

Implementation and Evaluation of the RBIS Protocol in 5G

Michael Gundall*, Julius Stegmann*, Christopher Huber*, Rüdiger Halfmann†, and Hans D. Schotten*‡

*German Research Center for Artificial Intelligence GmbH (DFKI), Kaiserslautern, Germany

†Nokia Bell Labs, Munich, Germany

‡Department of Electrical and Computer Engineering, Technische Universität Kaiserslautern, Kaiserslautern, Germany

Email: {michael.gundall, julius_raphael.stegmann, christopher.huber, hans_dieter.schotten}@dfki.de

{ruediger.halfmann}@nokia-bell-labs.com,

Abstract—5G that is strongly focused on improvements on QoS and QoS guarantees, which are necessary for industrial deployments, enables novel use cases. Here, nearly each use case requires time synchronization of the involved systems. While PTP in its variations, e.g. IEEE 1588 v2.1 or IEEE 802.1AS, has established as standard for wireline systems, time synchronization of wireless or hybrid systems is still subject to research.

Thus, the so-called RBIS protocol, which was originally developed and investigated for Wi-Fi, is mapped to 5G. This is possible, because both systems are infrastructure based and a suitable broadcast that fits to the requirements of RBIS protocol can be found in the control layer of 5G NR. Even if the $1 \mu\text{s}$ requirement that is required by some applications is not yet cracked, the accuracy of $1.3 \mu\text{s}$ and precision of $\leq 4.3 \mu\text{s}$ for non-invasive extension of existing 5G deployments is highly promising.

Index Terms—5G, RBIS, Clock Synchronization, Time Synchronization, IEEE 1588, IEEE 802.1AS

I. INTRODUCTION

5th Generation Wireless Communication Systems (5G) are the basis for future wireless applications. Thus, 5G systems recently enter factory halls to firstly solve stringent wireless industrial use cases [1]. The continuous integration of systems into the wireless landscape as well as the rising demands on quality of service (QoS) lead to more and more distributed systems. In contrary, dependant on specific applications, a dynamic system composition is feasible. Here, a high synchronization quality between the systems is urgently required. While clock synchronization is mostly solved for wireline systems, it is still under research for wireless or hybrid communication systems. As precise clock synchronization is already a challenge for wireless systems, hybrid systems demand the convergence of wired and wireless systems. This requires the integration of an existing time domain, e.g., a time-sensitive networking (TSN) network. While a transparent tunneling of Precision Time Protocol (PTP) messages via 5G User Plane Functions (UPFs) is possible, the performance is highly decreased due to both determinism of the air interface and the UPF scheduling in the core network.

This is a preprint of a work accepted but not yet published at the 2022 IEEE Globecom Workshops (GC Wkshps). Please cite as: M. Gundall, J. Stegmann, C. Huber, R. Halfmann, and H.D. Schotten: "Implementation and Evaluation of the RBIS Protocol in 5G". In: 2022 IEEE Globecom Workshops (GC Wkshps), IEEE, 2022.

Overall, the most demanding requirements regarding synchronicity are local devices that have to cooperate [2], whereas most challenging use cases require a clock synchronicity of $\leq 1 \mu\text{s}$ [3]. Furthermore, an increased synchronization between these devices not only leads to realization of novel use cases or improved product quality, but also to higher possible speed and product processing, e.g. by mobile robots. This leads to both increased efficiency and revenue.

Consequently, the following contributions can be found in this paper:

- **Concept for clock synchronization of 5G User Equipments (UEs) on the basis of Reference Broadcast Infrastructure Synchronization (RBIS) protocol.**
- **Evaluation of the proposed concept performing measurements on a testbed.**

Accordingly, the paper is structured as follows: Sec. II describes the principles and the state-of-the-art regarding clock synchronization, while Sec. III gives an overview of related work. Furthermore, Sec. IV proposes the mapping of the originally for Wi-Fi designed RBIS protocol to 5G. Furthermore, the testbed is introduced (Sec. V) and the concept is evaluated (Sec. VI). Finally, the paper is concluded (Sec. VII).

