Fundamental Constraints for Time-slotted MAC Design in Wireless High Performance: the Realistic Perspective of Timing

Xiaolin Jiang[1], Zhibo Pang[2], Roger N. Jansson[2], Fei Pan[3], and Carlo Fischione[1]

[1]Network and System Engineering, KTH Royal Institute of Technology, Stockholm, Sweden
{*xiaolinj,carlofi*}*@kth.se*
[2]Automation Solutions, ABB Corporate Research, Västerås, Sweden
{*pang.zhibo,roger.n.jansson*}*@se.abb.com*
[3]National Key Lab of Commun., University of Elec. Science and Tech. of China, Chengdu, China
panfeivivi@hotmail.com

***Abstract*—Industrial applications pose the most stringent requirements to the underlying communication networks. Wireless industrial communication is gaining popularity due to its cost-effectiveness, flexibility and capability of functioning in harsh environments. However, to replace the wired counterpart in the most critical applications performing real-time control and monitoring, the according requirements in terms of latency, reliability and determinism must be met, the latency among which is found to be the bottleneck. To improve the latency performance of wireless communication, modification or new techniques should be developed. Moreover, medium access control (MAC) design should be based on valid timing parameters, as to achieve ultra-low latency, the timing parameters are pushed to the limits. In this paper, we consider to provide fundamental timing constraints as valid inputs for MAC design for wireless high performance network. We start from determining the fundamental constraints as well as the affecting factor in the timing perspective, and then we review and analyze some state-of-art wireless implementations in terms of the timing indexes. Based on the investigation and analysis, realistic timing constraints of different algorithms, hardware, mechanisms are presented, and three main directions for MAC design in wireless high performance network are outlined.**

***Index Terms*—industrial wireless communication, critical industrial control application, MAC design, timing constraints**

I. Introduction

There has been growing interest to employ wireless communication in industrial applications due to its low cost, mobility support, and capability in the harsh environments [1]. The available wireless communication protocols fail to satisfy the requirements posed by the most critical industrial applications, and the bottleneck is in the time domain as the cycle time of the available protocols can not meet the corresponding latency requirements. Aiming for the most critical control applications including factory automation, power systems automation, and power electronics control, the design of wireless high performance (WirelessHP) network targets Gbps data rates, 10^{-9} packet error rate and 10-µs-level cycle time [2]. WirelessHP is a protocol solution that so far has targeted mostly the physical layer (PHY) [3]. The aim is to extend it to a complete communication stack with multiple transmitters, hence the need of a medium access control (MAC) coordinating these transmissions.

Zhibo Pang's work was partly funded by Vinnova (Swedish Innovation Agency) through the grant number 2015-06548 and 2017-02822.

To provide the determinism and certain latency requirements required by the industrial applications, there have been some wireless solutions proposed with experimental evaluation, which keep PHY of IEEE 802.11 and change the MAC into TDMA [4]–[6]. Specific techniques are proposed to reduce different latency components, for example, a flexible PHY packet structure based on the IEEE 802.11 OFDM to minimize the transmission time of short packets for WirelessHP [3], and the synchronization accuracy improvement through the implementation modification of IEEE 1588 [7]. The processing delay of transmitter and receiver are investigated in [8], [9]. However, as these solutions are either designed to satisfy given latency requirements or lack complete protocol structure, they all fail to provide an overall latency limit.

There are many researches focusing on upper layers or one specific aspect inside the upper layer. Some theoretical researches at the MAC layer have not been validated via experimental tests, and only take some parameters abstracted from the PHY layer [10], [11], which may be insufficient in practice. In these works, it is a common practice to consider the scheduling unit in the time domain as a scalable value, which is a correct assumption only with rather relaxed latency requirements. However, considering the demanding scheduling of the order of microseconds or nanoseconds targeted by WirelessHP, the scheduling unit is actually restricted by the physical and technical limit, which ultimately bounds the lower limit for the scheduling units. In this case, the lack of the practical assumptions of the related interface parameters may prohibit the applicability or even correctness of the research at the MAC layer. Thus we need to determine the fundamental constraints that limit the downward scalability of the scheduling slots in the timing perspective. Unfortunately, no study can be found on these fundamental constraints.

