

PHYSICAL LAYER ETHERNET CLOCK SYNCHRONIZATION

Reinhard Exel, Georg Gaderer
Austrian Academy of Sciences

Viktor Kaplan Straße 2, A-2700 Wiener Neustadt, Austria
E-mail: reinhard.exel@oeaw.ac.at, georg.gaderer@oeaw.ac.at

Nikolaus Kerö

Oregano Systems, Mohsgasse 1, 1030 Wien, Austria
E-mail: nikolaus.keroe@oregano.at

Abstract

Clock synchronization is a service widely used in distributed networks to coordinate data acquisition and actions. As the requirement to achieve tighter synchronization accuracy arose, protocols like the Precision Time Protocol introduced hardware timestamping, shifting the point where the timestamp is drawn from the application layer towards the physical layer. However, the spread of the synchronization service between hard- and software increased the complexity of the system and still could not solve the issue with asymmetric transmission delays.

In contrast to existing synchronization systems, this paper proposes a layer 1 clock synchronization system based on hierarchical clock distribution via Ethernet and an IEEE 1588-like clock synchronization protocol operating on a separate data channel orthogonal to the Ethernet's Multilevel Transmission encoding-3 (MLT-3). All clock synchronization-related tasks will be performed by an ASIC attached in parallel to the standard Ethernet PHY. As the ASIC captures the analog data from the line, it is able not only to create nanosecond-accurate timestamps, but also to perform true one-way delay measurements, which are a prerequisite to remove inevitable asymmetry of Ethernet cables. This innovative approach enables one to build lightweight nodes while still achieving unmatched synchronization accuracy.

INTRODUCTION

It is a well-known industrial trend to use proven consumer technology for industrial applications replacing proprietary solutions¹. An example for this is the history of fieldbusses, which evolved from a couple of different vendor-specific physical layer standards to the use of office-proven Ethernet technology as the underlying technology. As Ethernet products are produced in high-volume quantities for the last two decades, fieldbus vendors were able to cut the cost of their products and gain a competitive advantage. While the benefits of off-the-shelf technologies are evident, there are likely inherent drawbacks, because

¹ This work was partly financed by the province of Lower Austria, the European Regional Development Fund, and the FIT-IT project Ætas under contract 825904

the base technology has never been designed for some special requirements like clock synchronization. In the case of Ethernet, a common solution is to use it as a point-to-point bit pipe and shift all necessary functions to higher protocol layers or even the application. While this approach is appropriate for many common cases, for clock synchronization it simply is not.

Synchronization in Ethernet has evolved from a simple time protocol to sophisticated software synchronization schemes like the Network Time Protocol (NTP). NTP is known for its wide use over the Internet and provides a synchronization quality in the range of about 1 millisecond. The next step in terms of accuracy improvement is the introduction of hardware-assisted clock synchronization. With the IEEE 1588 standard, the purely software-based approach was complemented with hardware assistance to timestamp packets in order to exactly measure the ingress and egress times. With this step, the clock synchronization task was split between hard- and software, which resulted in other problems. First, the mapping of the timestamps to the corresponding frames needs a special treatment and tagging of information beyond the layers of the ISO reference model. Second, the control loop has new problems for small synchronization intervals, as the protocol stack running in the operating system is unable to react in a deterministic way. This paper proposes combining the distributed synchronization efforts as a service of the physical layer and, furthermore, use the properties of the baseband signal to achieve an optimal synchronization performance with definitive guarantees and manageable system complexity. If this system is built into a physical layer device (PHY) or in parallel to the PHY, the goal is to allow plug-and-play clock synchronization. That is, a device should automatically connect to the configured master and report its current synchronization status to the application without the necessity to run a synchronization protocol at the host CPU.

In Section 2, the state of the art of clock synchronization in Ethernet is described, together with the challenges to achieve unbiased 1-nanosecond accuracy. Section 3 describes the technical details for design and implementation of a physical layer clock synchronization system based on the requirements identified in Section 2. An analysis of the simulated performance based on different communication parameters is given in Section 4. The final section summarizes the findings and gives an outlook for future work in this area.

