ISPCS 2009 International IEEE Symposium on Precision Clock
Synchronization for Measurement, Control and Communication
Brescia, Italy, October 12-16, 2009

Timestamping for IEEE 1588 based Clock Synchronization in Wireless LAN

Aneeq Mahmood and Georg Gaderer
Institute for Integrated Sensor Systems
Austrian Academy of Sciences
2400 Wiener Neustadt, Austria
Email: {Aneeq.Mahmood, Georg.Gaderer}@oeaw.ac.at

***Abstract*—Clock synchronization is one of the enabling technologies for real-time networking. This paper is motivated by the need for WLAN support of real-time factory automation networks. These networks need synchronized clocks for the typical time-slotted arbitration schemes. As this type of service is not yet available wireless LANs, this work investigates possibilities for synchronizing clocks, based on higher layer synchronization approaches such as IEEE 1588 over WLAN. Different clock synchronization schemes are discussed in this study, some of those utilize the features of IEEE 802.11 and some of them not. In addition to that, this currently still ongoing work shows the difficulties of provision of highly accurate timestamps for the sake of synchronization.**

I. Introduction

The overall goal of clock synchronization, in general, is to have spatially distributed clocks adjusted in a way that the difference between any two clocks is below a certain margin for any moment in time. This goal, also called *internal* clock synchronization, can be also refined by a second requirement that any clock must not exceed a certain error compared to a global timescale, like UTC or GPS-Time. This second requirement is called *external* clock synchronization and implicitly fulfils the goals of the internal clock synchronization.

In order to achieve this many different measures can be taken. There are some processes where the requirements to the synchronized clocks are so low that is common practice to manually adjust the clocks once and have them free-running until the next manual maintenance event. Examples for this very primitive clock synchronization are simple wristwatches in our everyday live or non-networked embedded systems for control and monitoring, like simple temperature probes. In the last case, deviations of the logged data of several seconds or even minutes are acceptable.

Besides these low performance requirements, a more precise coordinated time within a distributed system is also an enabler for a large number of other applications. The application domains are situated from test and measurement as in the LXI approach [1] to applications for telecommunication [2]. A final use case, which is the motivation for this paper, is the application of synchronized clocks in factory automation. This use-case for real-time networks, which very often use TDMA based approaches is very well understood, which can be seen in industrial products like EtherCAT, PROFINET or Ethernet/IP. As all these approaches are usually fixed on (wired) industrial Ethernet, a large amount of applications require to generalize the concept of these networks to the wireless domain. Among many candidates for a communication technology, wireless LAN has been chosen for this paper as it is the office proven counterpart of Ethernet, which is enabled to build network structures opposed to simple cable replacements as in the case of Bluetooth. Finally, the motivation for this work in progress paper is to investigate the possibilities for clock synchronization in these networks, which is one of the key aspects to introduce real-time into IEEE 802.11 [3].

The remainder of this paper investigates first the general possibilities for clock synchronization, the basic conditions for wireless LAN and the related work on the topic of WLAN clock synchronization in a review of related work. This is followed by a detailed analysis of the problem together with a proposal for a new solution. Finally, the basic issue of accuracy of timestamps for synchronization in WLAN is raised. This is round up by conclusive comments and description of further work in this direction.

II. Wireless LAN Clock Synchronization

In the general case, the principle of clock synchronization is not restricted to a communication technology, which is represented the fact, that the Precision Time Protocol (PTP, IEEE 1588) [4] is held fully abstract and specifies only in its annexes the actual communication specific details needed for operation.

A. Clock Synchronization Approaches

As already mentioned, the principle functionality to synchronize clocks can be generalized to a very far extent. The two probably most commonly used protocols Network Time Protocol (NTP) [5] and the more precise PTP do clock measurements typically in the same way. Whereas the IEEE 1588 standard only specifies the protocol to obtain a remote time, the typically less accurate Network Time Protocol specifies the servo and the control mechanisms as well. Usually the approaches use two steps to adjust their clocks.

- In a logical (but not mandatory chronological) first try, the communication delay between two nodes (clocks) has to be determined. This can be done under the assumption of a symmetrical delay by measuring the round-trip delay.

