Clock Synchronization for IEEE 802.11 based Wired-Wireless Hybrid Networks using PTP

Conference Paper · September 2012

DOI: 10.1109/ISPCS.2012.6336617

2 authors:

A. Mahmood
TU Wien

Felix Ring
Danube University Krems

Clock Synchronization for Wired-Wireless Hybrid Networks using PTP

Aneeq Mahmood, Felix Ring
Institute for Integrated Sensor Systems
Austrian Academy of Sciences
Viktor Kaplan Straße 2
2700 Wiener Neustadt, Austria
Email: {aneeq.mahmood, felix.ring}@oeaw.ac.at

***Abstract*—The current goal for real-time communication infrastructure in factory automation is to merge and balance the best of both worlds, i. e. the reliability of wired networks, and the flexibility of wireless technology. Hybrid wired-wireless networks are proposed for establishing real-time communication infrastructure in factory automation. However, the provisioning of synchronized clocks is often taken for granted, and issues regarding the integration of wired and wireless synchronization are neglected. In this study, the requirements for providing clock synchronization for such networks are investigated for an IEEE 802.11 WLAN based hybrid network with IEEE 1588 as the desired synchronization protocol. Both, simulation and implementation based results of the clock synchronization performance in such WLAN based hybrid networks are presented for performance analysis.**

I. Introduction

The nowadays broad acceptance of Ethernet on the factory floor demonstrates the current trend of adapting existing, standardised communication technologies for automation instead of developing new ones from scratch. The anticipated continuation of this trend puts WLAN into the focus for real-time communication to further ease automation in the factories [1]. Wireless devices introduce features like flexibility and mobility on the factory floor, hence paving the way for a wide range of new applications. The commercial-off-the-shelf (COTS) WLAN hardware is cheap in price and thus attractive for many application scenarios. However, the idea of hundreds of wireless nodes moving 'freely' inside factories is far-fetched for the moment, because high monitoring and control efforts would be needed to ensure safety, security, and quality of service (QoS) and no such precedent is available at the moment. The state-of-the-art is to use a hybrid wired-wireless communication system in factories to combine the advantages of the reliable wired and the flexible wireless technology [2].

However, just as the case has been with Ethernet [1], the COTS WLAN devices are not suitable for real-time communication, as the channel access scheme is based on carrier sense multiple access with collision avoidance (CSMA/CA). This results in non-deterministic transmission times on the wireless channel which is not desirable for industrial communication. It would be unwise to shift away from standardised WLAN hardware, as this would defeat the goal of providing wireless real-time with low-cost, consumer grade hardware modules. A possible solution is a system as described in [3], where a special middleware between application and MAC layer provide the real-time communication along with several others features.

Meanwhile, with all the talk of providing real-time communication in hybrid networks, the issue of synchronizing the clocks and providing the common time base for the system is little dealt with. Often the clocks of the system are assumed to be synchronized [4] before-hand. Providing clock synchronization for a hybrid network architecture is not as straightforward as combining existing solutions. Indeed, two different communication technologies are present in such networks and a homogeneous time base is required which should be compatible with both technologies. This paper focuses on the requirements for clock synchronization for such a hybrid system and provides simulation and implementation results regarding the performance of IEEE 1588 [5] for a reference real-time capable hybrid network. The chosen scenario consists of a hardware (HW) timestamping enabled wired Ethernet part, as devices are already available on the mass-market. The WLAN part is made up of COTS devices and only supports software (SW) based timestamping.

The rest of the paper is structured as follows: Section II provides work related to the issues being dealt with in this paper; Section III highlights requirements for clock synchronization in hybrid networks; Section IV discusses various implementation aspects for synchronization in a typical hybrid network; Section V includes performance evaluation for clock synchronization for hybrid networks using both simulation and actual results. The conclusion and further discussion are covered in the last section.

II. Related Work

Though the issues of hybrid networks in automation have been discussed in literature, most of the solutions have fo-

This project was partly financed by the province of Lower Austria, the European Regional Development Fund, the FIT-IT project Ætas under contract 825904, the EU under the FP7 STREP ^{flex}WARE contract number 224350 as well as the Austrian Center of Competence in Mechatronics (ACCM) under contract A260501 and Oregano Systems Design and Consulting.

cused on providing a communication framework for such a system and have not discussed the provisioning of a common timebase. The authors in [2] have discussed the design and feasibility of a wireless fieldbus without considering synchronization. Another architecture for industrial communication has been proposed in [4], but the authors have presumed clock synchronization support in the system and have focused more on the quality of service (QoS) and mobility support in hybrid networks.