II. CLOCK SYNCHRONIZATION

A precise and accurate clock synchronization is the basis for a wide range of industrial use cases and applications. As of now, most industrial communication systems are wireline, due to several advantages, such as determinism, reliability, small and bounded latency, and more. Thus, lots of clock synchronization algorithms were developed specifically for this type of communication, whereby the most common ones can be grouped into one-way and two-way synchronization protocols [4]. While one-way synchronization protocols, such as Flooding Time Synchronization Protocol (FTSP), send only one message from the time master to one or more slaves, the quality is highly dependent on the latency and jitter of the Local Area Network (LAN) [5]. Therefore, two-way synchronization protocols, such as Network Time Protocol (NTP) were introduced [5]. Using NTP, a single message exchange following the ping-pong scheme and containing several timestamps can be used to reach a 10-100 ms accuracy in wireline

systems, depending strongly on the number of network devices located in the LAN [6]. While this accuracy is often sufficient for non-industrial end users, most operating systems of personal computers (PCs) are synchronized using this approach. However, for industrial environments, a higher accuracy is required. Hence, the PTP (IEEE 1588) was introduced [7]. As shown in Fig. 1, a total of four messages are sent. While *SYNC* and *DELAY REQ* are used as trigger for creating the timestamps, *FOLLOW UP* and *DELAY RESP* make the timestamps available for the slave [6]. During the complete process four different timestamps are produced. With the help of these timestamps, not only the offset between master and slave clock can be calculated (see Eq. 2), but also the delay of the packet transmission (see Eq. 1).

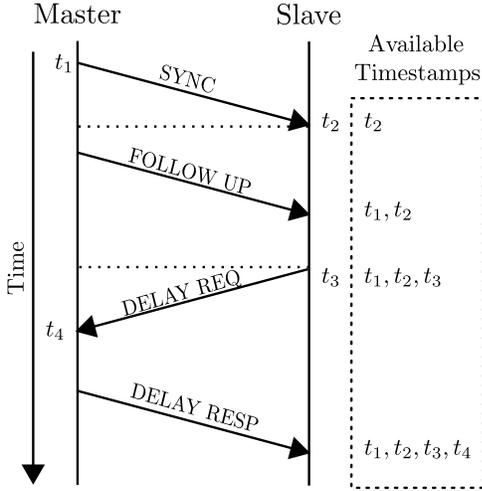


Fig. 1. Message exchange and corresponding timestamps in PTP (IEEE 1588).

$$Delay = \frac{(t_2 - t_1) + (t_4 - t_3)}{2} \quad (1)$$

$$Offset = \frac{(t_2 - t_1) - (t_4 - t_3)}{2} \quad (2)$$

Since the delay measurement and its correction highly improves the clock synchronization, it is widely used whenever a precise clock is required. Due to the fact that PTP is based on delay measurements, the best performance can be reached in local wireline networks, where the path of the packet and its delay is nearly static. Thus, in a LAN easily microsecond or even sub-microsecond accuracy could be reached, whereas the actual performance depends on the capabilities of the devices, i.e., master and slave providing hardware timestamping or the availability of network switches with PTP support. Hence, the recently developed TSN technology specifies a standard for time synchronization using PTP as basis [8]. This so-called IEEE 802.1AS standard extends PTP by a novel configuration. In addition, hardware timestamping capabilities and protocol stack message exchange directly at the MAC layer are required to comply with this standard. Using these features, nanosecond accuracy is possible. Since some use cases even require sub-nanosecond accuracy the IEEE 1588 v2.1 profile supports this

accuracy, but requires sophisticated devices and extensions [9].

While PTP and its deviations are well suited for clock synchronization in wireline systems, clock synchronization in wireless systems requires different approaches, since the introduced delay measurement is not as accurate as for wireline systems. This is because the propagation paths of the signals sent from mobile devices change frequently during operation. Thus, different concepts are required for the clock synchronization of devices using wireless communication links. Here, the so-called Reference Broadcast Synchronization (RBS) protocol, which belongs to the group of one-way synchronization protocols, was developed [10]. In the RBS protocol one node acts as master and sends a cyclic broadcast including the reference time to all slaves that have to be synchronized. In this way it makes use of the properties of the shared communication channel because it is assumed that the broadcast arrives at each of the slaves at nearly the same time. This assumption is possible, because the range of a Wi-Fi deployment is typically limited. Since the purpose of the RBS protocol is to synchronize infrastructure-free networks, but many novel wireless communication systems such as Wi-Fi or 5G are infrastructure-based, a separate version, the RBIS protocol, was introduced [11], as shown in Fig. 2.

