

# Intelligent TDMA Heuristic Scheduling by Taking into Account Physical Layer Interference for an Industrial IoT Environment

Margot Deruyck · Jeroen Hoebeke · Eli De Poorter · Emmeric Tanghe ·  
Ingrid Moerman · Piet Demeester · Luc Martens · Wout Joseph

Received: date / Accepted: date

**Abstract** In an Internet of Things environment, where multiple mobile devices are brought together, it is not always possible to serve all these devices simultaneously. We developed an intelligent TDMA (Time Division Multiple Access) scheduler which allows to plan the individual packets of the different streams in such a way that everyone can be served by taking into account the interference on the physical layer. The scheduler is applied in a realistic industrial environment and evaluated based on the maximum link latency, the channel occupancy, and the jitter. Two strategies are compared: one where the packets are sequentially allocated, and one periodically. Our results show that the periodically allocated strategy performs the best for the maximum link latency (for a packet size below 1200 bytes) and for the jitter. The channel occupancy is similar for both strategies. Furthermore, the performance can be improved by using a higher number of channels. Compared to classic CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), the channel occupancy and the jitter are reduced up to 69.9% and 99.9%, respectively. Considering the maximum link latency, the proposed TDMA strategies perform significantly better than the worst case CSMA/CA (up to 99.8%), however, when assuming a best case CSMA/CA scenario, CSMA/CA performs better. Furthermore, we clearly show that there are cases where it is not possible to plan all streams when using CSMA/CA while this becomes feasible when applying the proposed TDMA strategies.

**Keywords** CSMA/CA · heuristic scheduler · industrial environment · Internet of Things (IoT) · physical layer interference · TDMA

## 1 Introduction

In recent times, the mobile world around us has rapidly changed. Today, we no longer have just a laptop, a smart phone, and a tablet, but also wearables such as smart (sport)watches, activity trackers, and smart glasses. Furthermore, in industrial environments, machines are equipped with wireless radios and start to communicate with each other. In 2015, more than half a billion mobile devices and connections were added. By 2020, there will be 11.6 billion mobile-connected devices, including M2M (Machine To Machine) modules [24]. These numbers clearly show that we are moving towards an IoT (Internet of Things) environment, where everything in our daily life will be connected. Especially in industry this is an ongoing evolution as operators are continuously looking for ways to further automate processes, improve efficiency, and increase economic benefits. An industry where everything becomes connected to a network (e.g., the Internet or a private factory network) by means of a communication infrastructure is often referred to as Industry 4.0 [2, 3]. In some cases, wired solutions might be possible, however, at an excessive wiring cost. Wireless technologies are gradually being adopted to realize the required communication functionality and to offer a viable and cost-efficient alternative. For instance, the uptake of handheld devices on the work floor, the use of mobile robots such as Automated Guided Vehicles or the tracking of goods, all necessitate the use of wireless solutions. According to [3, 4], wireless networks can have an advan-

---

All authors  
IMEC - Ghent University - Department of Information Technology  
Technologiepark-Zwijnaarde 15 (iGent tower)  
B-9052 Ghent  
Belgium  
E-mail: margot.deruyck@ugent.be

tage in industry including the lower installation costs due to cabling and hardware reduction, the lower operational costs by eliminating cable failures, ability to connect hard-to-reach and remote areas, gains in productivity and efficiency due to equipment mobility, the higher flexibility of reorganizing the process structures, and finally a higher productivity and less downtime due to personnel mobility.

For the realization of this wireless connectivity, a variety of wireless communication technologies exist such as WiFi (Wireless Fidelity), WirelessHART (Wireless Highway Addressable Remote Transducer protocol), ISA 100 (International Society of Automation), Bluetooth, IEEE 802.15.4, etc. [3]. Each of these technologies operates in the ISM (Industrial, Scientific and Medical radio bands) band and requires a well-designed MAC (Medium Access Control) protocol to give every device a fair share of the wireless medium and to avoid collisions. In the context of industrial connectivity, devices will typically have very specific requirements in terms of reliability, data rate, latency, jitter, etc. To meet such requirements, MAC protocols are needed that enable fine-grained control over whom gets access to the wireless medium, at which moment in a contention-free way (no collisions). Typically, this results in TDMA-based (Time Division Multiple Access) protocols that provide different time slots to different data streams in a cyclically repetitive frame structure [5]. The resulting performance is strongly determined by the frame structure and slot allocation and requires intelligent scheduling when multiple mobile devices with their own requirements are brought together.

Today, TDMA-based solutions for 802.15.4 networks have become available and research on scheduling algorithms is ongoing [5–8]. Barring some exceptions [9, 10], today’s 802.11 based solutions, however, still make use of CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance), avoiding collisions by only transmitting when the channel is sensed idle [11]. In order to deal with very specific traffic requirements such as the ones encountered in industrial settings, it is expected that future industrial WiFi systems might also move to TDMA-based MAC protocols in combination with intelligent scheduling.

To anticipate this evolution, this paper proposes a solution for this problem by introducing an intelligent TDMA (Time Division Multiple Access) scheduler which allows to plan the individual packets of the different streams in such a way that all streams can be served and the interference on physical layer is minimized. The scheduler is demonstrated for an industrial environment. A TDMA scheduler for WiFi 802.11n that takes into account the inter-channel interference

is novel and has, to the best of our knowledge, never been proposed before. Note that our scheduler is only compliant with the IEEE 802.11n specification of the physical layer, since we have replaced the MAC layer by our own TDMA scheduler. Furthermore, a comparison is made between the proposed novel TDMA strategies and CSMA/CA, which is representative for today’s WiFi networks.

The outline of this paper is as follows. In the next section, we discuss the related work. In Section 3, the heuristic algorithm of the scheduler is discussed. Section 4 proposes the considered scenario and the different strategies and cases are evaluated for three different metrics: the maximum link latency, the channel occupancy, and the jitter. Section 5 summarizes the most important conclusions obtained from the results of Section 4.

## 2 Related work

As mentioned above TDMA-based solutions for WSN (Wireless Sensor Networks) networks have become available. Existing industrial WSN technologies have demonstrated that the IEEE 802.15.4e TSCH (Time Slotted Channel Hopping) effectively enables industrial-grade deterministic properties for control loops with low latency, ultra-low jitter, ultra-low power consumption and a high reliability [5]. Furthermore, the basic concept of TSCH is also incorporated in standards such as WirelessHART and ISA100.11a. [6] introduces Orchestra in which nodes autonomously compute their own, local schedules. They maintain multiple schedules, each allocated to a particular traffic plane (application, rating, MAC) and updated automatically as the topology evolves. Orchestra exploits the robustness of TSCH. In [7], an On-The-Fly (OTF) bandwidth reservation module plays a complementary role for TSCH. This is a distributed approach for adapting the scheduled bandwidth to the network requirements. Finally, [8] proposes a distributed PID (Proportional, Integral, and Derivative) based control for TSCH. This distributed scheduling policy is based on the well-known industrial control paradigm referred to as PID control. The proposed technique is completely decentralized, enabling the schedule to one another, according to its traffic demand.

Considering WiFi, all solutions for the considered industrial scenario are largely CSMA-based. Techniques such as PCF (Point Coordination Function) and EDCA (Enhanced Distributed Channel Access) are incorporated in the standard, but even with this kind of QoS extensions, WiFi cannot handle real-time traffic properly as shown in [12, 13]. In literature, some TDMA-

based solutions for WiFi are proposed to tackle these limitations. One of these solutions is RT-WiFi [14]. RT-WiFi is a TDMA data link layer protocol based on the IEEE 802.11 physical layer to provide deterministic timing guarantee on packet delivery and high sampling rate up to 6 kHz. It includes a link scheduler as component, but no real scheduling such as in our approach. [15] investigates an isochronous wireless communication system consisting of a deterministic MAC based on IEEE 802.11 and extended with additional features for isochronous communication. It uses IsoMAC which is based on TDMA and divides the communication into two different phases, a scheduled phase for real-time data and a contention phase for best-effort traffic. Only real-time cyclic traffic is thus considered here. Finally, [16] proposes OpenTDMF, an architecture to enable TDMA on commodity WLAN devices and is related to SDN (Software Defined Networking).