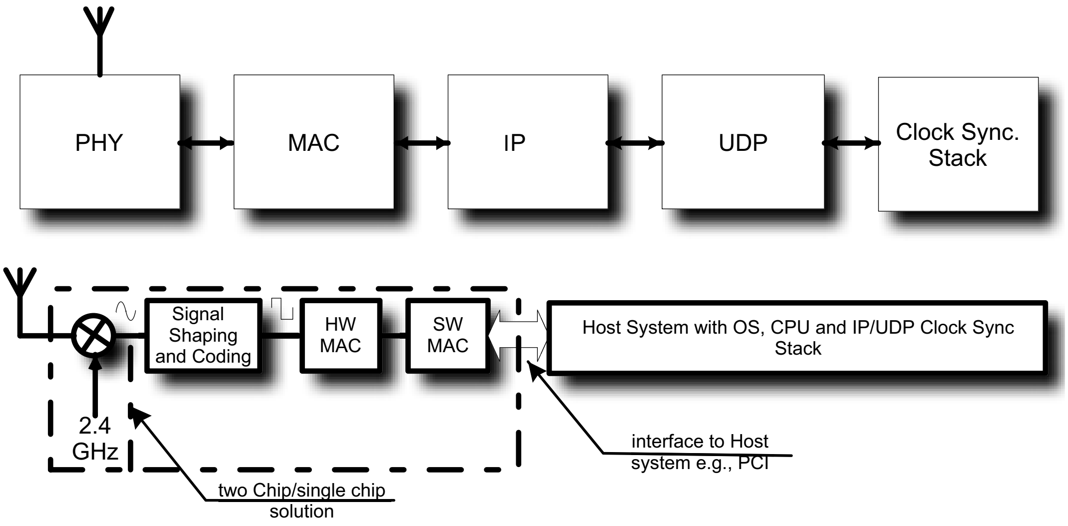

Fig. 1. Typical block diagram of a IEEE 802.11 b/g receiver

The idea behind that is to send a packet to a remote clock and returning it as fast as possible. If one subtracts the residual time at the remote clock, the traveling time of the packet can be calculated. In order to cope with changing delays this measurement has to be done periodically.

- The second part of the synchronization is then done by sending the actual time of each clock over the network. This transmitted time is then compared with the arrival time at the remote node and corrected with the communication delay.

Out of that principle, two conclusions can be drawn: First the functionality relies on a stable delay. If the delay for sending a message changes heavily, the adjustment of the clocks after the reception of a synchronization message is wrong. Second, the overall performance relies on the accuracy of the drawn timestamps for all measurements.

When discussing the last point, one has to consider that the jitter can be minimized drastically if the time stamps are taken on a low layer in the protocol stack. This is the reason why for high performance Ethernet implementations of the IEEE 1588 protocol time stamps are taken between the PHY and MAC. Although the current version of the standard does not describe in its annexes any locations in the stack for timestamping in Wireless LAN, it is obvious that for this communication technology also a timestamp should be drawn as soon as a packet arrives at or is sent to the communication medium, i.e., the free air or the physical layer interface.

B. Typical Structures Wireless of LAN Hardware

Figure 1 shows the usually needed components for a wireless receiver/sender. The Synchronization stack (e.g, PTP) sends packets to the network layer (e.g., UDP/IP), which is then interfaced to the MAC layer. The MAC layer, whose main task is the arbitration of the shared medium, hands the packets over to the physical layer, which does the signal shaping and up-conversion to the actual communication frequency in the ISM band.

When the actual available and standardized interfaces are investigated, a different situation has to be faced. As shown in figure 1, access between MAC and physical layer convergence procedure(PLCP)/physical medium dependent(PMD) layer of WLAN is not directly possible. Although older chipsets offer due to their dual chip architecture access to the down-converted signal, the analog signal available at this interface is yet still raised cosine shaped for the use of the further processing.

This point in the signal flow is not accessible in the more recently used single chip implementations. These chipsets only offer an interface to the host system. Between this interface and the receiving antenna, usually a hardware MAC and a software implementation of the physical layer arbitration is employed. In terms of timing, this use of a software stack within the chipset causes additional jitter.