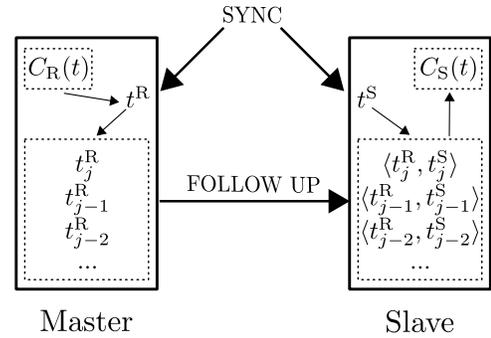


Fig. 2. RBIS protocol (refined from [12]).

In this case the infrastructure sends the cyclic synchronization broadcast and the connected nodes are synchronized, whereas the infrastructure cannot be synchronized using this approach. When the devices get the broadcast each device tags the incoming synchronization message to their recent timestamp that is device specific. Afterwards, the device that acts as master can distribute its reference timestamps t_j^R to all slaves as *FOLLOW UP* message. Then, all slave devices have both, their own and reference timestamps $\langle t_j^S, t_j^R \rangle$ for a specific synchronization message. Thus, the offset between master and slave clocks $C_R(t)$ and $C_S(t)$ can be calculated.

III. RELATED WORK

The benefits of RBIS protocol already motivated several investigations. Thus, [11] evaluated the concept based on Wi-Fi. Since the RBIS protocol requires cyclic broadcasts that all devices receive in parallel, the concept assumes that all devices are connected to the same infrastructure, i.e., all stations are connected to a single Access Point (AP), using Wi-Fi as communication system. In order to achieve a wider service area, [13] proposes

a concept for the possible extension of the RBIS protocol. It is based on the idea that there are Wi-Fi stations that are in the range of more than one AP and consequently receiving multiple synchronization broadcasts. If the time offset between these transmissions is considered, also clocks of devices can be synchronized that are only connected to different APs.

Furthermore, the use of the RBIS protocol not only in IEEE 802.11 but also mobile radio communications was investigated [14]. Accordingly, a first draft for using RBIS protocol in 5G was proposed, but the concept was evaluated using 4G. Moreover, no performance results on the clock synchronization of the UEs were provided. The evaluation was only analysed regarding measurements of a discrete factory automation demonstrator that requires a clock synchronization of 1 ms and less.

IV. MAPPING OF RBIS TO 5G

This section covers the application of RBIS protocol in 5G. In the beginning, Sec. IV-A describes the provision of a synchronization signal, Sec. IV-B a delay correction for using RBIS protocol in 5G, and Sec. IV-C introduces the implemented algorithms.

A. Cyclic Synchronization Signal

In order to map the RBIS protocol to 5G, a potential synchronization message has to be identified, whereas this message has to be received by all devices that have to be synchronized. Furthermore, it is important that this message is send cyclically and contains an unique identifier that the next message can be differentiated from the recent one. This is required to assign the timestamp t_j to a specific message. The so-called Physical Broadcast Channel (PBCH) that is located inside the Synchronization Signal Block (SSB) alongside Primary Synchronization Signal (PSS) and Secondary Synchronization Signal (SSS), broadcasted in the entire network for initial network access, is well suited for this task. Further, the SSB is the only always-on signal in 5G. This means that each UE receives the signal even if it is not connected.

Moreover, the PBCH contains the so-called System Frame Number (SFN). The SFN is a 10-bit counter variable that is incremented every 10 ms. This means that this value repeats every 1024 ticks and is not unique, however, a repetition of every 10.24 s is viable, if $C_R(t) - C_S(t) \leq 10.24/2 = 5.12$ s. In addition, the PBCH is send in an interval that can be adjusted by the base station (gNB) between 5 and 160 ms, which usually depends on the use of the corresponding carrier. A UE expects a new SSB at least every 20 ms in 5G NR for cell search in an initial (random) access. Thus, the basic requirements necessary for RBIS with respect to the cyclic synchronization signal are already fulfilled by utilizing existing 5G NR functionalities.