This overview of related work shows that there is definitely a need for appropriate scheduling mechanisms for IEEE 802.11. Considering schedulers, it is worth mentioning the contributions of [17,18]. [17] proposes an extension for the RT-WiFi of [14]. As discussed above, RT-WiFi tries to achieve predictable packet delivery latency. [17] proposes a RT-WiFi network manager design and algorithms for controlling jitter in network controlled systems. This is done by the HCJF (Harmonic Chain Jitter Free) scheduler which selects the sampling period for each communication task and by doing so eliminates the transmission jitter. However, it does not account for physical layer interference. The SchedWiFi of [18] provides flexible support to the scheduled traffic class i.e., a high priority traffic class that is transmitted according to a fixed schedule, over IEEE 802.11 ad-hoc industrial networks. However, it modifies the EDCA QoS mechanism allowing to transmit scheduled traffic without requiring any predefined superframe structure, or time slots, and can thus not be considered as a pure TDMA scheduler.

### 3 Heuristic TDMA scheduler

In this section, the heuristic algorithm of the TDMA scheduler is discussed. Fig. 1 shows the flow diagram of the algorithm. The algorithm can be divided in three major blocks: input block, making wireless connections (connecting block), and the scheduling block. Each block is discussed in detail in the following subsections.

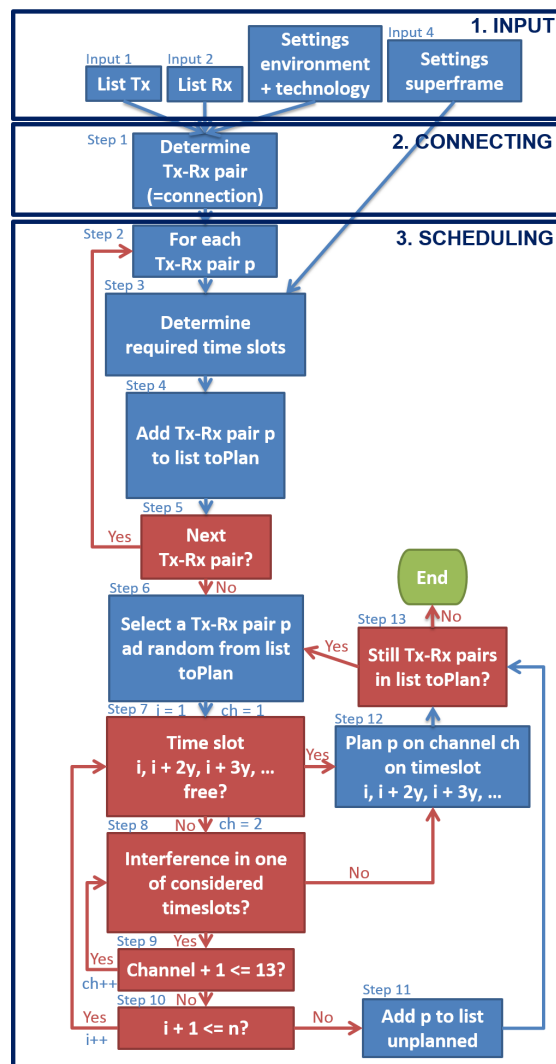


Fig. 1 Flow diagram of the heuristic algorithm.

#### 3.1 Assumptions

WiFi 802.11n is here considered as wireless technology as it has dominated wireless since it was introduced in 2009 and can be found in most homes and businesses today [19]. The performance of the different strategies and cases will be compared for three different metrics: maximum link latency, channel occupancy, and jitter. Each of these metrics will be discussed in detail in the following subsections. The packet size is varied from 100 bytes to 1500 bytes and is considered to be fixed. Smaller packet sizes are representative for monitoring applications where sensor systems periodically transmit collected, similarly sized monitoring data or process control applications where you have continuous periodic communication in both uplink and downlink. Larger packet sizes are more representative for more bandwidth demanding applications such as au-

dio or video streaming. Modelling the traffic streams in such a way, does not limit the proposed scheduling technique to constant bit rate applications. Industrial applications may also result in unpredictable traffic (i.e. traffic that can occur any moment) that needs to be delivered within a certain time frame. To accommodate such traffic streams in a schedule and to meet the latency requirements, it is mandatory to allocate slots in a similar way as for CBR traffic. The only difference however is that not every slot will be used at all times. We assume a centralized controller which broadcasts the schedule to all nodes. Furthermore, simultaneous use of overlapping channels is possible in certain conditions. Finally, the sender and receiver pairs are known in advanced, meaning that we know their locations and their traffic requirements. The 95th percentile over 150 simulations is considered here. The sequence in which the packets are planned will differ from simulation to simulation, resulting in a different schedule and thus different performance.

### 3.2 Input

As for every algorithm, some input is required before starting the calculations. Here, the input can be divided in four groups as shown in Fig. 1 (Block 1, Input):

- List of transmitters (Fig. 1, Input 1): For every transmitter present in the considered environment, the  $(x, y, z)$  coordinates are provided. Furthermore, it is also indicated which type of receiver can connect to each transmitter.
- List of receivers (Fig. 1, Input 2): For every receiver present in the considered environment, the  $(x, y, z)$  coordinates are provided, along with the required physical bit rate.
- Settings environment and technology (Fig. 1, Input 3): This input consists of two settings files: one describing the environment (e.g., an industrial environment) which allows us to select the most appropriate propagation model and one describing the technology. The other file describing the technology contains all the relevant link budget parameters as discussed in [20]. Note that considering the technology, we assume here homogeneity, meaning that all transmitters and receivers are supporting the same technology. However, the receivers have different bit rate requirements, resulting in heterogeneous traffic.
- Settings superframe (Fig. 1, Input 4): this input parameter indicates how many bytes are used per packet.

### 3.3 Connecting

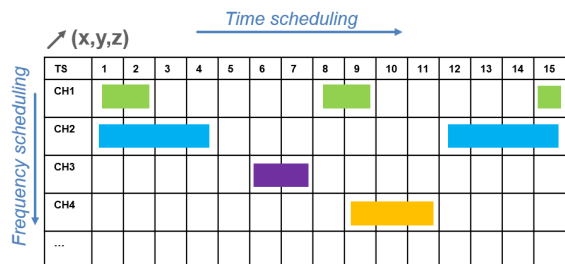
The second block in Fig. 1, "connecting", is responsible for determining which receiver connects with which transmitter. In case a receiver can only connect to one transmitter, the choice is obvious, but when the receiver can deliver its data to multiple transmitters, it should be connected to the one from which it receives the best signal quality. By taking into account the location of the receiver and the transmitter (Fig. 1, Input 1 and 2), the selected propagation model (Fig. 1, Input 3), and the link budget parameters (Fig. 1, Input 3), we can determine the path loss between each transmitter and the receiver. The transmitter from which the receiver experiences the lowest path loss is the one to which the receiver should be connected (Fig. 1, Step 1). This will be considered as a transmitter-receiver pair or stream from then on. Note that interference between the different connections is not accounted for in this stage, this will be taken into account when we are actually scheduling the connection in a certain time slot.