CHALLENGES IN ETHERNET CLOCK SYNCHRONIZATION

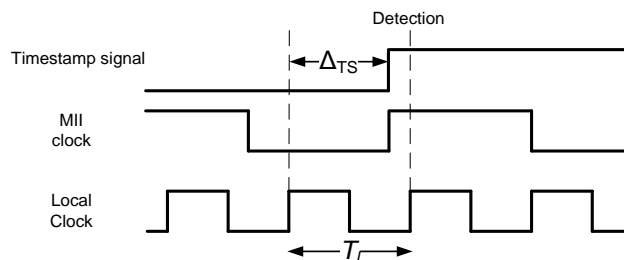
It is well known that the overall synchronization accuracy in packet-oriented networks is defined by few key figures: the stability of the oscillators, the granularity of the timestamps, the timestamp interval, and the loop bandwidth of the control loop. These aspects can be tackled individually, e.g., by oven-controlled oscillators (OCXO), high-resolution hardware timestamping schemes, shortened timestamp intervals, or control optimization. Yet, most enhancements have some kind of drawback like increased cost, complexity, network overhead, or slower clock convergence. This section analyzes these parameters for state-of-the-art IEEE 1588 systems and the challenges for 1-nanosecond-accurate clock synchronization. The findings in this section provide a basis for the proposed physical layer clock synchronization method.

TIMESTAMPING GRANULARITY

The quality of timestamps is an important factor for any synchronization protocol, as they are used to steer the local clock in such a way that it follows the reference time. As software timestamps taken at the application level are affected by the varying processing delay of the protocol stack and the operating system, IEEE 1588 suggests use of hardware timestamps. Although the Precision Time Protocol (PTP) defined in IEEE 1588 can be used on any kind of transport media, the media of choice for industrial, test,

and measurement application is Ethernet, typically 100 Base-TX. Thanks to the layered architecture of Ethernet, the IEEE 802.3 standard has a defined interface between the PHY and Media Access Control (MAC) layer, namely the AUI (Attachment Unit Interface), MII (Media Independent Interface), or GMII (Gigabit Media Independent Interface). This interface can be monitored and whenever a frame of interest is seen, a timestamp can be created and stored. Given that the Ethernet frame can be mapped to the same frame in the synchronization stack, the accurate hardware timestamps allow for much tighter synchronization than pure software approaches.

The detection of a frame of interest is bound to the granularity of the local clock, as every event can only be detected at the next clock edge, no matter at which instant the event occurs within one clock cycle. As Ethernet is an asynchronous network, transmit and receive clocks differ, although they have nominally the same frequency. As the MII receive clock is a local replica of the clock of the communication partner, this clock is not phase-locked to the independent local clock. At some instant (synchronous to the receive clock), a flag is asserted to create a timestamp, as depicted below. However, the timestamp event is detected at the next edge of the local clock. This creates an uncertainty Δ_{TS} of one local clock period T_l . As the timestamp jitter Δ_{TS} is equally distributed between 0 and T_l , the resulting variance is $\sigma^2 = T_l^2/12$ (as defined for a equal distribution with width T_l). The timestamp jitter can be reduced by simply narrowing the local clock period, but clock frequency restrictions limit the efficiency of this approach. The same problem applies to the PHY's transmit side, if the transmit clock is independent of the local clock. Yet the local clock can be used as transmit clock as well, removing this source of timestamp jitter.



Timestamp jitter with independent clocks.

The impact of the granularity of hardware timestamps was analyzed in [1], where the authors used two directly connected IEEE 1588 nodes and altered the timestamp granularity in the hardware between 2 and 16 ns. The clock servo parameters were optimized in a way that the standard deviation of the time error between the nodes became minimal. Two oscillators – one standard crystal oscillator (XO) and one oven-controlled oscillator (OCXO) – were tested as clock source for each node. It is shown that reducing the timestamp granularity from 16 to 8 ns reduces the clock error for intervals ranging from 0.5 to 4 ns to about the half value. However, it can be observed that, for very short synchronization intervals (below 0.25 s), the improvement does virtually not exist. It is worth mentioning that, even with the 128 synchronization messages per second and 2 ns timestamp resolution, it was not possible with the XO to reach the accuracy of the OCXO with just one synchronization message every 8 seconds. Several approaches to exactly measure the time span between two pulses have been adapted from other areas to the field of clock synchronization. The complexity of these approaches reach for just increasing the clock frequency of the timestamping unit, like in [2], over multiple phase shifted clocks up to tapped delay lines [3]. It might be tempting to apply averaging of multiple timestamps in order to decrease the timestamp jitter of each frame. However, averaging will only decrease the jitter if both clocks are not correlated within the time of interest. Unfortunately, this prerequisite cannot be met, as it is likely that clocks at nominally the same frequency (or a multiple thereof) have a constant phase difference for short time

spans (like a frame) and averaging is ineffective, as the timestamps coincide in one spot of the uniform distribution. All these measures operate on a one-shot basis, i.e., they do not exploit the fact that the arrival of the frame is aligned to the receive clock. This fact can be exploited in phase estimation methods like those presented in [4]. In this paper, the authors propose circumventing the short-term correlation by using a timestamping clock which is offset by some percent with respect to the receive clock. Together with frequency offset estimation, the phase estimation method was shown to timestamp with a standard deviation of only 26 ps.