B. Runtime Correction

Since the RBIS protocol is based on the *receiver/receiver* principle, it is assumed that all devices receive the synchronization signal at the same time. This assumption leads to an error depending on the distance deviation Δs between gNB and the different receivers (see Fig. 3). While Δs is small for UEs that

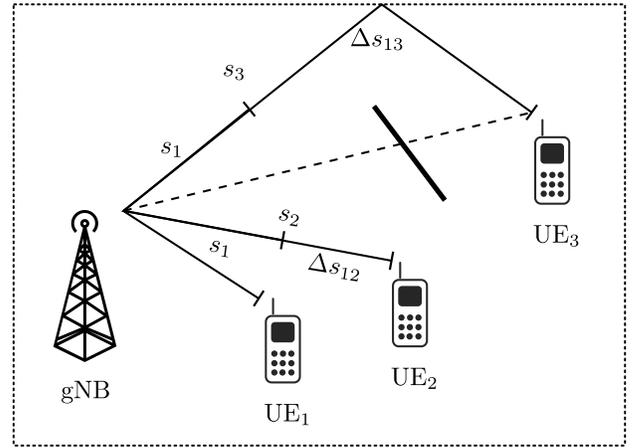


Fig. 3. Time synchronization error due to different Time of Arrivals (TOAs).

are equidistant to the gNB, a variation of the distance of both UEs to the gNB leads to an increased Δs . Thus, for high values of Δs , a error correction is required. For mobile communication systems, determining the UE-to-gNB propagation delay is already of great importance: Due to the provision of frequency- and time-distributed resource elements, the gNB expects the packets of the respective UE at very strict time slots. So called uplink synchronization must be guaranteed. For smaller time deviations, a cyclic prefix was introduced both between symbols and slots. These are short temporal buffer areas in which no user data is to be received, in order to avoid the overlapping of symbols arriving slightly delayed or ahead of schedule. With large spatial distances and associated relatively long transit times, however, a transmitted OFDM symbol will also exceed the limits of the cyclix prefix by far. Here, the Timing Advance (TA) comes into play [15]. TA ensures by a time correction that the uplink symbols from the UE are sent ahead of time so that they arrive at the gNB in the correct time slot. The base station determines the time difference to be set via the uplink signals received and transmits this to the UE via the Timing Advance Command (TAC). The time T_{TA} (see Eq. 3), assuming the same signal propagation delay t_{prop} in the uplink and downlink, describes in 5G NR the time difference to be considered for sending uplink symbols. In other words: the uplink frame should be sent by the UE exactly T_{TA} before the corresponding downlink frame from the gNB is sent back to the UE.

$$T_{TA} = (N_{TA} + N_{TA,off}) \cdot T_c = 2 \cdot t_{prop} + T_{off} \quad (3)$$

As can be seen in Eq. 3, T_{TA} depends not only on the signal propagation delay t_{prop} between UE and gNB but also on an fixed offset T_{off} . The offset is caused by the base station having to switch from receive to transmit between receiving the uplink from and transmitting downlink symbols to the UE.

It should be mentioned that in 5G NR time values are basically specified in a unique step size, the so-called Physical Layer Time Unit T_c . This can be expressed via $T_c = 1/(\Delta f_{max} \cdot N_f) \approx 0.509$ ns, where Δf_{max} is the maximum defined Subcarrier Spacing (SCS) frequency of 480 kHz and N_f the FFT size of 4096. Thus, T_c

can be considered as 5G NR sampling time. The propagation delay t_{prop} is also determined in T_c step size and specified via N_{TA} . There are two calculation schemes in 5G NR: An initial determination of N_{TA} during the random access procedure via the Random Access Response (RAR) message as well as continuous updates of N_{TA} via the MAC Control Element (MAC CE), which are both specified as TAC messages. The calculation of N_{TA} is performed for the RAR procedure according to

$$N_{\text{TA}} = \frac{N_{\text{TAC,RAR}} \cdot 16 \cdot 64}{2^\mu} \quad (4)$$

and for the MAC CE updates according to

$$N_{\text{TA,new}} = N_{\text{TA,old}} + \frac{(N_{\text{TAC,MAC}} - 31) \cdot 16 \cdot 64}{2^\mu}. \quad (5)$$

The variable N_{TAC} is actually the index transmitted as TAC. N_{TAC} in the RAR message has a size of 12 bits and vary between 0 and 3846. In contrast, only a 6 bit N_{TAC} index is transmitted in the MAC CE that is coded as unsigned and lies between 0 and 63. However, the first 31 values correspond to a runtime correction value, as it can be seen by the subtraction in Eq. 5. The parameter μ in Eq. 4-5 indicates the subcarrier spacing used and is based on the following relation: $\Delta f = 2^\mu \cdot 15 \text{ kHz}$.