Finally it is important to mention that not the whole implementation of the WLAN chipset is done in pure hardware. For example, it is common that a WLAN chipset has an ARM processor integrated, in order to process the software part of the MAC. The same applies to the IP/UDP part of the system. Both, these processors are typically working asynchronously and are connected via a bus interface as for example a PCI. It should be finally also noted that the firmware on the software MAC is vendor specific and usually not available as open source.

C. Related Work

The problem of synchronization has been already studied by other authors. One major contribution is the work of Mock et al.[6], where the authors present among other work an analysis of improved clock synchronization functionality at application layer over WLAN itself. This built-in clock synchronization is primary used for protocol timing. This is why several features have to be understood in this sense, like the servo algorithm of the clock. This algorithm sets the clock immediate and hard to specific values, which means that the clock can in principle run backwards over some periods. This work also unfortunately does not give any insight in the actual implementation, besides the hint that a chipset vendor enabled the inclusion of the synchronization in the firmware of the WLAN card.

Another investigation conducted by Cooklev et. al. [7] used COTS hardware, where hardware signals where intercepted. The accuracy of the work revealed with these timestamp points a IEEE 1588 synchronization in the microsecond range. Synchronization over WLAN to carry out software-based TDMA has been performed by Guo et. al.[8] where an accuracy of a microsecond has been established by using TSF timer and synchronization of system clocks with TSF. However, this scheme uses additional traffic to implement its TDMA scheme and is not standard compliant.

III. Problem Analysis and Proposed Solution

The description of the related work above shows that a generic timestamping capability for wireless LAN is needed in order to enable potentially all wireless LAN implementations to deploy a precise clock synchronization scheme for applications. This section defines the exact problem and gives an overview of possible solutions. It should be noted that WLAN provides its own MAC layer synchronization using timing synchronization function (TSF). However, it is not of much use in this investigation as TSF can only control activities initiated by WLAN and the intention of this study is to synchronize two systems by matching their system clocks.

A. Problem Description

The goal of this study is to analyze synchronization schemes over WLAN for application layer synchronization protocols like IEEE 1588. It should be noted that IEEE 1588 is typically improved in terms of performance, if hardware timestamping is used. In order to carry out hardware timestamping, modifications inside WLAN chipsets are required which is not possible unless one designs ones own WLAN chipset. Thus, in this study, focus is on investigating means of higher layer synchronization with software support only. Different possibilities for higher layer synchronization, similar to IEEE 1588, have to be discussed and investigations on timestamping accuracy have to be made. Though phenomena related to wireless communications such as interference, packet-loss, retransmissions etc. effect the synchronization accuracy, the accuracy of timestamps will be in limelight in this paper as it is easiest to investigate in this early phase of this study.

B. Proposed Solutions

Three approaches for carrying out application layer synchronization with means of software support and software time stamping is presented in the following and are equipped to work with COTS WLAN devices. These synchronization strategies are discussed below:

- *Replacing Wire with Wireless:* The simplest scheme to implement IEEE 1588 over WLAN is to (partly) replace a link between a master and a slave with a wireless link. This different degrees of handling of IEEE 1588 packets with higher layer timestamps over the wireless path will offer different levels of synchronization accuracy. For example, when the wireless router used in the link implements no Quality of Service (QoS) support, IEEE 1588 packets will be treated as normal traffic and may have to wait for potentially less predictable time inside queues. With QoS, the packets can be pushed into the queues for highest priority packets which ensures a prioritized and thus more traffic independent transfer time of packets to the radio interface. Of course, using hardware timestamps inside wireless routers can help to calculate the sojourn time of packets but time stamping in wireless network interfaces is not yet supported. Nevertheless, this scheme is by far the easiest to implement but compromises on accuracy of synchronization. The drawback of this solution is, that if the wireless network is working at high load, the WLAN devices may have to wait longer times to get access to the channel which will further increase the undeterministic delay to send a packet and hence synchronization scheme in such a scenario has less.
- *Beacon Forging:* This approach deals with an alternate way to synchronize over the wireless channel using software defined QoS inside the wireless network interface. This idea is very similar to the one presented in [9] where the author has underlying application of telecommunication networks over IP. The idea is to exploit beacon messages, which carry out synchronization in the so-called basic service set (BSS). These beacon messages can be used to act as `Sync` messages for higher layers as well. In IEEE 802.11, beacon messages are periodically sent with a fixed (tough software defined) period. Whenever, the timeout to send out beacon messages approaches, the access point suspends arbitration of all queues and starts transmission of the contents of the high priority queue which is reserved only for beacon messages. Thus, in this scheme, beacon messages are to be sent only when a `Sync` message from the higher layer arrives. Hence by controlling the beacons and sending `Sync` messages under the camouflage of beacons, a synchronization similar to IEEE 1588 can be achieved. The timestamp inside the beacon message can be replaced by the timestamp inside the `Sync` messages. Similarly, this timestamp (follow-up messages) can be passed on to the receiver in the consecutive messages before the next beacon.
- *Application Layer Support in WLAN:* To assist higher layer synchronization over WLAN, IEEE 802.11 also provides support with the `MLME-HL-SYNC` primitive. This can be deployed, when the transmitting client sends a multicast `Sync` packet, which contains the higher layer timestamp of the previous `Sync` packet. The receiving client saves the timestamp as soon as it receives a packet.