For a hybrid network, authors in [6] have presented a Precision Time Protocol (PTP) implementation which simply replaces the wired gateway to end-devices with a COTS wireless router. As the channel access mechanism of the wireless interface of the router is based on collision avoidance, the packet transmission can take considerable time, suffering non-deterministic forwarding delays. This not only affects the real-time performance of the system, it also degrade the quality of the timestamps of the `Sync` packets. As a solution, the authors have proposed to increase the `Sync` rate on the channel. This scheme hence offers a variable synchronization overhead without solving the fundamental problem of non-deterministic delays. The final synchronization jitter is in the range of milliseconds, which is deemed too high for synchronous networks such as network control systems [7]. Other similar implementations [8] are limited only to simulation and do not provide much insight about timestamping jitter or channel access delays.

III. Clock Synchronization Requirements in Hybrid Systems

Designing a hybrid real-time system can not easily be viewed separated from designing clock synchronization for such a system, as both tasks put requirements on each other. Some of these requirements are listed below:

a) Synchronization Accuracy: This requirement is defined by the targeted application. Applications which demand a clock synchronization accuracy of a few nanoseconds require HW timestamping support. A system with COTS WLAN devices can not guarantee such high accuracies, and is therefore better suited to be used for applications which require low synchronization accuracy in the range of milliseconds or microseconds. Hence, the presence of COTS WLAN devices can decide which applications the hybrid network can support and which not.

b) Shallow Depth: PTP has been developed for systems with a limited number of network elements like boundary clocks (BCs) and transparent clocks (TCs). Cascading BCs leads to an exponential increase in jitter, while the jitter with cascaded TCs only increases linearly [9]. However, if the TCs are not syntonized to the grandmaster, or if any of the TCs has a poor quality oscillator, the residual time calculations will be erratic. Measurements with hardware timestamping enabled network elements showed that the end-to-end (E2E) synchronization accuracy can be in the high nanosecond or microsecond range [10]. In case of software timestamping, the number of network elements is critical, as each element adds a significant jitter.

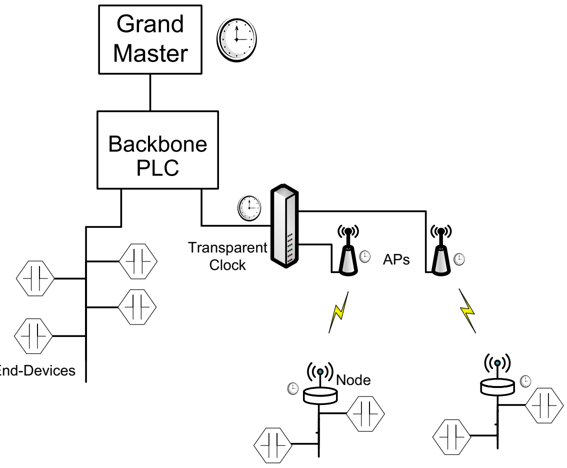

Fig. 1. A simple hybrid architecture showing wireless segments and clocks involved

c) Wireless Compatibility: PTP has been primarily developed for wired networks. Several profiles have been added for various wired communication technologies. It can not be seamlessly deployed over WLAN without making some changes in the protocol such as disabling multicasting for `Delay_Req` messages as discussed in [11]. Additionally, the synchronization traffic should produce little overhead on the bandwidth limited channel and be sent with high priority in order to reduce transmission delays.

d) Scalability: To avoid overloading one AP with a lot of nodes in one cell, the network should be scalable, so that new APs can easily be added for load sharing. Provisions should be made so that all APs synchronize to the same time source. Additionally, the mobile nodes can roam from one AP to another, and thus PTP should make sure that the roaming node chooses a new master before its clock loses synchronization.

IV. Basic Synchronization Architecture in Hybrid Systems

Fig. 1 displays a basic hybrid architecture and several clock entities associated with the devices in the hybrid system. From PTP's perspective, the grandmaster is an atomic clock or a GPS-synchronized clock which acts as the reference clock to the backbone network. The major backbone networking element is a programmable logic controller (PLC) which directly or indirectly controls a lot of I/O devices on the factory floor. This backbone communication technology should be PTP compliant in order to synchronize the I/O devices to the grandmaster clock. One such communication technology is IEC 61158 Type 10 (PROFINET), which supports transport of PTP messages as described in Annex I of the IEEE 1588v2 [5] standard.