The purpose of this paper is to systematically review the fundamental constraints that characterize time-slotted MAC. Such a review is the essential step for the future design of WirelessHP MAC, which is our long term ambition. With such an aim in mind, we first determine the fundamental constraints from the timing perspective, and in Section II we analyze the factors having effect on these constrains. In Section III, we review the state-of-art wireless communication implementations (that include time-slotted MAC) to support low-latency requirements, and compare the corresponding parameters we derived from Section II. In Section IV, the realistic timing assumptions based on different selections of algorithms, hardware, PHY structures are derived, and three future directions for WirelessHP MAC design are discussed. Finally, Section VI concludes this paper.

II. Key Factors in the Fundamental Time Constraints

In this section, we determine the fundamental constraints for the duration of the time slots in time-slotted MAC. The factors that affect the constraints are also analyzed.

A. Constraints

1) Service time: First we determine the involved delay components inside a one-way service time with automatic acknowledgement (ACK), which is from the start of the packet transmission until the ACK is received.

$$T_{\text{ser}} \geq T_{\text{guard}} + T_{\text{packet}} + T_{\text{proc}} + T_{\text{ACK}} + T_{\text{prop}}, \qquad (1)$$

where T_{guard} is guard time to tolerate slight desynchronization between the transmitter and receiver, T_{packet} is the packet transmission time, T_{proc} is the processing delay required for a wireless interface to process a received packet and to respond, T_{ACK} is the ACK transmission time, and T_{prop} is the round trip propagation delay. Retransmission time is not included in the service time, as retransmission is considered to be scheduled in a new service time if needed.

2) Synchronization accuracy: T_{guard}, which is lower bounded by the clock offset between transmitter and receiver, is used to avoid missing packets due to misaligned clocks between the transmitter and receiver. As clock drift always exists, the clock offset increases with time. To avoid setting T_{guard} to a very large value, the clocks should be re-synchronized. The synchronization protocols can be categorized into one-way protocols and two-way protocols depending on the transmission direction of the synchronization messages. One-way protocols include Global Positioning System (GPS) and IEEE 802.11 Timing Synchronization Function (TSF) [12]. The synchronization accuracy with GPS can be achieved within 40 ns. The main drawbacks of GPS system are security risk (the GPS signals can easily be jammed) and indoor devices may require long feeder cables. IEEE 802.11 TSF synchronize the nodes to a local clock source, which sends beacons with time-stamp periodically. The clock rate is 1 MHz clock with resolution of 1 μs , and the synchronization accuracy is within $\pm 0.01\%$. On a commercial level, industry vendors assume the 802.11 TSF's synchronization to be within 25 μs .

IEEE 1588 [13], which is also known as Precision Time Protocol (PTP), is the representative of two-way protocols. Exchanged synchronization messages with time-stamps are used to calculate the transmission delay and the clock offset. IEEE 1588-2008 can achieve ns-level accuracy. When digging a little deeper, besides the lower bound by the clock resolution, the synchronization accuracy of IEEE 1588 is also affected by the jitter of the time-stamp transmission delay.

Over all, the better accuracy provided by the synchronization algorithm, the shorter the synchronization interval, the smaller the clock offset and T_{guard}.

3) Processing delay: After the receiver received the packet, T_{proc} should be accounted before sending ACK back to the transmitter, which gives the receiver time to get the data from the PHY, verify it, and generate the ACK. T_{proc} includes packet processing delay at the receiver (at PHY and MAC), and radio frontend switching time to change circuitry from receiving to transmitting. Paper [9] presents the average sub-frame computational time of the major receiver modules of LTE Release 8 system using an Intel Core i5 computer. Channel estimation, decoding, and carrier frequency offset consume the most computational time. The presented latency numerical results indicate the relative latency of different processing modules, though the actual latency may vary with different devices, algorithms and parameters.

4) ACK Transmission: TDMA is considered suitable to support deterministic wireless communication, where the time resource is divided into slots. ACK transmission mechanism determines which components in the service time, as discussed in Section II-A1, are included inside a slot duration. There are generally two ACK transmission mechanisms. The first one is in-slot ACK transmission, where the Node A immediately sends ACK to the received packets from Node B. In this mechanism, the slot duration is the same as service time and include all the components in the service time.

$$T_{\text{slot}} \geq T_{\text{guard}} + T_{\text{packet}} + T_{\text{proc}} + T_{\text{ACK}} + T_{\text{prop}}. \qquad (2)$$

In the second mechanism for Node A is to piggyback the ACK in its next packet send to Node B (T_{ACK} is saved and the ACK information is piggybacked in the next T_{packet}). In this mechanism, the slot duration is bounded by the sum of T_{guard} and T_{packet}, as T_{proc} can be performed during the interval before next scheduled transmitting slot, and T_{prop} can also be removed with no risk of collision caused by the ACK.

$$T_{\text{slot}} \geq T_{\text{guard}} + T_{\text{packet}}. \qquad (3)$$

Piggybacking ACK transmission can achieve a shorter slot duration compared to in-slot ACK transmission. The disadvantage of piggybacking ACK transmission is that the retransmission delay may be larger compared to in-slot ACK transmission, as ACK is received later in the piggybacking ACK mechanism.