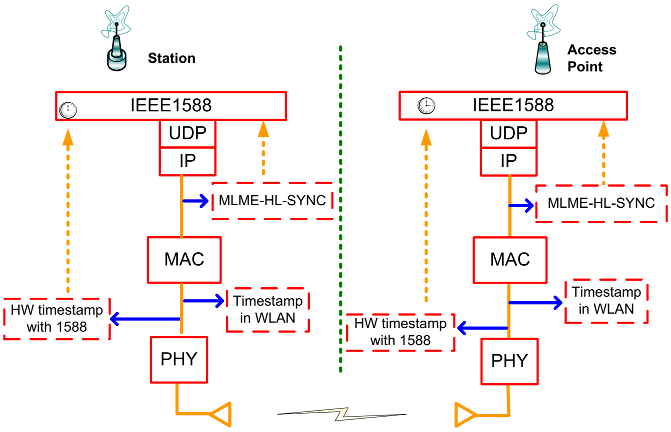

Fig. 2. Application layer clock synchronization support in WLAN and timestamping points in IEEE 1588

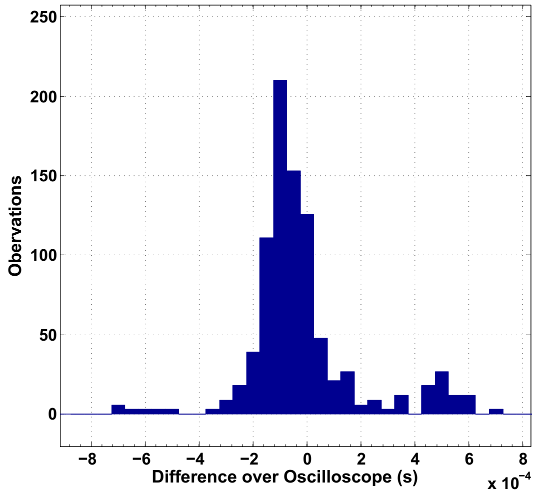

Fig. 3. Difference between successive packets observed on Tx pin of the WLAN card

It can then compare the timestamp for reception of previous `Sync` packet with the timestamp in the currently received packet and can pass it on to application layer. At this layer the usual synchronization algorithms can be used. The timestamping instant of these `Sync` messages coincide with the `PHY-TXSEND.confirm` and `PHY-RXSEND.indication` primitives of the 802.11 MAC. This means that the `MLME-HL-SYNC` functionality provides support of `Sync` and `Follow-Up` messages as used in IEEE 1588. When a transmitting station sends a `Sync` message which carries timestamp of previous packet, it is also providing a `Follow-Up` message which also contains the timestamp of the `Sync` message. Such a scheme is depicted in figure 2. The point "HW timestamp with 1588" shows the point where, similarly to Ethernet, a timestamp can be taken as indicated by IEEE 1588. The position "Time stamp in WLAN" is the point where a WLAN device takes a timestamp for synchronization based on the IEEE 802.11 standard.

This approach can provide the best synchronization accuracy of the all three schemes mentioned above due to the very low timestamping point in the stack. However, a hybrid combination of this scheme with beacon forging can provide higher synchronization as the `Sync` messages will be given appropriate priority and inaccuracies due to varying periods of `Sync` packets will not be present. An additional advantage in the faster processing of `Sync` packets is reduction of the network load because the `Sync` packets will be management packets then and will not be a included in data queues. Therefore, a implementation using this approach, has to be considered.