### 3.4 Scheduling

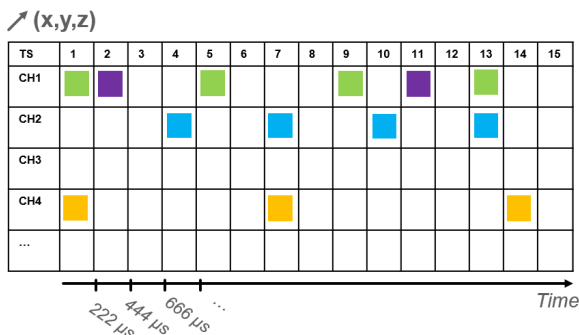
The last block, scheduling, in Fig. 1 contains all the logic to allocate time slots to the different connections or transmitter-receiver pairs. For each transmitter-receiver pair (Fig. 1, Step 2), the required number of time slots is determined (Fig. 1, Step 3). In this study, two different strategies are considered:

- Strategy I - Sequentially allocated: based on the required bit rate and the packet size (in bytes, Fig. 1, Input 4), we calculate how many time slots the transmitter - receiver pair needs. These time slots will consecutively be allocated for the considered stream. Note that this strategy is a sort of allocation where you have no limits on how frequent a node can send and thus a sequence of time slots can be allocated. Fig. 2 (a) shows a possible outcome of the scheduler when applying this strategy. The sequentially allocated strategy is denoted as "Sequentially" in the figures of this paper.
- Strategy II - Periodically allocated: similar as for the other strategy, we calculate the number of time slots required by the transmitter-receiver pair based on the demanded bit rate and the packet size. However, here, an extra requirement is imposed on the scheduling of a stream by taking into account that the stream demands a time slot every  $x$  time slots. Examples include periodic reporting or closed loop optimizations [21, 22]. This periodicity of  $x$  time slots can be calculated based on the bit rate offered by

the considered technology, the bit rate demanded by the stream, and the packet size. Fig. 2 (b) shows a possible outcome of the scheduler when applying this strategy. The periodically allocated strategy is denoted as "Periodically" in the figures of this paper.



(a) Example of the obtained schedule when using the sequentially allocated strategy.



(b) Example of the obtained schedule when using the periodically allocated strategy.

**Fig. 2** Comparison of the obtained scheduling when applying the two different strategies.

Once the number of required time slots (and the periodicity in case of the latter strategy) are known (Fig. 1, Step 3) for the considered stream, the transmitter-receiver pair is added to the list with streams that need to be planned (Fig. 1, Step 4). Steps 2 to 4 are repeated until the number of required time slots (and if needed the periodicity) of all connections is known (Fig. 1, Step 5).

In the next steps (Fig. 1, Steps 6 to 10), we will try to schedule the packets for each of the streams. Which transmitter-receiver pair will be handled first is chosen randomly (Fig. 1, Step 6). Each pair will be scheduled in time and in frequency as discussed in the following subsections.

### 3.4.1 Frequency scheduling

The algorithm will always start to plan the first packet of the considered stream in time slot 1 on channel 1. If

the algorithm detects that the time slots  $1, 1 + y, 1 + 2 \cdot y, \dots, 1 + m \cdot y$  (with  $y$  the periodicity which equals 1 when applying the sequentially allocated strategy and  $1 + m \cdot y \leq$  the maximum number of time slots available) are already occupied on channel 1, it will try to plant the streams in the same time slots but on another channel (2, 3, etc.). If all these time slots are free, the algorithm will plan the packets of this stream in these time slots on channel 1 (Fig. 1, Step 12). The frequency scheduling is indicated in Fig. 2(a) by a vertical arrow. Once the stream has been planned, the algorithm will continue with scheduling the next transmitter-receiver pair if any still available (Fig. 1, Step 13).

As already mentioned above, if the time slots  $1, 1 + y, 1 + 2 \cdot y, \dots, 1 + m \cdot y$  are already occupied on channel 1, the algorithm will try to schedule the stream in the same time slots but on a different channel. The packets can only be scheduled when the interference from the other streams planned in these time slots is low enough in order for the transmitter to obtain a good signal (Fig. 1, Step 8). This is contrast to traditional WiFi approaches in which only 3 non-overlapping channels (often 1, 6, and 11) are used. How the interference between the different scheduled streams is determined will be discussed in the next subsection. If the interference is low enough, the packets of the transmitter-receiver pair will be scheduled in time slots  $1, 1 + y, 1 + 2 \cdot y, \dots, 1 + m \cdot y$  on the considered channel (Fig. 1, Step 12) and the algorithm will proceed with scheduling the next stream if any still available (Fig. 1, Step 13). If the interference is too high, the algorithm will try to schedule the stream on the next channel (Fig. 1, Step 9) if still channels available to check (Fig. 1, Step 10) and repeat Steps 7 to 9 if needed. If all the channels are checked and no match is found, the algorithm will try to schedule in time (Fig. 1, Step 10) as discussed in subsection 3.4.2.

### 3.4.2 Time scheduling

As discussed above, when all channels are checked for a time slot and no match is found, the algorithm will try to schedule in the next time slot (Fig. 1, Step 10). The time scheduling is indicated as a horizontal arrow in Fig. 2(a). Adding TDMA i.e., time scheduling, to the scheduler is very novel compared to existing schedulers as mentioned in Section 1.

When a new time slot is selected, the frequency scheduling is repeated. In case no match is found with any time slot or channel, it will not be able to plan the considered stream (Fig. 1, Step 11).

### 3.4.3 Interference

To determine if a certain stream can be planned in a time slot on a certain channel, the interference between the already planned streams in this time slot and this ‘new’ stream should be determined. In this section, the transmitter and the receiver from the stream that needs to be planned will be indicated by  $Tx_c$  and  $Rx_c$  i.e., the considered transmitter and receiver respectively. Therefore, the SNR (Signal-to-Noise Ratio) experienced by each already planned transmitter  $Tx_p$  from  $Tx_c$  has to be determined. To this end, the amount  $P$  of overlap between the channels on which  $Tx_p$  and  $Tx_c$  are operating needs to be known, as well as the power  $P_r$  received by  $Rx_c$  from  $Tx_p$ . Indeed, only a part  $P$ , represented by the overlap of the channels, of  $P_r$  will be taken into account in the SNR calculation as shown in Fig. 3. To determine the amount  $P$  of overlap between the channels on which  $Tx_p$  and  $Tx_c$  are operating, the spectral masks of these channels are determined which can be found in the standard [11]. Fig. 3 shows as an example of the spectral masks assuming  $Tx_p$  is operating on channel 1 (in blue) and  $Tx_c$  on channel 5 (in orange). To calculate the amount of overlap between the two channels, the procedure proposed in [23] is used. Therefore, the surface  $S_{tot}$  of the full spectral mask of one of the channels is determined which can easily be done by integrating the function describing the spectral mask. For example for channel 1, this corresponds with the surface below the blue line in Fig. 3. Next, the surface  $S_{overlap}$  of the gray hatched area in Fig. 3 is determined; this can again be done by integrating the corresponding parts of the function describing the spectral mask of each channel. The percentage  $P$  of overlap is calculated as the ratio  $\frac{S_{overlap}}{S_{total}} \cdot 100$ .

To determine the power  $P_r$  received from  $Tx_p$  in  $Rx_c$ , an appropriate propagation model will be used as the distance between  $Tx_p$  and  $Rx_c$  is known. The intended  $SNR$  (in dB) is then determined as follows:

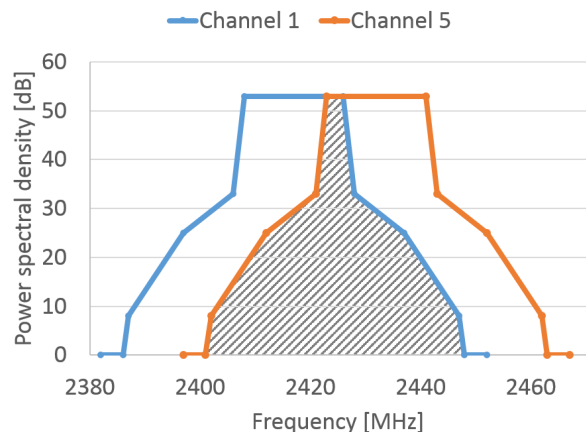
$$SNR = 10 \cdot \log_{10} P + P_r - NF \quad (1)$$

with  $NF$  the noise floor (in dB) i.e., the level of noise introduced by the system itself.