B. Factors affecting the time constraints

1) Hardware: Hardware affects T_{guard}, T_{proc}, T_{packet} in the service time. First, as discussed in Section II-A2, where to strike times-tamps affects the synchronization accuracy that further affects T_{guard}. It is best to strike and capture the time-stamps by the hardware upon passage of a designated symbol in the media interface, as it is close to the physical media and it gets rid of the inaccuracy caused by interrupts and other applications running in the processor. In IEEE 1588-2008, time-stamps are added with hardware assistance between MAC layer and PHY layer [13].

Second, the radio frontend switching time and the packet processing assigned to hardware also limit the T_{proc}. For devices based on system on chip (SoC), the processing delay depends on the capability of CPU. While for field-programmable gate array (FPGA), the processing speed is affected by the parallelization level of the implementation, which is a tradeoff between the complexity and the minimum processing sample time of FPGA. Higher parallelization enables faster processing speed, but also cause higher utilization of the resource and higher expense.

Third, the maximum transmission rate of the hardware affects the minimum achievable T_{packet} given fixed packet length.

2) Software: Software affects T_{guard} and T_{proc} in the slot length. First, when performing time stamp in the software stack, it will results in a worse synchronization accuracy compared to hardware time stamp. The software-only implementation timestamps packets in the application layer that is furthest away from the physical layer, which yields the largest amount of delay and jitter passing the time-stamp through the various layers. Software time-stamps typically give errors on the order of microseconds to milliseconds depending on the operating system and platform.

Secondly, the processing delay component T_{proc} is also affected by the software processing speed. To support the most time-sensitive applications with evolving protocols, heterogenous computing architecture is introduced, the signal processing of which is performed by a software processor and reconfigurable hardware. The key to achieve real-time performance is to determine the best processing components allocation to the hardware and software respectively. A hardware-software codesign for wireless transceiver is investigated based on Zynq based heterogeneous platform [8]. As processing speed of hardware is faster than that of software, software step time is set as the time to process per frame, while that for hardware is set as the time to process per sample (there are multiple processing samples per frame). In the timing perspective, to achieve the minimum latency is to find the best tradeoff between the step times of hardware and software by determining the best mapping of each component to either processor software or configurable hardware. From the results of IEEE 802.11a implementation in [8], IFFT in the transmitter and preamble detection, FFT and Viterbi decoding in the receiver are more suitable to be implemented by the hardware. The best mapping achieves near minimum latency without too much hardware resource utilization.

3) OS: The operating system (OS) is another affecting factor of T_{proc} component. General-purpose OS like Linux focuses on computing throughput, and uses a time-sharing architecture between each task. Linux support real-time scheduling option, which can guarantee at best "soft" real-time (real-time performance is achieved for most of the time). A real-time operating system (RTOS) is an operating system designed to provide scheduling guarantees to ensure deterministic behaviour and timely response events and interrupts. The scheduler in a RTOS is designed to provide a predictable and deterministic execution pattern. This is particularly of interest to embedded systems as embedded systems often have real time requirements. A RTOS should support programmer to set priorities to an overall application and for different tasks within an application, where the highest priority task can immediately preempt any lower priority tasks without completion of a time-slice. The RTOS needs much less CPU resources compared to the general-purpose OS such as Linux, which enables it to run on a microcontroller suitable for embedded systems.

Many RTOS are not full OS, but scheduling kernel only. Additional functionality, such as a command console interface, or networking stacks, can then be included with add-on components. FreeRTOS is a real time kernel that provides the core real time scheduling functionality, inter-task communication, timing and synchronisation primitives only. FreeRTOS achieve determinism by allowing the user to assign a priority to each task. The scheduler then uses the priority to know which task to run next.