IV. Platform and Experimental Setup

In this work in progress study only a platform for the third scheme of the last section is discussed as a first step. This can be used as a proof of concept and to highlight improved performance over schemes such as replacing wired links with their wireless counterparts.

A. Platform

The perquisite for this study is to use as far as possible commercial off the shelf wireless LAN hardware. In order to control such a hardware, the driver of the card has to be available in source. This is needed to control the behaviour of the card and thus the MAC layer should accessible by APIs. Currently, not many cards are available with such APIs, or even come with open source drivers. However ath5k driver [10], which support chipsets from Atheros, is completely open source and is written for the Linux operating system. This driver will be studied to investigate achievable accuracy of IEEE 1588 over WLAN. Yet, the question remains how the accuracy of synchronization over WLAN is affected by the accuracy of the timestamps. In order to highlight the impact of timestamping on synchronization, the accuracy of soft- and hardware timestamps are compared. For that, Intersil Prism cards are used in this investigation. The reason to use Intersil cards is that the pin diagram of their chipset is widely available which is not the case for Atheros chipsets. Hence, one can take software timestamps inside the driver whenever the packet leaves the card and can also monitor this instant on the oscilloscope with Intersil chipset but not with Atheros. For periodically sent packets over a long time period, the difference between two successive packets should remain equal to the rate of sending of packets. This rate of sending packets is observed and re- calculated using software timestamps. Finally, both of them are compared. The focus is not to investigate the actual jitter of the timestamps, but to highlight the difference between the accuracy achieved by software- and hardware- timestamps.

B. Experimental Setup and Observations

For a test-setup, a wireless network has been created between an Intersil card (client) and a wireless router (AP) is used. Both stations have been placed close to each other in a direct line of sight. The client-station is forced to send out periodical packets with ICMP ping requests at a rate of 1 packet per second. This period of ping packets has been chosen after a series of experiments. A lower value may coincide with the beacon reception routine of the client while bigger values indicated a higher variance of accuracy of timestamps. Whenever a packet leaves the card, a software timestamp is generated in the kernel space of Linux and is saved in the memory. The same packet is observed on the Tx pin of the card where an activity on the Tx pin provides the closest comparison to a hardware timestamps. To access the Tx pin itself, modifications, similar to the one in [7], have been done to the chipset itself and a connection from the chipset to the oscilloscope itself has been made. As the packets are sent with a period of one second, the oscilloscope should show this difference between successive packets. However, these experiments have been carried out in a laboratory environment, i.e., in the presence of other WLAN stations. Hence, due to medium congestion, some packets are delayed and thus the difference between successive packets is large. Such observations are removed from the final analysis. It should be noted that no packet loss has been observed for the experiments because of very close distance between sender and receiver. The study of Rx timestamps would also have been possible however their analysis would have been more prone to wireless fluctuations and hence focus has been on Tx timestamps.

Figure 3 shows the histogram of difference of successive packets, whose presence has been monitored on the oscilloscope. The graph shows that the jitter of the difference reigns over several μs. It has to be lined out that this is not the actual jitter of the timestamps as the time is measured via a signal generated inside the hardware which can has its own jitter. Also, in IEEE 802.11 each station has a random time before starting its transmission. This is part of the CSMA/CA channel access which is a potential source of variations in the figure. The results obtained from software timestamps are shown in figure 4. The top pane displays the histogram of variations in software timestamps. There is a fixed delay of 1 ms which can be attributed to the delay it takes the hardware to inform the MAC or higher layers that that a packet has been sent. After that there is wide range of time spread over which the timestamps are taken. These variations seems to spread over the range of milliseconds but a close inspection reveals that most of the variations are within a range of microseconds. The percentile distribution inside figure 4 in the bottom pane duly shows that appx. 85 percent of times these variations are within 600 microseconds. This is comparable to the variations of timestamps obtained from the hardware signals itself. Hence, with more sophisticated software time stamping schemes and precise calculations of delays through the software stack, it is possible to get more accurate timestamps (though not more accurate than by hardware means) which in turn lead to better synchronization.