The experienced interference  $I$  (in dB) is then a combination of the SNRs from all of the other transmitter planned in the considered time slot:

$$I = 10 \cdot \log_{10} \sum_{i=1}^N 10^{\frac{SNR_i}{10}} \quad (2)$$

with  $N$  the number of other transmitters planned in the considered time slot and  $SNR_i$  the SNR obtained by transmitter  $Tx_i$  (Eq. 1).  $I$  will be included in the final link budget. This link budget will be used to calculate the maximum allowable path loss between  $Tx_c$



**Fig. 3** Determining overlap between the spectral masks of two different streams or transmitter and receiver pairs ( $S_{overlap}$  = hatched area).

and  $Rx_c$ . If the experienced path loss between  $Tx_c$  and  $Rx_c$  (calculated by an appropriate propagation model) is higher than the maximum allowable path loss, the interference is too high and the transmitter-receiver pair can not be planned in this time slot and/or on the considered channel.

Note that the above described procedure needs to be repeated for every stream already planned in the considered time slot in order to make sure that the ‘new’ stream does not interfere with any of them. Furthermore, note also that we only assume interference between the nodes and not from other signals present in the environment.

### 3.5 Cases

In this study, three different cases are investigated:

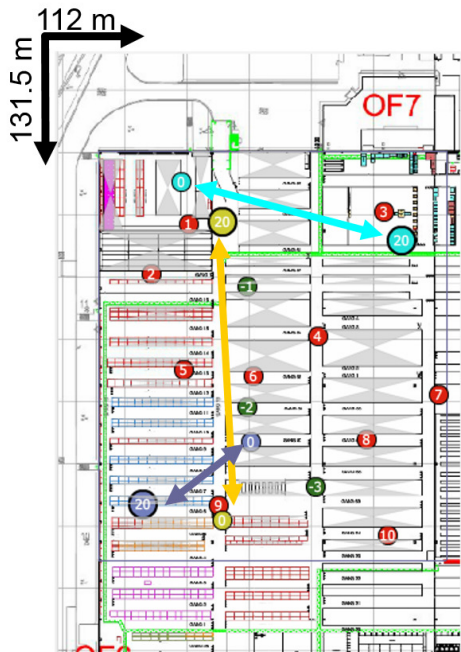
1. Only 1 out of the 13 channels of the WiFi 802.11n standard can be used which is a worst case scenario.
2. Only 3 out of the 13 channels of the WiFi 802.11n standard can be allocated. These are the three available non-overlapping channels: channels 1, 6, and 11. The approach of using only these 3 channels is used in literature [24–26].
3. All the 13 channels of the WiFi 802.11n standard can be used. This is the novel approach described in this paper.

Each of these cases will be combined with the two strategies (sequentially allocated and periodically allocated) of Section 3.4, resulting in 6 different combinations to compare.

### 3.6 Scenario

An indoor industrial scenario is assumed. Fig 4 shows the blueprint of the considered scenario in a processing factory. Four different types of streams are considered:

1. 5 Mbps stream (downlink): this corresponds with video streaming in HD (High Definition) quality. This type of stream is set up between the nodes connected by the cyan arrow in Fig. 4. The largest dot is the transmitter (located at 6 m), the smallest the receiver (2.5 m height).
2. 1 Mbps stream (downlink): this represents a downloading process. This type of stream is set up between the two nodes connected by the yellow arrow in Fig. 4. The largest dot is again the transmitter (6 m height), the smallest the receiver (2.5 m height).
3. 320 kbps stream (downlink): this corresponds with audio streaming. This type of stream is only requested between the two nodes connected in purple in Fig. 4. As mentioned above, the largest dot is the transmitter (6 m height), the smallest the receiver (2.5 m height).
4. 8 kbps stream (downlink): this corresponds with the transmitting process of sensor data. All the red dots in Fig. 4 (2.5 m height) can receive data from all the green dots in Fig. 4 (2.5 m height).



**Fig. 4** Industrial scenario with 5 different types of traffic stream.

Furthermore, we assume that all nodes are supporting the WiFi 802.11n standard [11]. For the considered indoor industrial environment, the propagation model proposed in [27], which is based on actual measurements in processing factories, is used for all simulations. Since we are assuming an industrial environment, nLoS (non-Line-of-Sight) is considered. The path loss in dB as a function of the distance  $d$  in m between the transmitter and the receiver is determined as follows:

$$PL(d) = PL(d_0) + 10n \cdot \log\left(\frac{d}{d_0}\right) \quad (3)$$

with  $PL(d_0)$  the path loss in dB at an arbitrarily chosen reference distance  $d_0$  in m and  $n$  the dimensionless path loss exponent. Based on the measurements of [27] for the 2.4 GHz frequency band, the following values are:  $PL(d_0) = 71.84$  dB at a reference distance  $d_0$  of 15 m,  $n = 2.16$ , and a standard deviation  $\sigma$  of 8.13 dB. Table 1 shows the considered values for the different link budget parameters.

Parameter	Value
Frequency	2.4 GHz
Input power antenna transmitter	20 dBm
Antenna gain transmitter	Based on radiation pattern
Antenna gain receiver	Based on radiation pattern
Feeder loss transmitter	0.5 dB
Feeder loss receiver	0 dB
Yearly availability	99.995%
Bandwidth	20 MHz
Receiver SNR	1/2 BPSK = 4.5 dB (6.5 Mbps)
Number of data carriers	56
Number of total carriers	64
Implementation loss receiver	0 dB
Antenna height transmitter	6 m or 2.5 m (as mentioned in Section 3.6)
Antenna height receiver	2.5 m
MIMO gain	0 dB
Noise figure receiver	10 dB
Fade margin	10 dB
Soft handover gain	0 dB

**Table 1** Link budget parameters for WiFi 802.11n in the industrial environment of Fig. 4 [28,29].

### 3.7 Metrics

The different strategies will be evaluated for three different metrics: maximum link latency, channel occupancy, and jitter. In this section, the definition and - if possible - the formula for the different metrics is discussed.

#### 3.7.1 Maximum link latency

The maximum link latency  $LL$  is defined as the maximum delay (worst-case) between the time a packet becomes available for transmitting and the time the

packet is actually sent (in microseconds). We assume that an application requires one slot every  $x$  slots. One slot has a duration of  $S$  microseconds. Depending on which strategy is applied, a different formula is used.

*TDMA - Strategy I - Sequentially allocated* When applying this strategy,  $x$  sequent slots will be allocated every  $y$  slots (as shown in Fig. 2(a)). The worst-case scenario occurs when the first packet becomes available just after the beginning of the  $x$ -th (i.e., the last) sequent slot. This slot can not be used anymore and the sender has to wait  $(y - x + 1)$  more slots before the packet can be sent. This results in a maximum link latency of  $(y - x + 1) \cdot S$  microseconds ( $S =$  slot duration).

*TDMA - Strategy II - Periodically allocated* As shown in Fig. 2(b), the periodicity of one slot every  $x$  slots is here guaranteed. The worst-case scenario occurs when the packet is ready to be sent when its slot has just started. The packet has to wait  $x$  more slots until a next slot allocated to this application. This results in a maximum link latency of  $x \cdot S$ .

