relatively small overhead (preamble, etc.) is sent in the downlink, the influence of the data rate is significantly stronger. The duration of short packets, on the other hand, is dominated by the overhead and the data rate has relatively little influence.

Finally, **Figure 6** shows curves for different packet sizes at a fixed data rate of 12 Mbit/s. In the uplink, significantly more time is required than in the downlink since every data frame needs its own preamble. In the downlink the super-frame period increases strongly with the size of payload data since the overhead is small (only one preamble) and the frame duration depends largely on payload size and data rate. This is much less pronounced in uplink, where the timing is dominated by the overhead (mostly the preamble), which does not scale with payload size. The curves for 4 and 8 bytes even coincide because the same number of OFDM symbols is in both cases. Including header overhead, both PHY payload sizes of 4 or 8 bytes fit into 3 OFDM symbols of 6 bytes each at 12 Mbit/s.

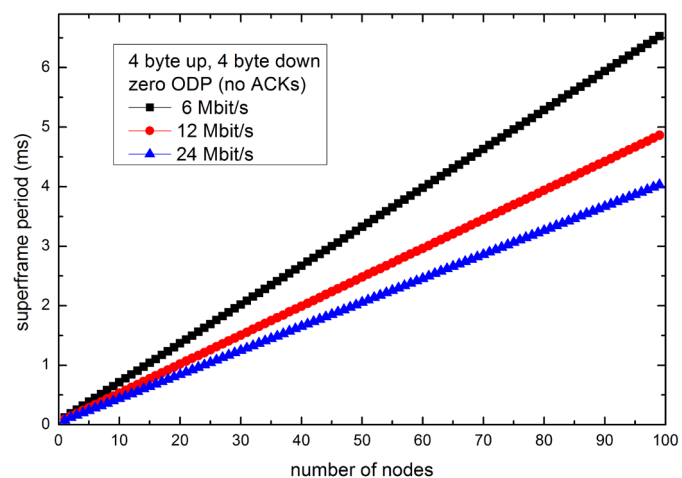


Figure 4. Minimum super-frame duration for 4 bytes in downlink and uplink each.

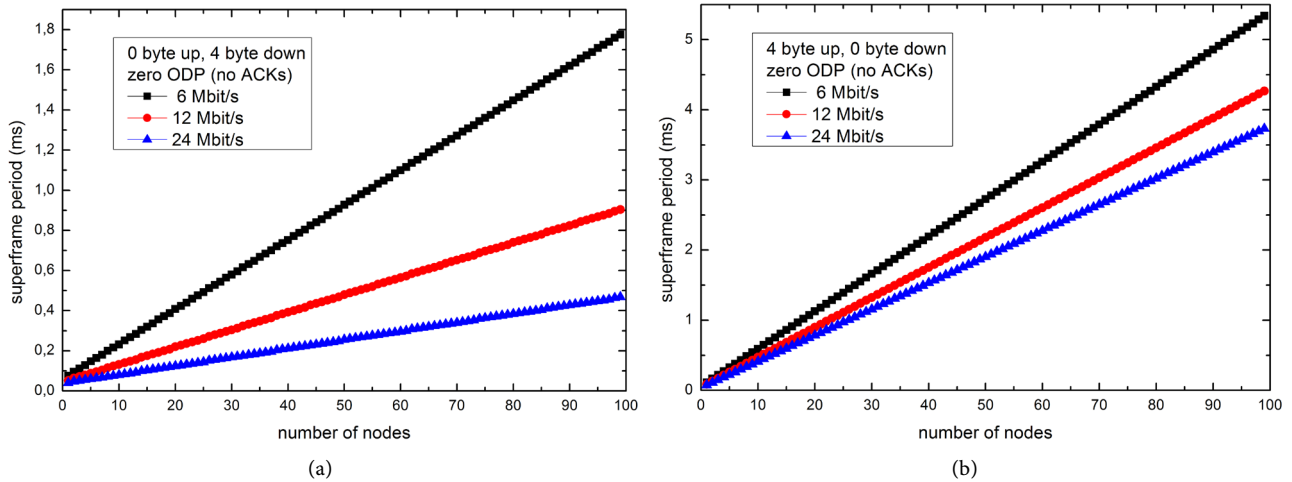


Figure 5. Minimum super-frame duration for 4 bytes either in downlink (a) or in uplink (b).