Thus, TA in 5G NR operates based on a correction in fixed step sizes. To get a better overview of correction accuracies, both the temporal and spatial step sizes for different SCS are shown in Table I. The distance in meters is also a metric to figure out which

Tab. I
TIMING ADVANCE ADJUSTMENT STEP SIZE FOR DIFFERENT
SUBCARRIER FREQUENCIES

μ	Freq. Range	SCS Δf [kHz]	Adjustment Step Size time [ns]	Adjustment Step Size distance [m]
0	FR1	15	520.85	78.13
1		30	260.42	39.06
2	FR1/2	60	130.21	19.53
3	FR2	120	65.10	9.77

distance an UE has to move before the TA value changes. The adjustment time step size is also the minimum possible correction time and limits the accuracy that can be achieved in combination with RBIS. If $\Delta f = 30 \text{ kHz}$, runtime differences up to twice the corresponding adjustment time can be compensated. In this case, this leads to an accuracy of slightly more than half a microsecond.

Also for runtime correction, the 5G NR standard with its TA functionalities provide a well-founded baseline. For this, however, the master UE must exchange its calculated T_{TA} value together with its timestamp to the other UEs, since the TAC response packets are only transmitted to the respective UE and not broadcasted within the cell. Then, slave UEs can perform a RBIS protocol-based synchronization, corrected for the different runtimes, from the combination of their TA value, their own timestamp and the received master values. The offset time T_{off} , on the other hand, is of no significance for the runtime correction,

since it can be assumed that this is the same for all transmitted cyclic broadcast signals.

C. Implementation

In order to implement RBIS protocol in 5G, two different algorithms have to be formulated, whereas Alg. 1 shows the algorithm for the Master UE M . First, it has to be ensured that all Slave UEs S_x are connected to the same gNB. Furthermore, a reference time t^R is required at M . Since the SFN repeats every 10.24 s, the clocks of all S_x have to be at least within the already mentioned 5.12 s limit. Thus, an initial timestamp t_0^R is created and send to all S_x alongside the Masters TA T_{TA}^R . Afterwards, M waits for a PBCH block and creates a timestamp t^R , when a new one is detected. Furthermore, both SFN and μ values are read. Then, a new pair $\langle \text{SFN}^R, t^R \rangle$ is created and sent to all S_x . Since μ defines the the minimal intervals T_I , where all UEs receive a new PBCH block, it does not make sense to send *FOLLOW UP* messages more frequently. Furthermore, N_{TA}^R is included in the *FOLLOW UP* message.

Algorithm 1 Master UE algorithm

Require: $t^R, x \in \mathbb{N}, T_{\text{TA}}^R$

Ensure: M, S_x are in the range of the same BS

- 1: Create timestamp t_0^R
 - 2: Send t_0^R to $S_x, \forall x$
 - 3: **while** true **do**
 - 4: Wait for PBCH
 - 5: Create timestamp t^R
 - 6: Read μ
 - 7: Read SFN^R
 - 8: Create pair $\langle \text{SFN}^R, t^R \rangle$
 - 9: Transmit $\langle \text{SFN}^R, t^R, N_{\text{TA}}^R \rangle$ to $S_x, \forall x$
 - 10: **end while**
-

Thus, the algorithm for the S_x is presented in Alg. 2. After its start up, each S_x waits for the initial timestamp t_0^R of M . When it arrives, a timestamp of its local time t_0^S is taken and the difference between both timestamps is calculated. If the timestamp is within the limit of 5.12 s, no adjustment of $C_S(t)$ is required. In the next step, each S_x waits for an incoming PBCH block and processes it similar to M . After the pair $\langle \text{SFN}^S, t^S \rangle$ is created, S_x waits for incoming *FOLLOW UP* messages. When a *FOLLOW UP* message is received, the included SFN is compared to the most recent one that was included in the last PBCH block. Dependent on the delay of the processing and transmission of the *FOLLOW UP* messages by S_x , it is possible that they differ. In order to get the suitable pair, S_x browses $\langle \text{SFN}_{i-1}, t_{i-1}^S \rangle$ pair until it finds the correct one. Afterwards, the offset that is defined as i is calculated and can be used for further iterations. Last but not least, the offset is calculated and the clock $C_S(t)$ is corrected.

V. TESTBED

In order to evaluate the proposed concept a testbed is used that contains the hardware that is listed in Tab. II.