A switch, acting as a TC, is required to connect several APs to the backbone network. The APs then control the wireless part of the system and provide connection to the nodes which are connected to the end-devices as shown in Fig. 1. In this topology, the APs act either as TC or BC, and the nodes are ordinary clocks. In such a topology, either the grand master clock or the AP will be the master clock to the nodes. With the nodes having a static status as slave clocks, they will not try to become master clock during initialisation phase or in the case of master clock's break down. Hence, such an arrangement will avoid any reconfiguration of the clocks in a cell [10].

A. Pre-requisites for Clock Synchronization in Hybrid Systems

For a real-time hybrid system, the presence of a wireless network raises the need for a deterministic communication over the wireless medium. This, in turn, requires the presence of an application-layer based wireless synchronization mechanism over WLAN using only COTS devices. These two issues of wireless synchronization and establishing wireless real-time communication are discussed below:

Wireless Clock Synchronization: Under normal operations [12], a WLAN device is responsible to synchronize to its associated AP using the Timing Synchronization Function (TSF). The TSF timer in the AP is independent of the AP's system time. Therefore, in order to synchronize the end-devices to a grandmaster, the AP would need to synchronize its TSF timer to the (PTP controlled) system time. Likewise, the system time at the node would need to be synchronized to the TSF timer. The TSF timer at the node is controlled by the WLAN hardware and may be reset upon loss of connectivity. Therefore, using the TSF timer for node synchronization makes wireless synchronization complicated and unpredictable. A simpler solution is to synchronize the node to the AP using PTP on the application level. Such an implementation has already been carried out in [11]. Using COTS WLAN devices with software timestamping a sub-microsecond accuracy could be reached.

Deterministic Propagation over Wireless Medium: To ensure real-time communication over the wireless medium, the contention based channel access scheme of WLAN needs to be replaced with a contention free scheme like TDMA to ensure deterministic propagation over the channel. One such scheme is presented in [13], where the authors have proposed the use of open-source WLAN device drivers to control the random back-off time in the MAC layer and transmit all the real-time packets with the shortest possible inter-frame space, thus controlling the medium.

To establish real-time communication, the beacon frames are sent at fixed intervals and delimit so-called superframes. The beacon frames contain the TDMA schedule and distribute it to the nodes. Each superframe consists of a contention-free phase (CFP) and a contention phase (CP). The CFP starts right after the beacon frame is sent and first handles the downlink traffic from the AP to the nodes in a cell. Then the nodes start sending their uplink traffic to the AP, one after another, according to the distributed schedule. After the last node is finished, the CFP is over and the CP begins, where the nodes, APs, and all legacy WLAN devices can communicate using the standard CSMA/CA method described in [12]. The start

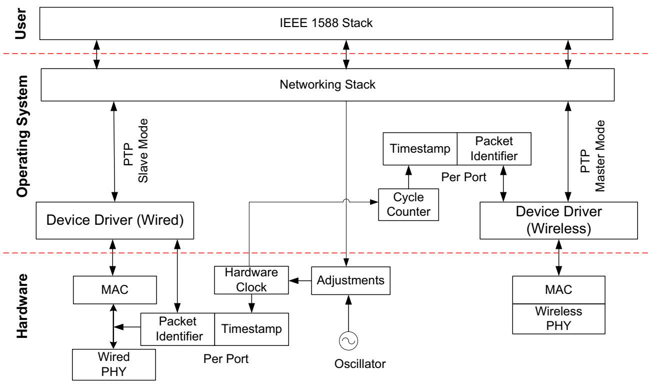

Fig. 2. A basic hybrid boundary clock structure with per port/interface PTP implementation

of the next beacon framen then indicates a new superframe, comprising of a CFP and a CP. As the CFP phase is reserved for the real-time traffic coming from and going to the PLC over the wireless channel, the best-effort traffic, including all PTP messages, is only allowed in the CP.