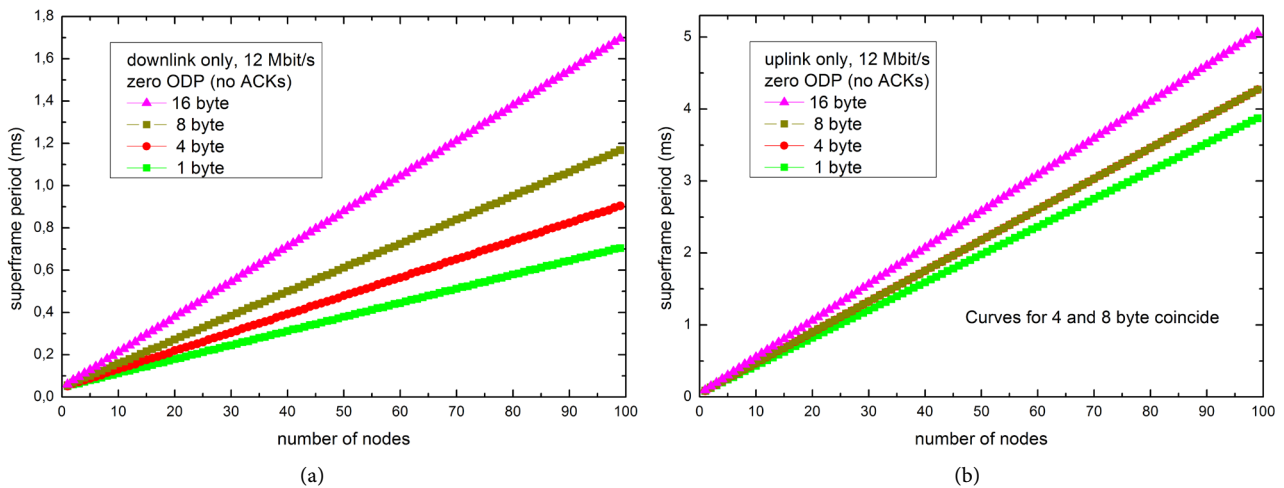


Figure 6. Minimum super-frame duration for variable data packet size either in downlink (a) or in uplink (b).

3.2.2. Data Transfer with ACKs and Retransmissions

If an On-Demand Period (ODP) is introduced, as shown in **Figure 3** or **Figure 7**, the possibility exists to send incorrectly received packets again (re-transmission). Moreover, the ODP can be used to transfer so-called acyclic data. These may occur occasionally (e.g. for system [re-]configuration). Normally, transfer of acyclic data is not critical with respect to latency, by correct delivery must be guaranteed. This means, that acknowledgements (and possibly re-transmissions) are a must.

For the ODP, a certain time is reserved in the super-frame period, which lengthens it. To estimate the degradation of the latency, we measure its duration in units of the uplink frame duration. This should be the typical case: a complete packet including preamble etc. needs to be repeated. This takes as long in the downlink as in the uplink.

From **Figure 8(b)**, it can be seen that at least for large networks with many nodes the super-frame period increases only insignificantly due to an ODP. This

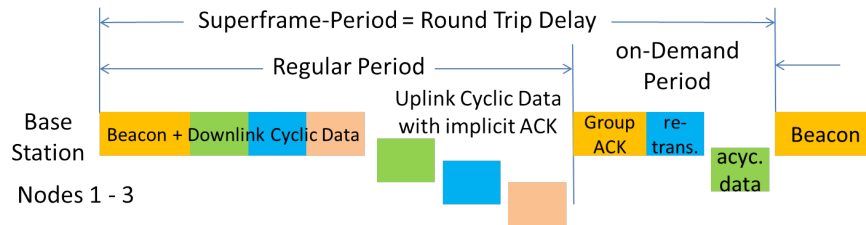


Figure 7. DEAL MAC super-frame with acknowledgement (ACK) and optional retransmission.