4) PHY: T_{packet} is determined by the PHY used, payload length, and available bandwidth. Higher coding rate and modulation order result in shorter packet, and vise versa. Currently, there is a general interest at investigating if short transmission time interval (T_{packet}) is practical. For example, "transmission time interval shortening" can lower the transmission latency is one of the key enabling features introduced in 5G Ultra-Reliable Low Latency Communication (URLLC) use case. It is believed that a short transmission time interval of 2 OFDM symbols can potentially enable the required low latency in critical Internet of Things applications [14].

By taking the advantage of deterministic and periodic traffic in the latency-sensitive industrial applications, a short PHY packet structure based on OFDM is proposed which consisted of one preamble symbol and several data symbols [3]. The optimal number of data symbols and OFDM parameters to minimize the T_{packet} are customized on the payload size.

III. State-of-art Implementations

In this section, we review some state-of-art wireless communication implementations to support low-latency applications, and compare their performance corresponding to the constraints and factors derived from the previous section and summarize the results in Tab. I.

A. SoC-based

A SoC is a chip that integrates all components of an electronic system including a microprocessor, analog interfaces and advanced peripherals like GPU and Wi-Fi module[1].

[1]The "SoC" here refers to the system with predefined internal elements, which is different from Programmable SoC with field-programmable gate array (FPGA) or a complex programmable logic device (CPLD).

TABLE I
STATE-OF-ART COMPARISON.

	RT-WiFi [4]	**WIA-FA** [5]	**EchoRing** [6]	**1588-2008 HW Impl** [7]	**WirelessHP PHY** [3]
Hardware	Atheros AR9285 Wi-Fi chip	Net5501 board	WARP	PicoZed SDR	USRP N210
Software	mac80211 and Ath9k	Ath5k	MicroBlaze processor	ARM Core	n/a
OS	Ubuntu 12.04	Linux	n/a	FreeRTOS	n/a
Sync Accuracy	20 µs	7 µs (hard)	n/a	75 ns (hard)	n/a
PHY	802.11 OFDM 54 Mbps	802.11a/g	similar to 802.11a (6 Mbps, 3 Mbps)	802.11a/g	WirelessHP PHY
T_{packet}	60 µs (200 B)	n/a	197.3 µs (100 B)	n/a	20.4 µs (104 bits)
T_{ACK}	28 µs (14 B, 20 Mbps)	n/a	64 µs (24B)	n/a	n/a
T_{proc}	10 µs	n/a	n/a	n/a	n/a
in-slot ACK	Yes	No	Yes	n/a	n/a
Slot length	118 µs	100 µs	>681.3 µs	n/a	n/a

1) RT-WiFi: RT-WiFi is a TDMA data link layer protocol modified based on IEEE 802.11 to provide deterministic timing guarantee on packet delivery and high sampling rate [4]. IEEE 802.11 TSF is used for synchronization, where nodes keep a local 1 MHz TSF timer and synchronize to the master clock through the periodical beacons sent by the master clock. 20 µs T_{guard} can be achieved with 100 ms beacon interval and ± 100 ppm synchronization accuracy. In-slot ACK is considered, thus T_{proc} should be considered in the slot duration. RT-WiFi uses Atheros AR9285 Wi-Fi chip, which supports IEEE 802.11 b/g/n MAC and baseband processing at 2.4 GHz frequency band with maximum transmission rate being 150 Mbps. The TDMA MAC layer design of RT-WiFi is build on Ubuntu 12.04 OS and two Linux software modules. The mac80211 module is a framework for developing software-based MAC driver, and hardware deriver module ath9k is for supporting AR9285. As Ubuntu 12.04 is not a hard RTOS, to achieve deterministic slot start, the execution of the interrupt at the beginning of a time slot is changed from the kernel to the interrupt routine (by avoiding it being blocked by other tasklets with the same priority). Moreover, separate queues for each real-time communication links are kept at the message transmission module to reduce queuing delay. T_{proc} in RT-WiFi is set to 10 µs according to IEEE 802.11g short interframe space (SIFS) duration. The physical layer uses 802.11 OFDM modulation with data rate 54 Mbps. T_{packet} is 60 µs consisted of 20 µs for physical layer preamble and header delay, and 40 µs with 200 byte payload together with 64 byte upper layer headers and frame check sequence. T_{ACK} is 28 µs with 14 byte frame size. By summing up the according components, the minimum slot duration is 118 µs.