Algorithm 2 Slave UE algorithm

Require: $t^S, x \in \mathbb{N}, T_{TA}^S$
Ensure: M, S_x are in the range of the same BS

```

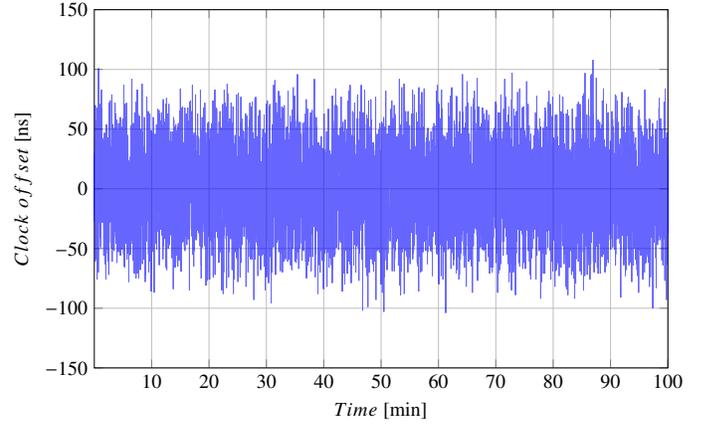
1:  $i \leftarrow 0$ 
2: Wait for  $t_0^R$ 
3: Create timestamp  $t_0^S$ 
4: if  $|t_0^R - t_0^S| \leq 5.12s$  then
5:    $C_S(t) \leftarrow t_0^R$ 
6: end if
7: while true do
8:   Wait for PBCH
9:   Create timestamp  $t^S$ 
10:  Read  $\mu$ 
11:  Read  $SFN^S$ 
12:  Create pair  $\langle SFN^S, t_{SFN}^S \rangle$ 
13:   $SFN^S \leftarrow SFN^S - i$ 
14:  Wait for  $\langle SFN^R, t^R, T_{TA}^R \rangle$ 
15:  if  $SFN^S \neq SFN^R$  then
16:     $i \leftarrow SFN$ 
17:    while  $SFN^S \neq SFN^R$  do
18:       $i \leftarrow (i - 16/2^\mu) \bmod 1023$ 
19:    end while
20:     $i \leftarrow SFN^S - i$ 
21:     $SFN \leftarrow i$ 
22:  end if
23:   $C_S(t) \leftarrow C_S(t) + t_{SFN}^R - t_{SFN}^S - (T_{TA}^R - T_{TA}^S)$ 
24: end while

```

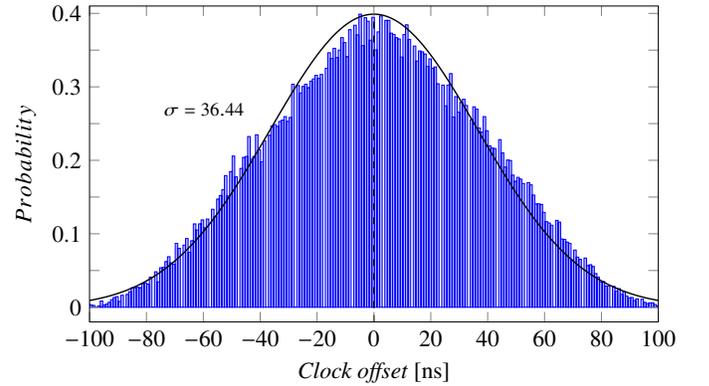
Tab. II
HARDWARE CONFIGURATIONS

Equipment	QTY	Specification
5GC	1	Nokia AirFrame server
5G BS	1	Nokia AirScale platform, AirScale Micro Remote Radio Head
5G UE (PC)	2	Intel Core i9-9900K, 16 GB DDR4, 2x Intel i210-AT NICs, Ubuntu 20.04.1 LTS 64-bit
5G UE (SDR)	2	Ettus Research X310
TSN Eval. Kit	1	RAPID-TSNEK-V0001, IEEE 802.1AS

Since the used 5G Core (5GC) is a in-house development for Nokia's Digital Automation Cloud and operated on a Nokia AirFrame server platform, it is not commercially available. The used gNB, on the other side, is commercially distributed by Nokia under the name AirScale. This is important because it directly proves that our concept does not require any adaptation of the functionality of a gNB. Additionally, two Ettus Research X310 Software-Defined Radio (SDR) platforms in combination with two high performance PCs are used as UEs. Furthermore, each UE runs *OpenAirInterface5G* software that is an open source experimentation and prototyping platform compliant with the 5G



(a) Readings of the measured clock offset between TSN Evaluation Kit and Intel NUC mini PC over time.



(b) Histogram of the probability density functions obtained for the readings of the clock offset.