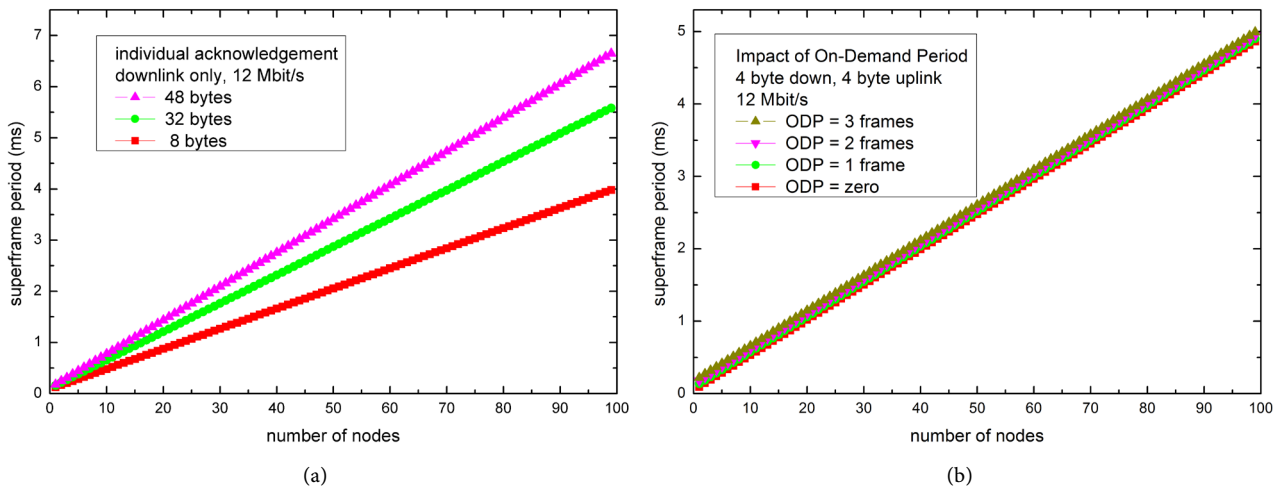


Figure 8. Minimum super-frame duration for different acknowledgement policies ((a): individual acknowledgement, (b): implicit/group acknowledgement in an ODP).

presumes that the packet error rate remains significantly smaller than the reciprocal number of stations, so that re-transmissions are rare and do not occur more than once per super-frame period.

The DEAL MAC supports different types of acknowledgements. The most inefficient one is an individual acknowledgement frame that is immediately sent by a receiver if it has correctly received a data frame. This results in an acknowledgement frame that requires the full overhead (preamble and inter-frame spacing) but transfers only one bit of information. For standard WLAN this is often the best choice, because this policy is the most flexible one for bursty traffic.

For DEAL, the impact of this policy is simulated in **Figure 8(a)**. The data shall be compared with **Figure 6**. Although there are only downlink data (as in **Figure 6(a)**), the achievable super-frame period compares with **Figure 6(b)** (uplink data). This means that from the efficiency point of view, the individual acknowledgement behaves like uplink data, but no (payload) information is transported.

If there are uplink data, the best choice is to combine acknowledgement for downlink data and uplink payload data into one frame (implicit acknowledgement). The uplink data are acknowledged by the base station in a group acknowledgement, sent in the ODP period (see **Figure 7**). There is only *one* frame required that covers one acknowledgement bit for each uplink frame.

Implicit acknowledgements come with nearly zero-penalty when they are piggy-backed to data frames of the opposite direction. Also, the impact of the OPD period that is required for the group acknowledgement and the possibly required re-transmission of a payload data frame results in a very small overhead, as seen in **Figure 8(b)**.