2) WIA-FA: Wireless Networks for Industrial Automation - Factory Automation (WIA-FA) is the communication protocol to support hard real-time communication requirements of discrete manufacturing factory automation applications [15]. A two-way time synchronization algorithm based on TDMA mechanism is proposed for WIA-FA [5], which can achieve 7 µs synchronization accuracy while causing moderate synchronization overhead. The TDMA MAC of WIA-FA is modified based on IEEE 802.11 standard. WIA-FA banned in-slot ACK to reduce the slot duration (by removing T_{proc} and T_{ACK}) and to avoid performance degradation caused by ACK collision. The experiments performed in [5] uses Atheros Net5501 board (RF chip is Atheros AR5212) and Debian Lennny OS with Linux 2.6.26 kernel. Atheros has IEEE 802.11 TSF timer with 1 µs resolution and high precision timestamping at hardware. The Atheros RF chip driver Ath5k is modified for TDMA operation. As Atheros can not ban the RF chip to perform physical carriere sensing, in WIA-FA, an extreme value is set for the clear channel assessment carrier sensing threshold. To achieve better determinism, the timer at Linux core (with 1 ns resolution) is used to trigger the interrupt for dividing time slots. The PHY is based on IEEE 802.11-2007, the time-slot is 100 µs without more details for individual latency components.

B. FPGA-based

FPGAs contain an array of programmable logic blocks, which can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR.

1) EchoRing: is wireless version data link layer protocol of Token-passing, where medium access is granted when a station receives the token, and it can hold the token for Token Holding Time (THT) [6]. In this way it is also TDMA and can guarantee determinism. In-slot ACK and fixed number of retransmission is scheduled inside one slot. By exploiting the frequently exchanged token, the nodes in the network proactively acquire the channel quality information, based on which each node select a cooperation station to retransmit (echo) a packet in case of a missing ACK. For the applications that have strict latency requirements, it is encouraged to exploit diversity in other domains instead of time domain, for example, spatial diversity is harvested by the echo from cooperation station. EchoRing is based on Wireless Open-Access Research Platform (WARP) platform [16], which offers a PHY similar to IEEE 802.11a, and CSMA MAC. PHY processing of WARP is divided across multiple cores and the MAC is primarily implemented in software running in two MicroBlaze CPUs, with a support core in the FPGA to achieve accurate inter-packet timing. EchoRing implements its MAC

TABLE II
REALISTIC ASSUMPTIONS FOR MAC DESIGN.

Category	Parameter	Implementation Options	Realistic Assumption
Synchronization	T_{guard}	GPS IEEE 802.11 TSF (soft) Two-way Sync (hard) IEEE 1588 (hard)	40 ns 20 µs several µs several tens ns
PHY	T_{packet}	IEEE 802.11 WirelessHP	>20 µs several µs
	T_{ACK}	In-slot ACK Piggybacked ACK	long (28 µs in 802.11) 0
	T_{slot}	IEEE 802.11 (TSF, In-slot ACK) + SoC-based device WirelessHP + Hard time-stamp Sync + Piggybacked ACK + FPGA-based device	100 µs 10 µs

by modifying the original MAC at software level. BPSK and QPSK are used to modulate headers and payload with 1/2 coding rate , the corresponding transmission rate of which are 3 Mbps and 6 Mbps respectively [6]. With 24 B packet headers, Token and ACK, and 100 B payload size, the THT (one token and payload packet retransmission respectively) is as long as 681.3 µs by the calculation in [17]. Note that the slot duration should be larger than the THT value as the THT value does not include T_{guard} or T_{prop}, which are not given from the experimental parameters [6].

2) IEEE 1588-2008 Hardware-based Timestamp: An IEEE 1588-2008 hardware-based timestamp implementation embedded in 802.11 a/g is presented in [7]. The implementation uses PicoZed SDR platform, which contains a Zynq FPGA with Programmable Logic and an ARM double core microcontroller. Programmable Logic contains the IEEE 1588-2008 processes with high timing requirements, such as timestamping, while the rest of the protocol are been programmed in the ARM core that runs on the RTOS kernal FreeRTOS. The clock runs at 20 MHz frequency, so its timestamp resolution is 50 ns, and the synchronization accuracy is characterized by a standard deviation below 20 ns and a maximum error of 75 ns.