*CSMA/CA* The best case for CSMA/CA occurs when a node can transmit as first one. In this case it depends on your back-off size how large the latency becomes. In the very best case, the packet has to wait only for one DIFS (DCF (Distributed Coordination Function) Interframe Space). One DIFS for WiFi 802.11n has a duration of  $34 \mu s$  [11]. This case is denoted as "CSMA best" in the figures of this paper. On average, the packet has to wait  $DIFS + \frac{CW}{2} \cdot S$  with  $CW$  the considered contention window and  $S$  the duration of a time slot (in microseconds). This case is denoted as "CSMA best on average  $CW = cw$ " with  $cw$  the chosen contention window. The maximum time that the packet has to wait, even if it can be send as first one is:  $DIFS + CW \cdot S$ . This case is denoted as "CSMA best max  $CW = cw$ " in the figures of this paper. The worst-case CSMA/CA scenario is when the packet has to wait until all the other streams in the environment have sent a packet. In this case, the maximum link latency  $LL$  is determined as follows:

$$LL = \frac{\sum_{i=1}^n (DIFS + CW \cdot S + \sum_{j=1, j \neq i}^n p_j)}{n} \quad (4)$$

with  $n$  the number of streams,  $CW$  the contention window,  $S$  the slot duration in  $\mu s$ , and  $p_i$  the duration to send one packets of stream  $i$  in  $\mu s$ . Again, the duration of the DIFS is  $34 \mu s$ . In the figures of this paper, this will be denoted as "CSMA worst  $CW = cw$ " with  $cw$  the considered contention window. Note that we do not account for retransmissions and collisions, which will increase the link latency even further.

### 3.7.2 Channel occupancy

The average channel occupancy is defined as the amount of time that the available channels are occupied and is expressed as a percentage. The average channel occupancy  $CO$  is calculated as follows:

$$CO = \frac{TS \cdot S}{S_{tot}} \cdot 100 \quad (5)$$

with  $TS$  the number of used time slots over a duration of  $S_{tot}$  (in seconds),  $S$  the duration of the time slot (in seconds), and  $n$  the number of available channels.

The duration of the time slot  $S$  accounts not only for the time needed to send the application data but also for the overhead. The 802.11n physical frame consists of  $20 \mu s$  (fixed time) for the preamble and signal, 16 bits service data, the PSDU (Physical layer Service Data Unit) payload, and 6 tail bits. The PSDU payload on his turn consists of 28 bits overhead and the MSDU (MAC Service Data Unit) payload. The MSDU payload contains the LLC (Logical Link Control) header of 8 bits and the IP (Internet Protocol) packet [11]. Assuming IPv4 and UDP (User Datagram Protocol), the IP packet has a size of 20 bytes for the IPv4 header, 8 bytes for the UDP header, and  $X$  bytes for the application data. Taking all the above into account, the duration of the time slot  $S$  is determined as follows (in microseconds):

$$S = 20 \mu s + \frac{22 + (28 + 8 + 20 + 8 + X) \cdot 8}{B} \quad (6)$$

with  $X$  the packet size (in bits) and  $B$  the physical bit rate (in Mbps). As we assume here WiFi 802.11n,  $B$  equals 67.5 Mbps. Note that a channel occupancy of more than 100% can be obtained. This means that it will be impossible to serve all the streams.

### 3.7.3 Jitter

The jitter is defined as the deviation of the time (in microseconds) between the moment the packet is ready to be sent and the moment that the packet is actually scheduled. The following formula is used:

$$J = \frac{\sum_{i=1}^n (TS_{s,i} - TS_{p,i}) \cdot S}{n} \quad (7)$$

with  $n$  the number of packets that needs to be sent,  $TS_{s,i}$  the number of the time slot where the packet is actually been scheduled,  $TS_{p,i}$  the number of the time slot where the packet was originally planned, and  $S$  the duration of a time slot (in seconds).

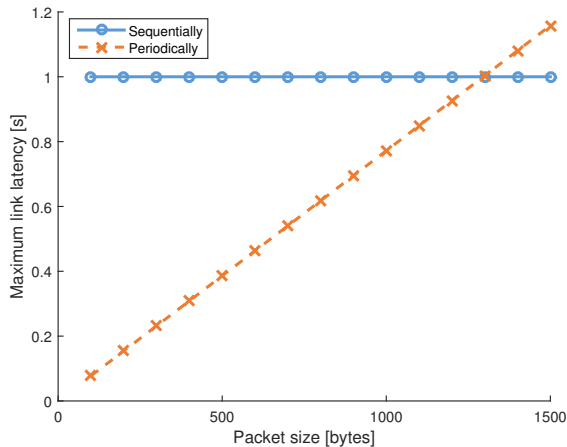


## 4 Results

### 4.1 Comparison of the different TDMA strategies

#### 4.1.1 Maximum link latency

For every type of stream described in Section 3.6, the maximum link latency is calculated as described in Section 3.7. The average of the maximum link latency (in seconds) over the 13 streams is shown in Fig. 5 as a function of the packet or slot size (in bytes) for the two considered strategies. For the sequentially allocated strategy in Fig. 5, the maximum link latency does not depend on the size of the packet or the slot. The larger the packet size, the less packets will be needed to send the same amount of data, and the shorter the maximum link latency will be in terms of slots. However, the larger the packets, the higher the duration in time of a slot. This results in the same maximum link latency expressed in time i.e., 1 second.



**Fig. 5** Comparison of the maximum link latency for the considered strategies.

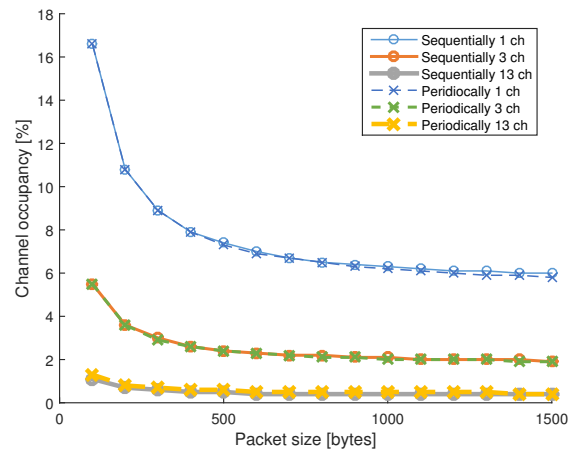
Fig. 5 shows that, on the contrary, when considering the periodically allocated strategy, the maximum link latency linearly depends on the packet size (0.08 s for 100 bytes and 1.16 s for 1500 bytes for WiFi 802.11n). The larger the packets, the larger the maximum link latency due to the fact that the ratio of 1 slot every  $x$  slots remains the same. Again, a higher slot duration is obtained for a larger packet size thus results in a higher maximum link latency.

When comparing both strategies in Fig. 5, the periodically allocated strategy performs the best for packet sizes less than 1200 bytes (for example, 0.85 s versus 1.0 s for 1100 bytes). For packet sizes greater than 1200 bytes, the sequentially allocated strategy performs

better (1.0 s versus 1.08 s for 1400 bytes). For a packet size of 1200 bytes, both strategies have an equal performance (Fig. 5).

#### 4.1.2 Channel occupancy

In this section, the average channel occupancy is compared. Ideally, the channel occupancy should be as low as possible. Fig. 6 shows the 95th percentile over 150 simulations for the different strategies and cases. Based on Fig. 6, we conclude that a higher number of available channels results in a lower channel occupancy. E.g., assuming the periodically allocated strategy with a packet size of 500 bytes, the channel occupancy equals 7.3% if 1 channel is available, 2.4% when using 3 channels, and 0.5% if there are 13 channels available. This is logical as the same amount of data needs to be sent irrespectively of the number of available channels. The higher the number of available channels, the more the traffic is spread over the different channels. As the traffic can be spread over different channels, the interference will be reduced thus allowing more packets to be sent simultaneously. This results in a lower average channel occupancy.



**Fig. 6** Comparison of the channel occupancy for the considered strategies and cases.

Fig. 6 shows that the difference in channel occupancy is very limited when comparing the two strategies. When considering only 1 channel, differences are lower than 1% (Fig. 6). As all packets should be scheduled on the same channel, the interference becomes too high. Due to this, packets can not be sent simultaneously which results in the same channel occupancy independent of the considered strategy. Also when we use 3 channels, the same channel occupancy is obtained as shown in Fig. 6. For the 13 channels case, the difference

in channel occupancy is not significant (Fig. 6). The sequentially allocated strategy performs slightly better with a 0 to 0.2% lower channel occupancy (depending on the considered packet size). The main reason is that some (parts of the) streams can be scheduled completely simultaneously when using the sequentially allocated strategy, while this is not the case when periodically assigned due to the periodicity requirement. If the periodicity of these streams is not a multiple of each other, some of their packets might not be scheduled during the same time slots.