Fig. 4. Clock offset between TSN Evaluation Kit and Intel NUC mini PC for a sync interval of 31.25 ms (2^{-5} s).

standard. Moreover, a TSN evaluation kit is used. It is connected to the Master UE in order to synchronize its clock to the TSN time. In order to allow time synchronization, the Master UE runs *LinuxPTP* module that is an open source implementation of IEEE 802.1AS and 802.1AS-REV specifications. To perform offset measurements between the TSN time and the synchronized time, the Slave UE is also connected to the TSN evaluation kit, for evaluation purposes only.

VI. EVALUATION

In order to determine the performance of RBIS protocol using 5G offset measurements were performed. Since these measurements rely on a very precise measurement of the TSN time, Fig. 4 depicts offset measurements for the synchronization of the Master UE clock. It can be seen, that a maximum synchronization error of ≈ 100 ns occurs, by using the minimum sync interval of 31.25 ms (2^{-5} s). Furthermore, the precision is 0.014 ± 36.44 ns. Since this value is much lower than the expected clock precision using RBIS in 5G, this error can be neglected. Next, the results of the performed measurements are discussed. Thus, Fig. 5 shows the histogram of the readings obtained for 35,000 data points.

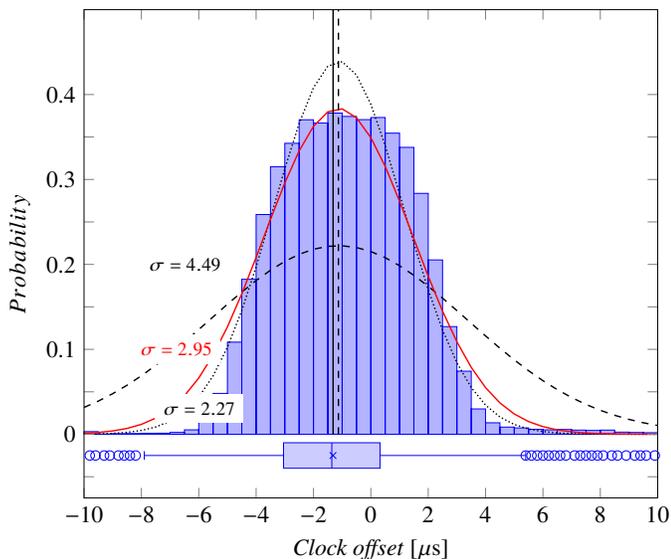


Fig. 5. Histogram of the reduced data set and probability density functions obtained for different standard deviations and corresponding box plot.

Indeed, a total of 180 outliers that exceeded the $\pm 10 \mu\text{s}$ borders and are $< 0.5\%$ of all values are not shown in the plot. Further, three different probability density functions are depicted, obtained for the listed standard deviations. While the dotted density function that belongs to the complete data set is too conservative, the dashed density function for the reduced data set, which contains the remaining 99.5% of all data points, is too optimistic. Thus, the red curve shows an optimized density function for the measured data set. Using this density function as basis, Tab. III lists the measured clock precision for different confidence intervals. Furthermore, P_1 is the probability value for the reduced data set, while P_2 takes the outliers into account.

Tab. III
MEASURED CLOCK PRECISION FOR DIFFERENT STANDARD DEVIATIONS.

	σ	2σ	3σ
$E(x = \mu) [\mu\text{s}]$	-1.32 ± 2.95	-1.32 ± 5.9	-1.32 ± 8.85
P_1	0.6827	0.9545	0.9973
P_2	0.6793	0.9497	0.9937

VII. CONCLUSION

In this paper, we highlighted the impact on a precise clock synchronization on upcoming use cases and technologies. Therefore, we proposed the state-of-the-art in wireline as well as wireless communication systems and addressed corresponding challenges. Furthermore, a mapping of the technology independent RBIS protocol to 5G was proposed. Moreover, chances and possible drawbacks of the RBIS protocol were discussed, before a performance evaluation was carried out. The measurements that were performed with real hardware indicate that with only small error

corrections, a both precise an accurate wireless clock synchronization can be performed, taking the very low overhead into account that this protocol requires. Indeed, a clock accuracy of $1.32 \mu\text{s}$ with precision of $2.95 \mu\text{s}$ was achieved. Thus, approximately 99.5% of all data values of the offset between master and slave clock lie in the interval of $[-10 \mu\text{s}, 10 \mu\text{s}]$.