4. System Implementation

A complete wireless communication system should fulfil the following demanding requirements: very low round-trip delay of <0.5 ms, very low jitter of <20 μ s, and very high reliability (packet error rate less than 10^{-9}) for a network size of 2 - 100 nodes. In order to support efficient development and roll-out, we implement the system relying on existing standards (e.g. Field Bus) and innovative solutions.

4.1. MAC Layer

In this section, we briefly describe our implementation of the DEAL MAC layer. The main reason for implementing it in hardware, and not in software like in [21], is to meet the extremely challenging latency and jitter requirements for industrial automation. Also, the data traffic is quite regular and frame structure is simple. So, the administrative effort is relatively low and can be handled by hardware, which is usually less flexible than software. In a standard digital hardware design flow, one would design all components in VHDL or Verilog, write respective test benches, simulate, synthesize and layout the design and manufacture it. Usually, first an FPGA-based system would be built for extensive real-time tests. Later, an ASIC implementation could be carried out.

In our proposed DEAL MAC, we have tried a design flow that runs at a higher abstraction level and that is closer to software development principles. The language *SystemC* is an extension to the well-known C programming language. It allows partitioning the system into different modules operating in parallel (like entities/instances in VHDL). Within each module, several processes (or methods) can be declared (like parallel threads in software). We have used this approach for a pure hardware implementation. Generally, it would also be possible to carry out (and optimize) a hardware-software partitioning of the system. The main advantage of our *SystemC* approach is that operations can be performed in parallel (since it is hardware), but the syntax used for designing is basically the software programming language C. This is easy to learn for software developers. Cadence's "Stratus" compiler is used for generation of HDL code (Verilog) from *SystemC*. Some peripheral blocks like the external Field Bus (SPI) interface have been designed directly in HDL. The main reason is that they have asynchronous or pausable clocks, which cannot be handled in *SystemC*. Finally, all HDL sources are combined with a top-level VHDL module and a constraint file to generate the FPGA implementation in a standard Xilinx ISE design flow. Also, an ASIC implementation would be possible.

The SystemC-based design flow that has been exercised included the following steps (for details see also [16]):

- Design of the MAC layer with all functional details in *SystemC*,
- Verify of the correct MAC function using *SystemC* test benches and simulations,
- Convert the *SystemC* MAC model into RTL-level synthesizable Verilog code using the *Cadence's* "Stratus" tool (formerly "CtoS" = "C to Silicon"),
- Co-simulate the generated Verilog code with the *SystemC* test benches,
- Optionally, build VHDL or Verilog test benches for a pure Verilog/VHDL simulation,
- Implement the MAC processor on an FPGA using standard Xilinx design tools.

The hardware platform for the MAC implementation is a board designed around a Xilinx Artix7 XC7A200T FPGA. Apart from the FPGA chip (which is jointly used for MAC and digital baseband), the board provides power supply, the SPI interface to attach the Field Bus converter chip for FieldBus connections, the interface to the RF frontend board, an LCD display and a number of general purpose pins that can be used for debugging. More details are given at the end of chapter 4.2. A photo is shown in **Figure 11**.

4.2. Digital Baseband Processor

The DEAL physical layer complies with the IEEE 802.11a standard [13] (5-GHz OFDM WiFi). It provides data rates of 6, 12, and 24 Mbit/s and is proven to be robust in multi-path environments. The PHY frame structure (16 μ s preamble, 4 μ s signal field, variable length PHY payload) exactly complies with the standard. The structure inside the PHY payload, however, is completely different. We also reduce the nominal inter-frame spacing in order to improve the latency (reduce the overhead) for the DEAL packets, which are typically quite small (around 10 - 50 bytes).