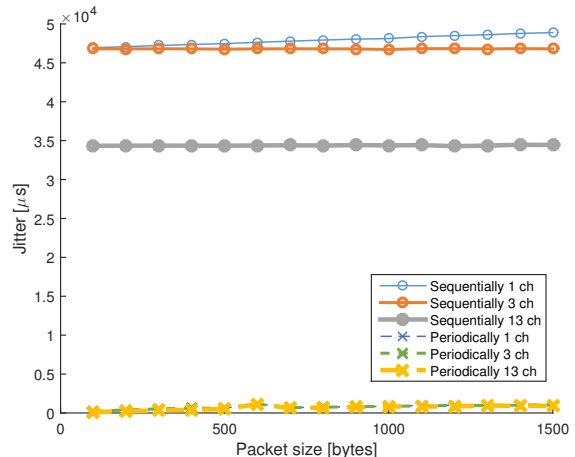
Finally, the packet size has also a high influence on the obtained channel occupancy. For each considered strategy and case, Fig. 6 that the larger the packet size, the lower the channel occupancy becomes. For example, if we consider the periodically allocated strategy and 3 available channels, a channel occupancy of 5.5% is obtained for a packet size of 100 bytes and 2.0% when using a packet size of 1000 bytes. This difference in channel occupancy is due to the overhead as it makes no difference in time if, for example, we send 1 time 1000 bytes or 10 times 100 bytes of application data. When taking into account the overhead, 146.4  $\mu\text{s}$  (Eq. 6) are needed to send 1000 bytes of application data, while sending 100 bytes of application data takes 39.8  $\mu\text{s}$  (Eq. 6). However, for the latter, we need 10 times 39.8  $\mu\text{s}$  (Eq. 6) i.e., 398  $\mu\text{s}$ , thus resulting in a higher channel occupancy.

#### 4.1.3 Jitter

In this section, the jitter is compared for the different strategies and cases. Ideally, the jitter should be as low as possible. Fig. 7 compares the jitter averaged over all the packets that need to be sent for the considered streams in Section 3.6. The 95th percentile over 150 simulations is presented.

Fig. 7 shows that the sequentially allocated strategy clearly results in a higher jitter. An improvement of 96.7% to 99.7% (depending on the considered number of channels) is found for the periodically allocated strategy compared to the sequentially allocated strategy. The reason is twofold. First, if two streams can not be scheduled simultaneously, the second stream has to wait until the first stream is completely finished before starting when considering the sequentially allocated strategy. However, when these streams are scheduled periodically, the second stream has to wait only one time slot before another slot comes available, and maybe a couple more to ensure that its periodicity does not match the periodicity of the first stream. In general, when the periodically allocated strategy is applied, a stream has a smaller waiting period before it can send its first packet.

Second, the periodically allocated strategy guarantees the periodicity of each stream. So, there might be a delay in the first packet (as discussed above) but for all the other packets a slot will be available as soon as the packet becomes available which is not the case with the sequentially allocated strategy.



**Fig. 7** Comparison of the jitter for the considered strategies and cases

Furthermore, Fig. 7 shows that the number of available channels also influences the obtained jitter. When applying the sequentially allocated strategy, the jitter is reduced by 0.3% to 4.3% (depending on the packet size) when increasing the available channels from 1 to 3. A further reduction of 26.4% to 26.6% is obtained when 13 channels become available. This is a total reduction of 26.9% to 29.5% compared to the case where only 1 channel is available. The main reason is that a higher number of available channels allows to plan the streams more simultaneously, the streams can be planned earlier, and time slots can be more guaranteed. Due to this a lower jitter is obtained. The same reason is true when allocating the packets periodically, although it is not visible in Fig. 7. Therefore, Table 2 shows the jitter for the three different cases and different packet sizes ranging from 34335  $\mu\text{s}$  to 48889  $\mu\text{s}$  for strategy I and from 118  $\mu\text{s}$  to 1100  $\mu\text{s}$  for strategy II. For completeness, we have also added a comparison with the sequentially allocated strategy. For the periodically allocated strategy, a reduction of 0.2% to 7.6% is found when moving from 1 channel to 3 channels and 0.1% to 46.2% when allowing 13 channels instead of 3 (i.e., a total reduction of 1.1% to 46.3%) as shown in Table 2.

Finally, the jitter increases when increasing the packet size (Fig. 7 & Table 2). For example, when the packets are periodically scheduled with 13 channels, a jitter of 118  $\mu\text{s}$  and 923  $\mu\text{s}$  is found when using packets of

Packet size	Sequentially			Periodically		
	Channels			Channels		
	1	3	13	1	3	13
100 bytes	46945 $\mu$ s	46803 $\mu$ s	34335 $\mu$ s	220 $\mu$ s	220 $\mu$ s	118 $\mu$ s
500 bytes	47467 $\mu$ s	46760 $\mu$ s	34344 $\mu$ s	633 $\mu$ s	624 $\mu$ s	516 $\mu$ s
1000 bytes	48119 $\mu$ s	46714 $\mu$ s	34388 $\mu$ s	855 $\mu$ s	847 $\mu$ s	846 $\mu$ s
1500 bytes	48889 $\mu$ s	46801 $\mu$ s	34451 $\mu$ s	1100 $\mu$ s	1017 $\mu$ s	923 $\mu$ s

**Table 2** Comparison of the jitter for the considered strategies and cases and a selection of packet sizes.

100 bytes and 15000 bytes, respectively (Table 2). This is due to the slight mismatch between the time that the packet becomes available and the time the packet is actually sent. Assuming that it takes 20  $\mu$ s to send some data and each slot has a duration of 6  $\mu$ s, in some case, the slot will be immediately available, while in other case the packet has to wait up to 6  $\mu$ s. The longer the slots become i.e., the larger the packet size, the larger this waiting period can be, and thus the higher the jitter.

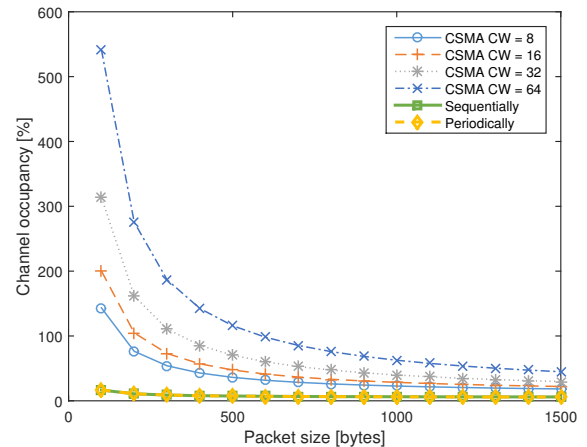
#### 4.2 Improvement of TDMA compared to classical CSMA/CA for WiFi 802.11n

The main aim of this section is to make a comparison between legacy CSMA/CA and the two TDMA strategies we propose. For a thorough comparison of the performance of the two TDMA strategies, we refer to Section 4.1. Note that the results for CSMA/CA are based on theoretical calculations and are not obtained by a simulator as been done for the TDMA strategies.

##### 4.2.1 Channel occupancy

The first considered parameter is the channel occupancy. Fig. 8 shows the channel occupancy for CSMA/CA and the two proposed strategies as a function of the packet size when only 1 channel is available. For the CSMA/CA case, four different content windows are considered: 8, 16, 32, and 64. Based on Fig. 8, we conclude that both the proposed TDMA strategies perform much better than CSMA/CA. The channel occupancy decreases with 67.2% to 96.9% (depending on the used contention window and packet size) when using a TDMA strategy. The same improvement is obtained by both TDMA strategies as they have the same performance when using only one channel as discussed in Section 4.1.2. The main reason for the better TDMA performance is due to the fact that there is no longer a need to send DIFSs and to account for a contention window. Once a packet is assigned to a certain time slot, it is guaranteed that the packet can be sent.