3) WirelessHP PHY Design: To reduce the communication latency, the PHY of WirelessHP is designed to minimize packet transmission time by cutting the preamble symbol number and optimizing OFDM parameters according to given payload length. From the validated experimental results, 20.4 µs T_{packet} can be achieved with 8PSK and 5 MHz bandwidth for 104 bit payload. However, when taking the same modulation and coding with RT-WiFi, it is expected to achieve 6.4 µs T_{packet}. The USRP N210 platforms employ a Xilinx Spartan 3A-DSP FPGA module that performs digital/analog conversion and sample buffering, but cannot perform the baseband processing. We do not compare its processing performance, as the hardware processing capability is rather low, which is not suitable for low latency communication.

IV. DIRECTIONS TOWARDS WIRELESSHP MAC DESIGN

A. Realistic Assumption for MAC Design

Based on the above investigation and analysis, we summarize the realistic timing constraints in terms of the constraints and the affecting parameters in Tab. II. The synchronization accuracy that limits the T_{guard} component ranges from several tens ns to several tens µs by different synchronization algorithm and implementation. For the PHY, as IEEE 802.11 has fixed packet structure, the preamble duration of which is $20\,\mu s$, T_{packet} as well as T_{ACK} can not be shorter than $20\,\mu s$. Customized packet structure like WirelessHP can achieve lower T_{packet}, which is rather scalable with the payload length. Piggybacked ACK will not introduce latency as explained in Section II-A4. T_{proc} is not numerically presented, as it depends on the hardware capability and the complexity of the implementation. Generally, FPGA-based device is considered to have stronger processing capability and achieve better latency performance compared to SoC-based device. Lastly, T_{slot} can be derived by summing up the corresponding delay components. Characterized by IEEE 802.11 with TSF implemented at software layer, in-slot ACK, fixed packet structre and SoC-based hardware, 100 µs is considered as the realistic assumption. While by combining more advanced/timing-efficient designs with flexible WirelessHP packet structure, hardware timestamping, piggybacked ACK and FPGA-based hardware, 10 µs T_{slot} can be expected. The results Tab. II can be taken as reliable input parameter for MAC design.

B. New PHY

From Tab. I, the T_{packet} components of RT-WiFi and EchoRing accounts for a large portion of the slot duration. The listed representative protocols all adopt IEEE 802.11 OFDM PHY for its support for high data rates and available implementation. IEEE 802.11 OFDM PHY has long preamble and the FFT size is fixed. For example, in RT-WiFi, 20 µs out of 60 µs is consumed by the preamble with 200 byte payload. However, for short payload size (around 100-200 bit), the preamble duration of the packets will become longer than the payload duration, which is not latency efficient. Moreover, the fixed FFT size prohibits to achieve the optimized packet transmission time, as when the payload size is not integer times of the payload subcarrier number when performing IFFT transformation, padded zero bits are used to fill the blank, which causes resource waste both in the frequency and time domains. For the latency-sensitive applications, the preamble structure should be shortened or customized, cutting off the overhead to support generality or backward/forward compatibility. The data symbol size should be flexible and be able adjust to the actual payload size. Customized PHY that can optimize parameters based on payload and transmission time requirement is encouraged, en example practice is in [3].

C. Hardware Timestamping in Existing Standards

Regardless of the ACK mechanism, T_{guard} component is always needed. From the Sync Accuracy comparison in Tab. I, the T_{guard} value varies greatly among different synchronization implementations. Hardware timestamping achieves better synchronization accuracy than software synchronization, as the hardware is nearer to the physical media and less jitter is introduced. The previous wireless communication protocols are designed for latency-relaxed scenarios, which have relaxed requirements for T_{guard}. When modifying available protocols to satisfy strict latency requirements, an effective way to reduce T_{guard} is to move the timestamping from software to hardware. If higher synchronization precision is required, more accurate algorithm like IEEE 1588-2008 should be applied.

D. Hardware-software Codesign

The processing latency is related with both hardware and software. Though the hardware processing speed is many time faster than the software, it is not the best practice to allocate all the processing tasks to the hardware, which may cause too much utilizaton of the hardware resource. A balanced and near-optimal mapping is one that neither the hardware processing time or the software processing time acts as a bottleneck to further decrease the overall processing delay for the chosen processing unit (for example a data frame). This can be done by experimental evaluation of different mappings as in [8], which can achieve the most accurate results but takes effort to disign both implementation of each processing module in both hardware and software. Another way is to determine the processing flow, and estimate the processing complexity for each processing modulel. The processing modules, which has less complexity and the processing result of which is called fewer by other modules, should be mapped to the software. This method avoids the implementation of each module in both hardware and softwawre, however it does not guarantee to obtain the optimal mapping. People can choose between the two method or combine the two methods by taking the device resource, the availability of differnt modules and the difficulty to implement the modules into consideration.