B. Wired-Wireless-Interface Clock

An AP can either be implemented as a PTP TC or BC. For a TC implementation, the AP would behave as a wired-wireless bridge, and all the PTP traffic coming from the wired interface can be directly passed on to the wireless interface. However, for a TC implementation, the packet residence time inside the AP should be calculated and compensated using hardware timestamps. The wired interface can be equipped with a one-step hardware clock, but the wireless interface uses COTS hardware without HW timestamping support, hence affecting the accurate residence-time calculation for the TC. Additionally, as the PTP traffic is only allowed in the CP, the packet has to wait in the AP and will not be sent out as soon as possible as indicated in [5]. Thus, for the better control of PTP traffic on the channel, the AP will act as a hybrid BC (HBC). A basic architecture of such a clock is shown in Fig. 2

Compared to a traditional wired BC, the HBC presents a different design paradigm. For a start, the number of end-devices attached to a traditional wired BC depends on the number of available physical interfaces on the BCs. On an HBC, only a single wireless interface is required to connect to numerous end-devices. Thus, the design of an HBC is a lot simpler than of a traditional BC, as a typical HBC can be realised with only one wired and one wireless physical interface. The downside is that the wireless channel is only half duplex, and its software timestamps are less accurate than hardware timestamps, which are reserved for the wired interface.

Similar to a typical BC, the HBC implementation may have more than one PTP port per physical interface, but only one port per interface is considered for a simpler design as highlighted in Fig. 2. Nevertheless, the per port/interface implementation inside the HBC will be different for the wired and the wireless interface. At first, given the topology in Fig. 1, the wired interface will always be the slave port, and the wireless one will be in the master state. Thus, the PTP implementation should fix the status of the ports before-hand, based on the interface. Additionally, separate per-interface implementations should be carried out for packet identification and timestamping inside the HBC because of the different PHY and MAC headers for Ethernet and WLAN. For Ethernet, this takes place inside the hardware itself while for WLAN, packet identification for software timestamping can only take place when the hardware has handed the packet to the device driver as shown in Fig. 2.

Lastly, the timestamping will be different for the two interfaces though they share the same clock. The SW timestamping will be drawn by scheduling a call to read a higher abstraction of the clock (e. g. a counter), while the HW timestamp will be drawn by reading the time registers of the hardware clock directly as indicated in Fig. 2.

C. Wireless Support for PTP

With the help of a basic HBC, the wireless devices can be synchronized to the grandmaster. However, further wireless support can be provided in such networks to make the system more scalable and reduce the synchronization overhead on the network. These methods are discussed below:

Fast Master Handover: Although the AP is always the master clock for the node, there is a need to support fast roaming of nodes from one AP to another, in order to make the system more scalable and to accommodate new applications in the system. Hence, the protocol implementation inside the HBC should make sure that the roaming nodes do not lose synchronization upon handover. According to the IEEE1588v2 standard, in case the existing master clock goes down, the slave will wait for the `PTP_ANNOUNCE_RECEIPT_TIMEOUT` before changing its state from slave to listening state. In the case of a handover the same situation is experienced by a node, and the node will ignore all `Sync` messages from the new AP. Once in the listening state, the slave will wait for at least two `Announce` messages from the new master clock before it goes back to the slave state. Therefore, in the case of a handover, although the slave will not try to be the master, it will ignore the `Sync` messages from the new AP and its clock will be running freely thus jeopardising the real-time communication. A solution for this issue is that the slave should accept all `Announce` and `Sync` messages from the new AP and bypass the default best master clock algorithm. As the node can only be associated to one AP at a time, and can only have one synchronization master at a time, the node will accept all the PTP messages from its current AP and will stay synchronized during and after a handover. For this solution to work the WLAN driver of the slave must make sure that the ID of the master clock (MAC address of the AP) of the incoming PTP messages matches the ID of the AP the node is associated to. This ensures that the slave will accept only the PTP messages from its own AP and not from any other AP which is operating on the same frequency.

Low Synchronization Overhead: As already discussed, the PTP event and general messages are sent in the CP, where the PTP packets have to compete with traffic from other nodes in the network. This can either be PTP traffic, such as `Delay_Req` messages, or some other traffic associated with the end-devices. Depending on the network load it can happen that the PTP traffic from the AP is not transmitted in the CP phase before the next superframe starts. In a heavily loaded network this could even lead to longer periods of time where no `sync` messages are received, resulting in free running nodes which deteriorate the synchronization, and eventually threaten the whole communication in the network.