The block diagram of the digital baseband processor is shown in **Figure 9**. Major building blocks are (see also [22]):

- Scrambler/Descrambler: data randomizer to suppress long constant bit sequences
- Convolutional coder/Viterbi decoder: forward error correction (conv. coder, rate 1/2) for correcting errors in received data
- (De-) Mapper + (De-) Interleaver: spreading of data bits to individual OFDM sub-carriers
- FFT (this block is shared between TX und RX): fast Fourier-Transformation frequency \leftrightarrow time domain
- Synchronizer: synchronization of received frames with respect to time and frequency

The baseband processor has been designed in the traditional way using the VHDL hardware description language. The main clock frequency is 20 MHz,

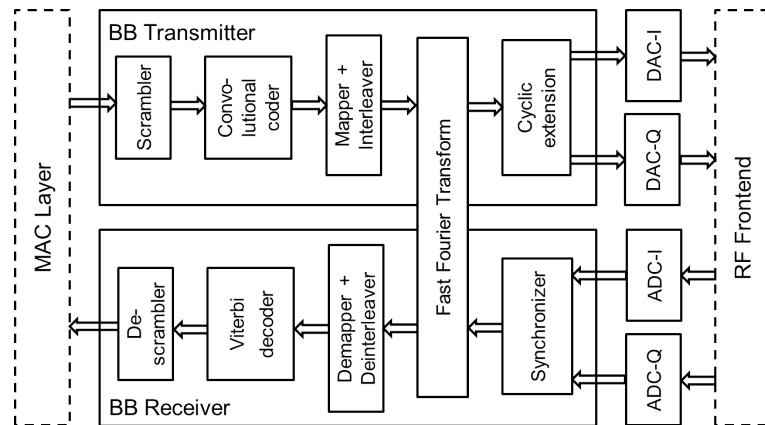


Figure 9. Block diagram of the DEAL OFDM baseband processor (IEEE 802.11a compliant).

which corresponds to the sample rate of the baseband signal delivered to the DAC converters and received from the ADCs, respectively. In order to simplify low-pass filtering of these signals, however, the sample rate of the converters is doubled to 40 Msample/s. The converters are integrated into the RF frontend chip. The DAC/ADC data bus width is 12 bits.

For DEAL, we have made an FPGA implementation on a Xilinx Artix7 XC7A200T FPGA. This device is large enough to accommodate the MAC together with the baseband processor. In total, the following FPGA resources are needed:

Number of Slice Registers	30,663	(out of 269,200)
Number of Slice LUTs	41,426	(out of 134,600)
Number of RAMB36E1s	7	(out of 365)
Number of RAMB18E1s	20	(out of 730)
Number of DSP48E1s	59	(out of 740)
Number of BUFG/BUFGCTRLs	9	(out of 32)
Number of bonded IOBs	123	(out of 285)

4.3. Radio Frontend

The RF radio frontend in our current first DEAL module implementation is realized using the *RF Agile Transceiver chip* AD-9361 by *Analog Devices* plus an external power amplifier. The AD-9361 is a very complex, flexible and powerful radio chip well suited for our purpose. In the second stage we will use a tailored RF frontend chip developed at the DEAL partner TU Dresden that additionally will support beamforming and beamsteering on patch antennas [18].

Figure 10 shows the block diagram of this DEAL RF front-end. It is a zero-IF architecture with modifications for transmitting and receiving FMCW chirps as needed for localization. The requirements for transmitter linearity are set by the needs of the OFDM communication system since the localization is purely frequency modulated where linearity is not important issue. Furthermore, the localization system does not need a base band, because it uses the Phase-locked