Future work on this topic aims to further minimize the clock offset. Here, especially the use of a clock discipline algorithm, e.g. as proposed by [12], improves both accuracy and precision of the clock, by taking not only one but multiple timestamp pairs into account using a filter, or by adopting the frequency of the local oscillators. In this context, the temperature in particular is responsible for frequency variations [16].

REFERENCES

- [1] M. Gundall, J. Schneider, H. D. Schotten, M. Aleksy, *et al.*, "5G as Enabler for Industrie 4.0 Use Cases: Challenges and Concepts," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2018, pp. 1401–1408. DOI: 10.1109/ETFA.2018.8502649.
- [2] N. Enayati, O. D. Ramos-Cantor, T. Neugebauer, F. Dai, *et al.*, "Using Cellular Connectivity for On-the-move Cooperation of Stationary Manipulator and Mobile Platform," in *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2021, pp. 1–8. DOI: 10.1109/ETFA45728.2021.9613325.
- [3] I. Godor, M. Luvisotto, S. Ruffini, K. Wang, *et al.*, "A Look Inside 5G Standards to Support Time Synchronization for Smart Manufacturing," *IEEE Communications Standards Magazine*, vol. 4, no. 3, pp. 14–21, 2020. DOI: 10.1109/MCOMSTD.001.2000010.
- [4] D. Krummacker, C. Fischer, K. Alam, M. Karrenbauer, *et al.*, "Intra-Network Clock Synchronization for Wireless Networks: From State of the Art Systems to an Improved Solution," in *2020 2nd International Conference on Computer Communication and the Internet (ICCCI)*, 2020, pp. 36–44. DOI: 10.1109/ICCCI49374.2020.9145977.
- [5] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, "The flooding time synchronization protocol," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004, pp. 39–49.
- [6] T. Neagoe, V. Cristea, and L. Banica, "NTP versus PTP in Computer Networks Clock Synchronization," in *2006 IEEE International Symposium on Industrial Electronics*, vol. 1, 2006, pp. 317–362. DOI: 10.1109/ISIE.2006.295613.
- [7] J. C. Eidson, M. Fischer, and J. White, "IEEE-1588™ Standard for a precision clock synchronization protocol for networked measurement and control systems," in *Proceedings of the 34th Annual Precise Time and Time Interval Systems and Applications Meeting*, 2002, pp. 243–254.
- [8] J. L. Messenger, "Time-Sensitive Networking: An Introduction," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 29–33, Jun. 2018. DOI: 10.1109/MCOMSTD.2018.1700047.
- [9] M. Lipiński, T. Włostowski, J. Serrano, and P. Alvarez, "White rabbit: a PTP application for robust sub-nanosecond synchronization," in *2011 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, 2011, pp. 25–30. DOI: 10.1109/ISPCS.2011.6070148.
- [10] S. Ganerwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003, pp. 138–149.
- [11] G. Cena, S. Scanzio, A. Valenzano, and C. Zunino, "Implementation and Evaluation of the Reference Broadcast Infrastructure Synchronization Protocol," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 801–811, 2015. DOI: 10.1109/TII.2015.2396003.
- [12] G. Cena, S. Scanzio, and A. Valenzano, "A neural network clock discipline algorithm for the RBIS clock synchronization protocol," in *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, IEEE, 2018, pp. 1–10.
- [13] M. Gundall, C. Huber, S. Melnyk, and H. D. Schotten, "Extending Reference Broadcast Infrastructure Synchronization Protocol in IEEE 802.11 as Enabler for the Industrial Internet of Things," in *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, 2021, pp. 118–124. DOI: 10.1109/ICPS49255.2021.9468146.

- [14] M. Gundall, C. Huber, P. Rost, R. Halfmann, *et al.*, "Integration of 5G with TSN as Prerequisite for a Highly Flexible Future Industrial Automation: Time Synchronization based on IEEE 802.1AS," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 2020, pp. 3823–3830. DOI: 10.1109/IECON43393.2020.9254296.
- [15] A. Mahmood, M. I. Ashraf, M. Gidlund, J. Torsner, *et al.*, "Time Synchronization in 5G Wireless Edge: Requirements and Solutions for Critical-MTC," *IEEE Communications Magazine*, vol. 57, no. 12, pp. 45–51, 2019. DOI: 10.1109/MCOM.001.1900379.
- [16] M. Mongelli and S. Scanzio, "Approximating Optimal Estimation of Time Offset Synchronization With Temperature Variations," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 12, pp. 2872–2881, 2014. DOI: 10.1109/TIM.2014.2320400.