To solve this issue, the `Sync` and `Announce` messages can be embedded in the wireless beacon frame, as discussed in [13]. The IEEE 802.11 beacon frame has a vendor specific field which can contain tagged data. This field can be used to embed PTP messages. On the slave side, the PTP related information needs to be extracted from the beacon frame and put into dummy PTP frames, so that this solution is transparent to the higher protocol layers. This method significantly reduces the PTP traffic overhead, which is another argument for operating the AP as an HBC instead of a TC.

It should be noted that, in addition to the CSMA based channel access, the PTP packets can fail to reach the slave. The wireless channel is always prone to disturbances or subject to shadowing effects. If the beacon frames are not received by the slave, its clock will start to drift from the master. Hence there is always the risk of a synchronization break-down because of the unreliable nature of the wireless medium. However, the failure of a beacon frame to be received by a node has a more severe impact on the system than only losing synchronization. If the communication between the AP and a node is interrupted for a longer period of time, the node will be disassociated from the AP. In such a scenario a broader solution for reintegrating the node needs to be employed, with resynchronization being only a part of the problem. However, if a system is designed to overcome such challenges, the beacon frames can carry PTP information in a very efficient manner.

V. Performance Evaluation

To evaluate and analyse the synchronization performance of the hybrid networks, a two-fold approach involving both simulation and implementation is used. The simulation allows to flexibly extend a system to a large number of participants, and allows to monitor interfaces, states, and values which are unreachable in hardware. This makes it very practical for the evaluation of large and complex systems, like a hybrid network with distributed clocks. The hardware implementation, on the other hand, can provide insight about the behaviour of the system in practice and highlight any potential dependencies or shortcomings in the system which are not covered in the simulation.

A. Simulation

In order for the simulation model to produce valid results, it needs to model the characteristics of the real system. A typical simulation tool for network simulations is a Discrete Event Simulator (DES). Unfortunately, common DESs do not provide support for the simulation of distributed clocks. The authors have developed a local clock model for OMNeT++, which is described in more detail in [14]. The local clock model has two tasks. It needs to model the imperfections of a real oscillator, incorporating long-term drift as well as short term jitter, and it needs to serve as a timing interface between the local time-base and the global simulation time-base. Simulation components that operate with the local clock need to register their timing events at the local clock module, which then calculates and schedules events at the simulation engine with the global time-base.

The local clock is set to simulate a 100 MHz clock. This defines the timestamp granularity to 10 ns, which has a major impact on the clock synchronization accuracy on the wired part.

In order to simulate the software-based timestamping performance of the COTS wireless hardware, a stochastic delay is added to every timestamp that is taken. As interrupt latencies are very platform dependent, there are not many general models available. Different probability distribution functions have been proposed, but measurements like in [15] rather fit an Erlang distribution. This is backed by viewing the interrupt handling process as a queue in the broadest sense. As queues usually are modeled with an Erlang-k distribution, such a distribution is used for the simulation, with $k = 3$ and $\theta = 400$.

B. Measurement Setup

For establishing the synchronization accuracy for an actual hybrid network, a simple test setup is created involving a time master, an AP and a node. A single AP can be connected directly to the master clock via Ethernet and hence there is no switch between the AP and time master. Therefore, no TC exists for this measurement setup. The main hardware for all the three afore-mentioned devices consists of 2.4 GHz industrial PCs. Each industrial PC is equipped with a syn1588® network interface card (NIC) which carries a 25 MHz oscillator and an adder-based clock. This NIC can also generate a one pulse per second (1PPS) signal from the oscillator which can be compared on an oscilloscope to establish the synchronization accuracy of the clocks against each other.

The WLAN hardware consists of COTS Atheros 5212 chipset which is supported by open-source drivers. The wired synchronization takes place using the Ethernet based profile of PTP with HW timestamping, while PTP over UDP is employed for wireless synchronization. The SW timestamps for wireless synchronization are drawn from a counter which is driven by the adder-based clock of the NIC. These timestamps are taken whenever the interrupt signal from the wireless NIC eventually notifies the driver about successful transmission or reception of a packet. For the measurements, standard PTP messages are employed on the wireless channel, beacon frames are not used for this purpose. A `Sync` interval of 1 s is used. The load on the wireless network is also kept fixed to marginalise its impact on the synchronization measurements.