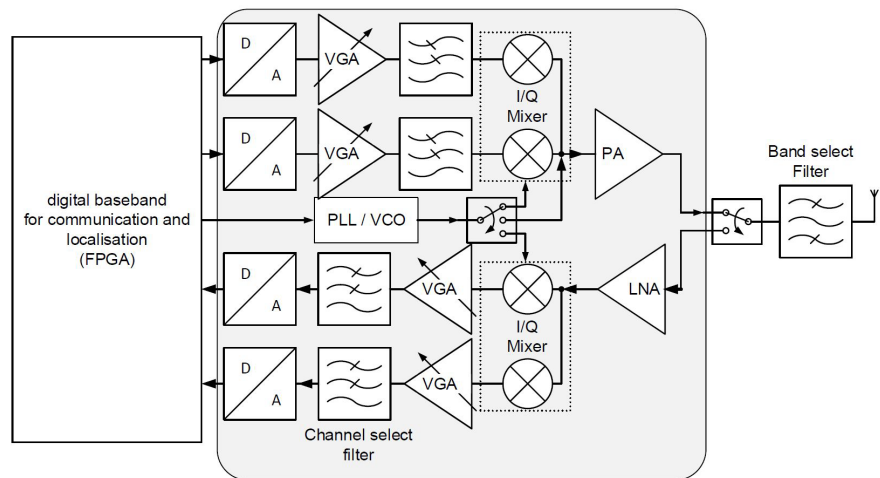


Figure 10. Block diagram of the DEAL Radio Frontend (by courtesy of B. Lindner).

Loop (PLL) to directly modulate the oscillator and generate the chirp. In the transmit mode, the output of the PLL can be switched directly to the input of the Power Amplifier (PA) for localization or to the mixer for communication. When receiving a chirp, it is mixed down with another chirp from the PLL. Then, as for communication, the resulting base band signal will be filtered with a configurable low pass filter and amplified by a Variable Gain Amplifier (VGA).

4.4. Communication Module

A photo of the current DEAL communication module is shown in **Figure 11**. It consists of two boards: the lower one for the AD9361 RF frontend, the upper one containing the Xilinx FPGA for the MAC and baseband processors and the power supply. The external interface is an SPI that may be attached to common Ethernet-based Field Bus systems using commercial off-the-shelf Field Bus converter chips (see **Figure 1**). The SPI cable can be seen in the photo's upper right corner. We are using two separate antennas for transmit and receive. The base size of the module is about $8 \times 9 \text{ cm}^2$.

As first experimental setup to try the whole communication system, we have placed two DEAL modules (base station and one node) in a lab environment. The Field Bus network and controller are replaced by an SPI connection to a standard PC (via SPI \leftrightarrow USB converter) and a terminal program to control the setup. Payload data are provisionally generated by a small hardware block inside the FPGA. A logic analyser was used to visualize and verify the functionality of the system.

Figure 12 shows a snapshot from the Logic Analyser. We see the quasi-analogue ADC/DAC signals (I and Q components) of two MAC frames. The left one is a beacon, concatenated with downlink data, the other one an uplink data frame. At the right border, the next beacon starts. The white waves belong to the DEAL base station, which sends the beacon and receives the uplink data. The yellow waves belong to the DEAL node, where the direction is reversed. At the beginning of received frames, one can see some irregular waveforms



Figure 11. Photo of the DEAL communication module.

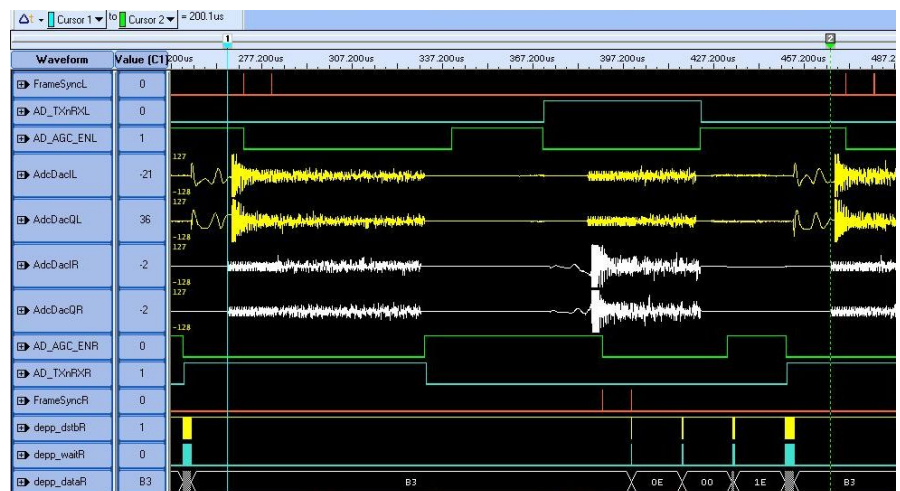


Figure 12. Screen shot from Logic Analyzer depicting a DEAL frame transfer.

that are due to settling of the AGC (automatic gain control) integrated into the AD-9361 radio receiver.