In Fig. 8, a channel occupancy of more than 100% is obtained for CSMA/CA in combination with certain



**Fig. 8** Comparison of the channel occupancy for CSMA/CA and the proposed strategies (1 channel available).

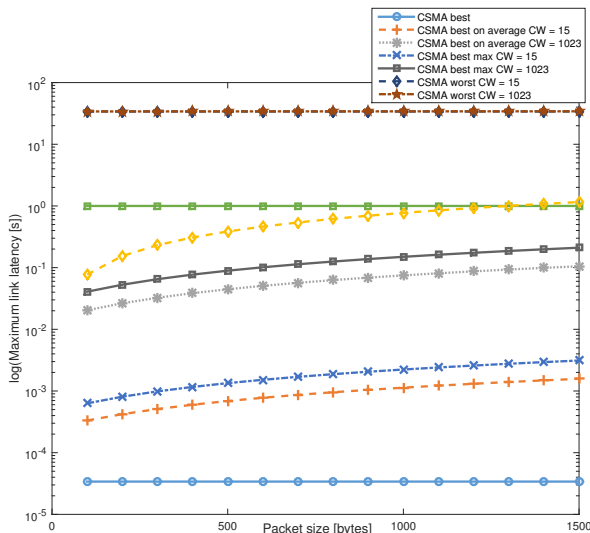
packet sizes. When the channel occupancy is higher than 100%, it is not possible to serve all the streams. For a contention window of 8, this means that the streams can only be served when using packet sizes from 200 bytes on. Similar for a contention window of 16, 32, and 64, a packet size equal or higher than 300, 400, and 500 bytes, respectively, is needed to serve all streams. For the proposed TDMA strategies, the channel occupancy is also lower than 100% and thus all streams can be served for all packet sizes. This is due the fact that no overhead is needed for the DIFS and the contention window as mentioned above.

Finally, Fig. 8 shows that a larger packet size (CSMA/CA) results in a lower channel occupancy, similar as for the TDMA strategies (see Section 4.1.2).

##### 4.2.2 Maximum link latency

Fig. 9 shows the maximum link latency for the considered CSMA/CA cases and the two TDMA strategies as a function of the packet size. A lower link latency is obtained by all the CSMA/CA best cases. For the best on average scenario, a maximum link latency between 332.5  $\mu$ s and 0.1 s (depending on the considered packet size and contention window) is obtained compared to latencies between 0.08 s and 1.2 s for our strategies. A 47.2% to 99.9% lower link latency is obtained for the CSMA/CA best case scenarios compared to our strategies. Note that such low link latencies are only possible when a node can transmit his data before all the other nodes, as mentioned above. In many cases, the medium will be occupied by another node and the node has to wait to send his data. In the worst case scenario, the node has to wait until each other node has send his packet as discussed above. A maximum link latency of

34.2 s is found when using a contention window of 1023 and a packet size of 1500 bytes. Compared to the worst case scenario for CSMA/CA, both our strategies perform significantly better (between 0.08 s and 1.2 s). Depending on the considered packet size and contention window, an improvement of 97.1% to 99.8% in link latency is found.

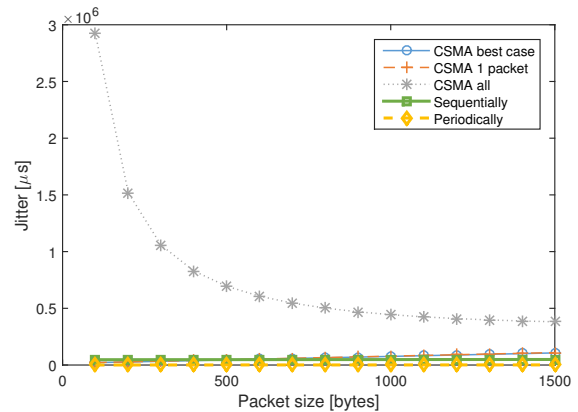


**Fig. 9** Comparison of the maximum link latency for CSMA/CA and the proposed strategies.

#### 4.2.3 Jitter

Fig. 7 compares the jitter for the three considered CSMA/CA cases and the two TDMA strategies as a function of the packet size when only 1 channel is available. The lowest jitter for TDMA (ranging from 220.1  $\mu$ s to 1100.4  $\mu$ s depending on the packet size, Fig. 10) is obtained when using the periodically allocated strategy. This is an improvement between 97.8% and 99.0% compared to the CSMA/CA best case and up to 99.9% compared to the CSMA/CA worst case, where all the other streams send one packet first. When applying the periodically allocated strategy, the difference in time between the moment the packet is ready to be sent and the moment that it is actually sent is very limited as a timeslot every  $x$  timeslots is guaranteed for each stream. In all CSMA/CA cases, one has to check if the medium is free and one has to back off in case it is occupied, thus larger time differences can occur. Furthermore, the chosen contention window also influences the moment that the packet can actually be sent.

The sequentially allocated strategy performs also better than the CSMA/CA worst case scenario (Fig. 10). A jitter reduction between 87.2% to 98.4% is obtained.



**Fig. 10** Comparison of the jitter for CSMA/CA and the proposed strategies (1 channel available).

This also due to the fact that for the CSMA/CA case, the packet can only be sent when the medium is available instead of having a guaranteed timeslot. A lower jitter is obtained for the two other CSMA/CA cases than for the sequentially allocated strategy when the packet size is below 600 bytes. From 600 bytes, the sequentially allocated strategy has the best performance with a jitter reduction up to 55.4%. We refer to Section 4.1.3 for the explanation of the better performance of the sequentially allocated compared to the periodically allocated approach.

## 5 Conclusion

In a world where everything will be connected, the limitations of today's wireless technologies will quickly come to light. Indeed, when multiple mobile devices with each their own requirements considering bit rate and latency are brought together in an environment such as an office or warehouse, it is no longer possible to serve all these devices simultaneously. We developed an intelligent TDMA scheduler which allows to plan the individual packets of the different streams in such a way that all streams can be served by taking into account the interference on the physical layer. Two different strategies are proposed, one where the packets are sequentially allocated and another one where the packets are periodically allocated. The scheduler is evaluated for a realistic industrial scenario with WiFi 802.11n as wireless technology and the results show that it is possible to serve all the required connections.

The strategies are evaluated based on the maximum link latency, channel occupancy, and jitter for different packet sizes. The periodically strategy performs the best for the maximum link latency for packet sizes below 1200 bytes and the jitter. The latter reduces with

97% to 99% compared to the sequentially strategy. The channel occupancy does not significantly differ between the different strategies. Furthermore, the number of available channels has a high influence on the channel occupancy and the jitter. Using 13 channels reduces the channel occupancy up to 30% and 46% for the sequentially and the periodically allocated strategy, respectively.

Compared to CSMA/CA, the proposed TDMA strategies reduce the channel occupancy significantly with an improvement between 67.2% to 96.9%. When using CSMA/CA, in some cases a channel occupancy of more than 100% is obtained which is not achievable in reality. We clearly showed that by using one of our TDMA strategies, it is still possible to plan all the considered streams for the industrial scenario, in contrary to CSMA/CA. For the maximum link latency, CSMA/CA performs 99% in the best case better than the proposed TDMA strategies, however, when considering the CSMA/CA worst case, in which all other nodes send one packet first, the proposed TDMA strategies perform 99% better than CSMA/CA. For the jitter, the periodically allocated strategy performs the best for all considered cases (98% and 99.9% lower than CSMA/CA best and worst case, respectively).

We recommend to use the periodically allocated strategy with 13 channels as best solution for the considered scenario.