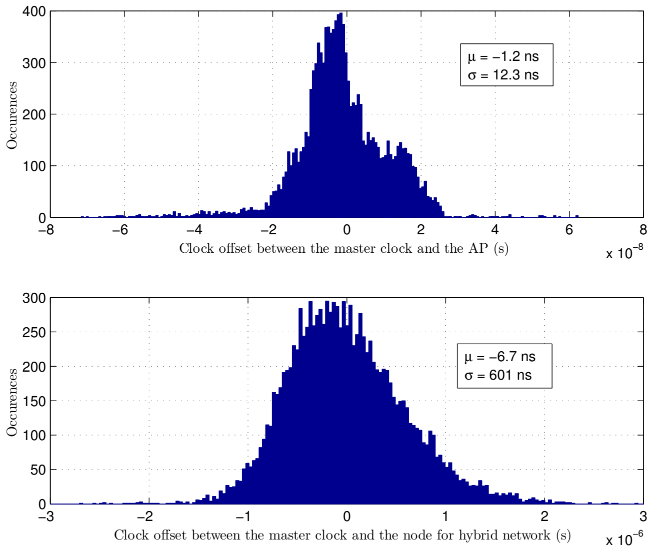

Fig. 3. Simulated results of clock offset in hybrid system

C. Results

1) Simulation results: The upper part of Fig. 3 shows the histogram for the clock offsets of the simulated wired part of the system, i. e. between the master and the AP. The simulated imperfections of the quartz oscillator, together with the limited timestamp resolution due to the 100 MHz clock are the main impacting factors on the clock offset.

The lower part shows the histogram of the simulated clock offset of the hybrid system, i. e. between the master and the wireless node. Because of the simulated additional jitter on the timestamps of the wireless part of the network, the main impact on the clock offset now comes from the timestamp uncertainty. Hence, the standard deviation is almost two orders of magnitude larger than for the wired only synchronization.

2) Measurement results: Figure 4 presents the histogram for the clock offsets between the master clock and the HBC (AP), and between the master clock and the wireless end-device. The better accuracy for the wired synchronization comes from HW timestamping support. The existing offset and jitter comes from the oscillator's imperfections and delays in the cable. The software timestamping in the wireless part of the hybrid system suffers from the interrupt handling delays inside Linux and hence, the final end-to-end synchronization of the system is approximately two orders of magnitudes inferior to the wired synchronization. It should be noted that the results for end-to-end synchronization are similar to the ones obtained only for wireless synchronization in [11]. This means that the major hindrances in achieving higher accuracies than the ones mentioned in this paper comes from software timestamping on the wireless channel. Therefore, the addition ried out. Both methods display the same basic trend regarding the synchronization accuracy of the system and indicate the areas where further improvements are needed. In this vein, the future work involves implementing beacon frame embedded PTP messages in the wireless network, quantifying the variable interrupt delays in the operating system, investigating the effect of changing channel conditions, and the resulting varying wireless path-delays.

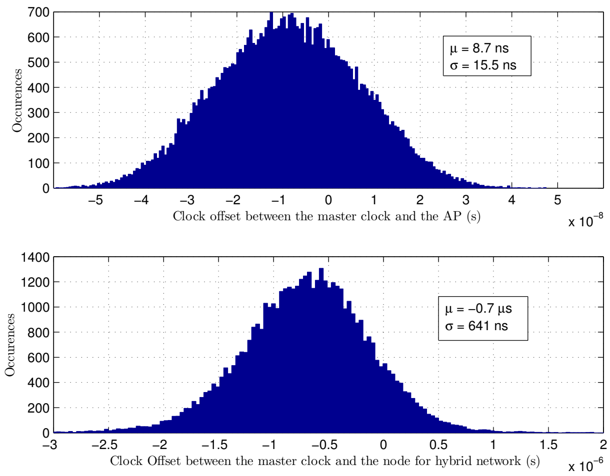

Fig. 4. The measure of clock offsets in the hybrid networks

of any more HBCs in the system will further decrease the final synchronization accuracy. To be able to achieve the required accuracy in the range of a few microseconds, the depth of the network should be limited to one HBC.

3) Comparison of simulation and measurements: When comparing the results of the simulation and the measurements, two major differences are apparent. Firstly, the offset distribution of the wired connection (Master–AP) appears to be much smoother in the measurement. This is due to a not perfectly-simulated oscillator. A more realistic model consists of several computational intensive filters, which would slow down the simulation without having a significant impact on the results, as the local clock model still allows to simulate distributed systems with individual, imperfect clocks.