The blue waves labelled “AD_TXnRX” indicate where a module is transmitting. The green waves labelled “AD_AGC_EN” indicate where the AGC is frozen, either because of the station is in transmit state or when the AGC has settled after roughly half of the preamble. The pink waves show the synchronization pulses “FrameSync” that indicate correct preamble detection. At the bottom, the data traffic at the MAC-PHY interface is visible. Individual bytes are not resolved on this time scale, but can be observed in detail by zooming in on the time axis.

From the two cursors one can see that the super-frame period is 0.2 ms. This is a bit longer than one would expect from **Figure 4** for one node. The reason is that due to non-optimal TX ↔ RX turnaround time of the AD-9361 chip, the inter-frame spacing must currently be set to values around 32 μ s instead of 8 μ s

(= 2 OFDM symbols) as presumed when calculating the curves in section 3.2.

5. Conclusion

An ultra-low-latency and high-reliability WLAN MAC protocol for industry automation applications has been designed, simulated and implemented fully in hardware. We gave a description of the functionality, the digital design flow and implementation of the proposed DEAL MAC based on SystemC and VHDL/Verilog. We have studied and simulated the latency performance of the proposed DEAL MAC as function of network parameters like number of nodes, data traffic, PHY data rate, and others. The simulation results have shown that in our proposed DEAL MAC a network with 10 nodes can be served within 0.5 ms of cycle time (super-frame period). Cycle time scales linearly with the number of nodes. Due to MAC and PHY overhead, it is not linear with PHY data rate and payload data size. Uplink data traffic suffers much more under these problems than downlink traffic.

Acknowledgements

The research leading to these results has received funding from the German Federal Ministry for Education and Research (2015-2017) under the grant agreement No. 16KIS0179 also referred as DEAL. The authors wish to thank the DEAL project partners for fruitful contributions. Special thanks go to Bastian Lindner and Elena Sobotta from TU Dresden for their contribution to section 4.3, in particular **Figure 10**.

References

- [1] Aijaz, A., Dohler, M., Aghvami, A.H., Friderikos, V. and Frodigh, M. (2017) Realizing the Tactile Internet: Haptic Communications over Next Generation 5G Cellular Networks. *IEEE Wireless Communications*, **24**, 82-89.
<https://doi.org/10.1109/MWC.2016.1500157RP>
- [2] Knezic, M., Dokic, B. and Ivanovic, Z. (2011) Increasing EtherCAT Performance Using Frame Size Optimization Algorithm. *Proceedings of 16th IEEE Conference on Emerging Technologies for Factory Automation (ETFA)*, Toulouse, 5-9 September 2011, 1-4. <https://doi.org/10.1109/ETFA.2011.6059157>
- [3] Prytz, G. (2008) A Performance Analysis of EtherCAT and PROFINET IRT. *Proceedings of IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Hamburg, 15-18 September 2008, 408-415.
<https://doi.org/10.1109/ETFA.2008.4638425>
- [4] EtherCAT Technology Group. <https://www.ethercat.org/en/technology.html>
- [5] PROFINET Technology. <http://us.profinet.com/technology/profinet/>
- [6] SERCOS Automation Bus. <http://www.sercos.org/technology>
- [7] Saifullah, A., Lu, C., Xu, Y. and Chen, Y. (2010) Real-Time Scheduling for WirelessHART Networks. *Proceedings of 31st IEEE Real-Time Systems Symposium (RTSS)*, San Diego, 30 November-3 December 2010, 150-159.
- [8] Mayank, J., et al. (2011) Practical, Real-Time, Full Duplex Wireless. *Proceedings of 17th Annual International Conference on Mobile Computing and Networking*