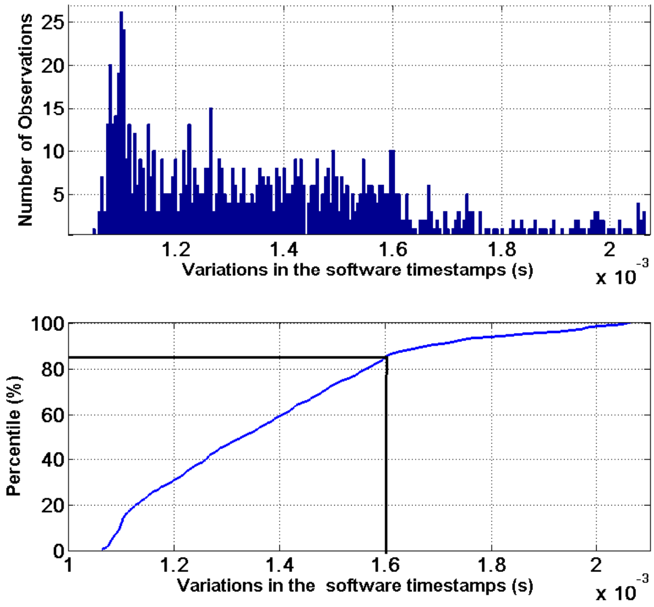

Fig. 4. Difference between successive software timestamps. The top pane shows the histogram of the variations of software timestamps and the bottom pane shows the cumulative distribution of these variations. The thin black line displays that more than 80 percent of variations lie in the range of appx. 600 microseconds.

V. Conclusions

The role of IEEE 1588 for providing precise clock synchronization in real-time networks is important. On the other hand, the increasing tendency of industrial automation being carried out by wireless technology requires a robust clock synchronization scheme which can deal with the challenges of wireless communication. In this study, different possibilities for providing clock synchronization based on 1588 in WLANs have been proposed. IEEE 802.11 also proposes a higher layer scheme for synchronization which can be used to provide IEEE 1588 support in wireless networks. However, with some further modifications to this higher layer schemes, like controlling the beacons of WLAN, can provide more accurate synchronization. The accuracy of synchronization is largely dependent on accuracy of timestamps. As the existing devices do not support hardware timestamps, software timestamps have to be generated which has an adverse affect on the performance of synchronization scheme. The future work of this study includes a prototypical implementation of discussed scheme for providing IEEE 1588 over WLAN with more sophisticated timestamping techniques over existing WLAN devices.

References

[1] J. C. Eidson, “The application of IEEE 1588 to test & measurement systems,” White Paper, December 2005, pp. 9–13.

[2] S. Rodrigues, “IEEE 1588 and synchronous ethernet in telecom,” in *Proceedings of the IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication ISPCS 2007*, G. Gaderer and K. Lee, Eds., 2007, pp. 138–142.

[3] "IEEE standard 802.11 - wireless lan medium access control (mac) and physical layer (phy) specifications. the institute of electrical and electronics engineers, inc., 2007."
[4] J. McKay, Ed., *IEEE 1588 (tm) 2.1 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*. IEEE, March 2006.
[5] D. L. Mills, "Network time protocol (version 3) specification, implementation and analysis," RFC 1305, March 1992.
[6] M. Mock, R. Frings, E. Nett, and S. Trikaliotis, "Clock synchronization for wireless local area networks," in *Proc. 12th Euromicro Conference on Real-Time Systems Euromicro RTS 2000*, 2000, pp. 183–189.
[7] T. Cooklev, J. Eidson, and A. Pakdaman, "An implementation of IEEE 1588 over ieee 802.11b for synchronization of wireless local area network nodes," *IEEE Trans. Instrum. Meas.*, vol. 56, no. 5, pp. 1632–1639, 2007.
[8] F. Guo and T. Chiueh, "Comparison of qos guarantee techniques for voip over ieee802.11 wireless lan," in *Proc. of 15th Annual Multimedia Computing and Networking Conference (MMCN 2008)*, January 2008.
[9] B. Dipankar, "Prototyping a packet scheduler from time driven priority network to 802.11 access network," Master of Science Thesis, KTH, Royal Institute of Technology, November 2006.
[10] ath5k, "the madwifi project." [Online]. Available: http://madwifi-project.org/wiki/About/ath5k