V. Conclusion

This paper investigates the fundamental constraints in the timing perspective which can be used for the MAC design in Wireless High Performance communication network. Synchronization accuracy, packet transmission time, acknowledgement mechanism, and processing delay are considered as the main constraints. The realistic timing assumptions are derived by analysing the affecting factors of each constraints and the state-of-art implementations. From the summarised results, we argue that new physical layer structure, hardware timestamping in existing standards, and hardware-software codesign need further investigation to reduce the latency.

Acknowledgement

We thank Michele Luvisotto from ABB Corporate Research, Västerås, Sweden, who shared insight and expertise during the discussion, and provided comments that greatly improved the manuscript.

References

[1] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[2] M. Luvisotto, Z. Pang, and D. Dzung, "Ultra high performance wireless control for critical applications: Challenges and directions," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1448–1459, 2017.

[3] M. Luvisotto, Z. Pang, D. Dzung, M. Zhan, and X. Jiang, "Physical Layer Design of High-Performance Wireless Transmission for Critical Control Applications," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 2844–2854, 2017.

[4] Y.-H. Wei, Q. Leng, S. Han, A. K. Mok, W. Zhang, and M. Tomizuka, "RT-WiFi: Real-time high-speed communication protocol for wireless cyber-physical control applications," in *Real-Time Systems Symposium (RTSS), 2013 IEEE 34th*. IEEE, 2013, pp. 140–149.

[5] Y. Yutuo, L. Wei, Z. Xiaoling, and L. Shuai, "Time Synchronization Method of Wireless Network for Factory Automation," *Journal of Computer Research and Development*, vol. 51, no. 3, pp. 511–518, 2014.

[6] C. Dombrowski and J. Gross, "EchoRing: a low-latency, reliable token-passing MAC protocol for wireless industrial networks," in *European Wireless 2015; 21th European Wireless Conference; Proceedings of*. VDE, 2015, pp. 1–8.

[7] O. Seijo, C. Cruces, R. Torrego, J. A. Lpez-Fernndez, and I. Val, "IEEE 1588 Hardware-Based Timestamp Implementation over 802.11a/g for IWSAN with High Precision Synchronization Requirements," in *2017 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*.

[8] B. Drozdenko, M. Zimmermann, T. Dao, K. Chowdhury, and M. Leeser, "Hardware-Software Codesign of Wireless Transceivers on Zynq Heterogeneous Systems," *IEEE Transactions on Emerging Topics in Computing*, 2017.

[9] H. Chen, R. Abbas, P. Cheng, M. Shirvanimoghaddam, W. Hardjawana, W. Bao, Y. Li, and B. Vucetic, "Ultra-reliable low latency cellular networks: Use cases, challenges and approaches," *arXiv preprint arXiv:1709.00560*, 2017.

[10] H. Yan, Y. Zhang, Z. Pang, and L. Da Xu, "Superframe planning and access latency of slotted MAC for industrial WSN in IoT environment," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1242–1251, 2014.

[11] V. N. Swamy, S. Suri, P. Rigge, M. Weiner, G. Ranade, A. Sahai, and B. Nikolić, "Cooperative communication for high-reliability low-latency wireless control," in *Communications (ICC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4380–4386.

[12] IEEE 802.11 Working Group, "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*, 2016.

[13] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)*, pp. 1–300, July 2008.

[14] P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, S. A. Ashraf, B. Almeroth, J. Voigt, I. Riedel *et al.*, "Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 70–78, 2017.

[15] IEC, PD, "PAS 62948: 2015 Industrial networks," *Wireless communication network and communication profiles. WIA-FA. BSI*, 2015.

[16] "WARP Project." [Online]. Available: http://warpproject.org

[17] C. Dombrowski, S. Junges, J.-P. Katoen, and J. Gross, "Model-Checking Assisted Protocol Design for Ultra-Reliable Low-Latency Wireless Networks," in *Reliable Distributed Systems (SRDS), 2016 IEEE 35th Symposium on*. IEEE, 2016, pp. 307–316.