- (*MobiCom*), Las Vegas, 19-23 September 2011, 301-312.
- [9] Hua, Q. and Qin, C. (2013) TDMA-Based Real-Time Delay Routing with Emergency Data in IWSN. *Sensors & Transducers*, **25**, 203-210.
 - [10] Yilmaz, O.N., Wang, Y.-P.E., Johansson, N.A., Brahmi, N., Ashraf, S.A. and Sachs, J. (2015) Analysis of Ultra-Reliable and Low-Latency 5G Communication for a Factory Automation Use Case. *Proceedings of IEEE International Conference on Communication Workshop (ICCW)*, London, 8-12 June 2015, 1190-1195. <https://doi.org/10.1109/ICCW.2015.7247339>
 - [11] DEAL Project, BMBF, Germany. <http://industrialradio.de/Projects/Home/DEAL>
 - [12] DEAL Project, IHP, Germany. <http://www.ihp-ffo.de/en/research/wireless-systems-and-applications/projects/deal.html>
 - [13] IEEE Computer Society (2007) IEEE Standard for Information Technology—Local and Metropolitan Area Networks-Specific Requirements Part 11: Wireless LAN MAC and PHY Specifications. IEEE Standard 802.11. <http://standards.ieee.org/about/get/802/802.11.html>
 - [14] IEEE Computer Society (2007) IEEE Standard for Information Technology—Local and Metropolitan Area Networks-Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). IEEE Standard 802.15.4. <http://standards.ieee.org/about/get/802/802.15.html>
 - [15] Odhah, N., Tittelbach-Helmrich, K. and Stamenković, Z. (2017) A Low-Latency Medium Access Control Protocol for Industrial Wireless LAN Applications. *Proceedings of 30th International Conference on Microelectronics*, Nis, 243-246.
 - [16] Tittelbach-Helmrich, K. and Stamenković, Z. (2016) Implementing a Medium Access Control Protocol in Pure Hardware using System C. *Dresdner Arbeitstagung Schaltungs- und Systementwurf*, Cottbus, 69-73.
 - [17] Tittelbach-Helmrich, K. and Stamenković, Z. (2016) Hardware Implementation of a Medium Access Control Layer for Industrial Wireless LAN. *Proceedings of IEEE International Symposium on Design and Diagnostics of Electronic Circuits & Systems*, Kosice, 62-67. <https://doi.org/10.1109/DDECS.2016.7482465>
 - [18] Lindner, B., Joram, N. and Ellinger, F. (2017) Study of Combined Indoor Communication and Localisation Systems for Industrial Applications. *Proceedings of European Wireless Conference (European Wireless)*, Dresden, 1-3.
 - [19] Stamenković, Z., Miletić, E., Obrknezev, M. and Tittelbach-Helmrich, K. (2009) MAC Protocol Implementation in RF-MIMO WLAN. *Proceedings of 16th IEEE International Conference on Electronics, Circuits, and Systems*, Yasmine Hammamet, 303-306. <https://doi.org/10.1109/ICECS.2009.5410996>
 - [20] Tittelbach-Helmrich, K., Miletić, E., Wcislek, P. and Stamenković, Z. (2010) MAC Hardware Platform for RF-MIMO WLAN. *Proceedings of 53rd IEEE International Midwest Symposium on Circuits and Systems*, Seattle, 339-342. <https://doi.org/10.1109/MWSCAS.2010.5548838>
 - [21] Loeb, H.-P., Gries, M. and Sauer, C. (2009) Implementing a Software-Based 802.11 MAC on a Customized Platform. *Proceedings of 6th IEEE Consumer Communications and Networking Conference*, Las Vegas, 1-6. <https://doi.org/10.1109/CCNC.2009.4784695>
 - [22] Krstić, M., Maharatna, K., Troya, A., Grass, E. and Jagdhold, U. (2004) Baseband Processor for IEEE 802.11a Standard with Embedded BIST. *Facta Universitatis, Series: Electronics and Energetics*, **17**, 231-239. <https://doi.org/10.2298/FUEE0402231K>