The proposed scheduler has been evaluated in the 2.4 GHz band and hence compared with 802.11n solutions. Today, a majority of the Wi-Fi devices operate in the 2.4 GHz band and this band is becoming more and more crowded. As such, it is very hard to design dependable systems with more deterministic behavior when relying on CSMA/CA. In the future, we plan to assess the proposed mechanisms in the 5 GHz band and in light of the 802.11ac standard that supports beamforming. More channels are available in this band at the expense of a smaller effective range. Also, joint optimizations between 2.4 GHz and 5 GHz Wi-Fi bands might be considered. Furthermore, the performance of the scheduler will be experimentally validated and a comparison with PCF will be made. Finally, we will also further focus on the optimization of the scheduling opportunity in the spatial domain (i.e., the order in which the nodes are treated).

**Acknowledgements** Part of the work was supported by the iMinds IoT research program. M. Deruyck and E. Tanghe are Post-Doctoral Fellows of the FWO-V (Research Foundation - Flanders, Belgium).

## References

1. Cisco, *Visual Networking Index: Global Mobile Data Traffic Forecast 2015-2020*, 2015.
2. R. Drath, A. Horsch, *Industrie 4.0: Hit or Hype*, IEEE Industrial Electronics Magazine, Vol. 8, pp. 56-58, 2014.
3. X. Gong, J. Trogh, Q. Braet, E. Tanghe, P. Singh, D. Plets, J. Hoebeke, D. Deschrijver, T. Dhaene, L. Martens, W. Joseph, *Measurement-based wireless network planning, monitoring, and reconfiguration solution for robust radio communications in indoor factories*, IET Science, Measurement & Technology, 2016, doi: 10.1049/iet-smt.2015.0213.
4. Cisco and Rockwell Automation, *Wireless Design Considerations for Industrial Applications: Design and Deployment Guide*, pp. 5-6, March 2014.
5. M.R. Plattella, P. Thubert, X. Vilajosana, T. Watteyne, Q. Wang, T. Engels, *6TiSCH Wireless Industrial Networks: Determinism Meets IPv6*, Internet of Things: Challenges and Opportunities, Springer International Publishing, pp. 111-141, 2014.
6. S. Duquennoy, B. Al Nahas, O. Landsiedel, T. Watteyne, *Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH*, 13th ACM Conference on Embedded Networked Sensor Systems (SenSys), pp. 337-350, 2015.
7. M.R. Palattella, T. Watteyne, Q. Wang, K. Muraoka, N. Accettura, D. Dujovne, L.A. Grieco, T. Engel, *On-the-Fly Bandwidth Reservation for 6TiSCH Wireless Industrial Networks*, IEEE Sensors Journal, Vol. 16, No. 2, pp. 550-560, 2016.
8. M. Domingo-Prieto, T. Chang, X. Vilajosana, T. Watteyne, *Distributed PID-based Scheduling for 6TiSCH Networks*, IEEE Communications Letters, No. 99, pp. 6-9, March 2016.
9. D. Panigrahi, B. Raman, *TDMA Scheduling in Long-Distance WiFi Networks*, IEEE INFOCOM, pp. 2931-2935, 2009.
10. A. Dhekne, N. Uchat, B. Raman, *Implementation and Evaluation of a TDMA MAC for WiFi-based Rural Mesh Networks*, IEEE Transactions on Mobile Computing, Vol. 13, No. 2, pp. 392-406, 2014.
11. IEEE Computer Society, *IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - IEEE Std 802.11-2012*, 2012.
12. R. Costa, P. Portugal, F. Vasques, C. Montez, R. Moraes, *Limitations of the IEEE 802.11 DCF, PCF, EDCA, and HCCA to handle real-time traffic*, 13th IEEE International Conference on Industrial Informatics (INDIN), pp. 931-936, 2015.
13. G. Cena, L. Seno, A. Valenzo, C. Zunino, *On the Performance of IEEE 802.11e Wireless Infrastructures for Soft-Real-Time Industrial Applications*, IEEE Transactions on Industrial Informatics, Vol. 6, No. 3, pp. 425-437, 2010.
14. Y.-H. Wei, Q. Leng, S. Han, A.K. Mok, W. Zhang, M. Tomizuka, *RT-WiFi: Real-Time High-Speed Communication Protocol for Wireless Cyber-Physical Control Applications*, 34th IEEE Real-Time Systems Symposium (RTSS), pp. 140-149, 2013.
15. H. Trsek, T. Tack, O. Givehchi, J. Jasperneite, E. Nett, *Towards an Isochronous Wireless Communication System for Industrial Automation*, IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA), pp. 1-4, 2013.
16. Z. Yang, J. Zhang, K. Tan, Q. Zhang, Y. Zhang, *Enabling TDMA for Today's Wireless LANs*, IEEE Conference on

- Computer Communications (INFOCOM), pp. 1436-1444, 2015.
17. Q. Leng, Y.-H. Wei, S. Han, A. K. Mok, W. Zhang, M. Tomizuka, *Improving Control Performance by Minimizing Jitter in RT-WiFi Networks*, IEEE Real-Time Systems Symposium, pp. 63-73, 2014.
  18. G. Patti, G. Alderisi, L. Lo Bello, *SchedWiFi: An Innovative Approach to support Scheduled Traffic in Ad-hoc Industrial IEEE 802.11 networks*, 20th Conference on Emerging Technologies & Factory Automation (ETFA), pp. 1-9, 2015.
  19. E. O'Dowd, *Breaking Down the Evolution of 802.11 Wireless Standard*, 2015, Online: <http://solutionsreview.com/wireless-network/breaking-down-the-evolution-of-802-11-wireless-standard>, Accessed: 2016-07-01.
  20. M. Deruyck, E. Tanghe, W. Joseph, W. Vereecken, M. Pickavet, L. Martens, B. Dhoedt, *Model for power consumption of wireless access networks*, IET Science, Measurement & Technology, Vol. 5, Iss. 4, pp. 155-161, July 2011.
  21. K. Pister, P. Thubert, S. Dwars, T. Phinney, *RFC 5673: Industrial Routing Requirements in Low-Power and Lossy Networks*, The Internet Engineering Task Force (IETF), 2009.
  22. V. Cagri Güngör, G. P. Hancke, *Industrial Wireless Sensor Networks: Applications, Protocols, and Standards*, ISBN: 978-1-4665-0051-8.
  23. V. Angelakis, S. Papadakis, V. Siris, A. Traganitis, *Adjacent Channel Interference in 802.11a: Modeling and Testbed Validation*, IEEE Radio and Wireless Symposium, pp. 591-594, 2008.
  24. Cisco, *Channel Planning Best Practices*, Online: [https://documentation.meraki.com/MR/WiFi\\_Basics\\_and\\_Best\\_Practices/Channel\\_Planning\\_Best\\_Practices](https://documentation.meraki.com/MR/WiFi_Basics_and_Best_Practices/Channel_Planning_Best_Practices), Accessed: 2016-06-16.
  25. B.-J. J. Kim, K. K. Leung, *Frequency assignment for multi-cell IEEE 802.11 wireless networks*, US 7206586 B2, 2007.
  26. C. T. Lea, *Scalable wireless mesh networks*, US 8761125 B2, 2014.
  27. E. Tanghe, W. Joseph, L. Verloock, L. Martens, H. Capoen, K. Van Herwegen, W. Vantomme, *The Industrial Indoor Channel: Large-Scale and Temporal Fading at 900, 2400, and 5200 MHz*, IEEE Transactions on Wireless Communications, Vol. 7, No. 7, pp. 2740-2751, 2008.
  28. D. Plets, W. Joseph, K. Vanhecke, E. Tanghe, L. Martens, *Simple Indoor Path Loss Prediction Algorithm and Validation in Living Lab Setting*, Wireless Personal Communications, Vol. 68, No. 3, pp. 535-552, 2013.
  29. M. Deruyck, E. Tanghe, D. Plets, L. Martens, W. Joseph, *Optimizing LTE Wireless Access Networks towards Power Consumption and Electromagnetic Exposure of Human Beings*, Elsevier Computer Networks, Vol. 53, No. 11, pp. 29-40, 2015.