For the wireless measurement, a large mean value is observed compared to almost none at all in the simulation. One reason for the high mean value can be the asymmetry in the interrupt handling of sending and receiving packets. Further investigations are planned on that end.

Lastly, for a software timestamping based hybrid network, the synchronization accuracy achieved in this work via both simulation and implementation is vastly superior to the hybrid implementation in [6] and the simulation results from [8].

VI. Conclusion

This study presents a PTP implementation over a wired-wireless hybrid system where the wireless segment is supported by COTS devices and only employs software time-stamping for synchronization. This work has discussed basic features which a hybrid system should possess, in order to benefit from the flexibility of wireless communication while still providing a required synchronization accuracy. Also, the design of a hybrid BC has been analysed in this work and it is discussed how PTP implementations should be adjusted to adapt to hybrid systems. To study the performance of the system, simulation and an actual implementation has been carried out.

References

[1] T. Sauter, “The three generations of field-level networks - evolution and compatibility issues,” *Industrial Electronics, IEEE Transactions on*, vol. 57, no. 11, pp. 3585–3595, Nov. 2010.

[2] G. Cena, A. Valenzano, and S. Vitturi, “Hybrid Wired/wireless Networks for Real-time Communications,” *Industrial Electronics Magazine, IEEE*, vol. 2, no. 1, pp. 8–20, Mar. 2008.

[3] G. Gaderer, P. Loschmidt, and A. Mahmood, “A novel Approach for Flexible Wireless Automation in Real-Time Environments,” in *Proc. of the IEEE International Workshop on Factory Communication Systems,WFCS*, G. Cena and F. Simonot-Lion, Eds., IEEE. IEEE, May 2008, pp. 81–84.

[4] T. Sauter, J. Jasperneite, and L. Lo Bello, “Towards New Hybrid Networks for Industrial Automation,” in *Emerging Technologies Factory Automation, 2009. ETFA 2009. IEEE Conference on*, 2009, pp. 1–8.

[5] “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. c1–269, 24 2008.

[6] S. Schriegel, H. Trsek, and J. Jasperneite, “Enhancement for a Clock synchronization Protocol in Heterogeneous Networks,” in *Proc. of the International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication, ISPCS*, Brescia, Italy, Oct. 2009.

[7] J. Moyne and D. Tilbury, “The emergence of industrial control networks for manufacturing control, diagnostics, and safety data,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 29–47, Jan. 2007.

[8] H. Wang, J. Huang, Y. Shi, L. Li, and B. Li, “Research on clock synchronization of wireless distributed measurement system,” in *Testing and Diagnosis, 2009. ICTD 2009. IEEE Circuits and Systems International Conference on*, Apr. 2009, pp. 1–4.

[9] D. Mohl and M. Renz, “Improved synchronization behavior in highly cascaded networks,” in *Proc. of the International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication, ISPCS*, Oct. 2007, pp. 96–99.

[10] J. C. Eidson, *Measurement, Control, and Communication Using IEEE 1588*. Spinger, 2006, pp. 45–50, 62–67.

[11] A. Mahmood, G. Gaderer, H. Trsek, S. Schwalowsky, and N. Kerö, “Towards High Accuracy in IEEE 802.11 Based Clock Synchronization using PTP,” in *Proc. of the International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication, ISPCS*, Munich, Germany, 2011.

[12] *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. 802.11-2007, 2007.

[13] H. Trsek and J. Jasperneite, “An Isochronous Medium Access for Real-time Wireless Communications in Industrial Automation Systems - A Use Case for Wireless Clock Synchronization,” in *Proc. of the International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication, ISPCS*, 2011, pp. 81–86.

[14] F. Ring, G. Gaderer, A. Nagy, and P. Loschmidt, “Distributed Clock Synchronization in Discrete Event Simulators for Wireless Factory Automation,” in *Proc. of the International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication, ISPCS*, Oct. 2010, pp. 103–108.

[15] P. Regnier, G. Lima, and L. Barreto, “Evaluation of interrupt handling timeliness in real-time linux operating systems,” vol. 42, no. 6. New York, NY, USA: ACM, Oct. 2008, pp. 52–63.