ASYMMETRY

In the PTP, the one-way delay Δ between the master and the slave is calculated by measuring the round trip delay and dividing it by two, under the assumption that the communication path is symmetric. However, in real-world systems, this is hardly ever the case. The reason for asymmetry can be located inside the PHY and the transmission line itself. The asymmetry of the PHY is due to its internal structure. In particular, the generation of the receive clock is of interest for timestamping. The clock from the analog line is recovered using a clock recovery block generating a 125 MHz signal. This signal is divided down by a factor of 5 to 25 MHz in order to drive the MII receive clock. At this point, asymmetry might be introduced into the PHY if the receive clock is generated by simple division of the recovered clock without aligning it to the 4B5B encoded symbols [5]. Hence, such a PHY might generate an additional delay of 0, 8, 16, 24, or 32 ns, depending on the clock edge used for division. As the clock edge is selected during auto negotiation, it remains constant once a link is established and, therefore, cannot be filtered by any means. This issue has been tackled by some manufacturers (e.g., National Semiconductor's DP83640 and DP83848 [5]). Still, the asymmetry caused by differences in the length of cable pairs still remains. Category 5E UTP cables, for instance, are allowed to have a specified delay skew of 0.2 ns/m for frequencies below 100 MHz, resulting in the maximum length of 100 m in a skew of 20 ns or an asymmetry of 10 ns respectively.

COMPLEXITY

While hardware timestamping removes all jitter sources above the physical layer, it introduces additional complexity in the system. First of all, specialized hardware is required typically based on FPGA solutions, making such a system power hungry and expensive. Secondly, as the synchronization is now also dependent on the timely behavior of the PTP stack running on the CPU, the synchronization itself is affected by the reaction time of the CPU, which defines a lower bound for the synchronization interval. For the low-power embedded system used in [1], the clock synchronization could only be improved up to 16 synchronization packets per second, while more packets did not yield improved synchronization. If the node running the synchronization stack does not run a real-time operating system, a strict accuracy bound cannot be defined at all, because it can never be guaranteed that synchronization messages are handled within a certain timely limit. Hence, the quality of synchronization can just be defined by statistics, but no strict bounds can be maintained. The latter problem was partially addressed by the PTP version 2 standard introducing layer 2 clock synchronization, enabling one to run clock synchronization in a dedicated hardware above the MAC. The path from software-to-hardware timestamping to layer 2 clock synchronization logically leads to physical layer clock synchronization.

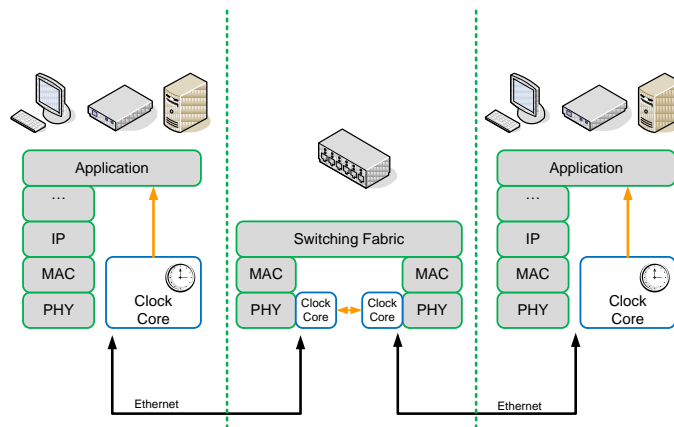
PHYSICAL LAYER CLOCK SYNCHRONIZATION

Clearly, the IEEE 1588 standard was a significant step towards reliable and tight synchronization in Ethernet. However, due to the technical advances in terms of hardware timestamping and optimized synchronization architectures, physical factors like oscillator stability and asymmetry play a dominant

role for the quest to the nanosecond. Clock synchronization on the physical layer is logically the next step. It can be understood as the evolution of clock synchronization moving from a pure software approach to hardware-assisted synchronization over layer 2 clock synchronization to a purely hardware layer 1 clock synchronization. Clock synchronization is seen as a service of the physical layer and maintained within the physical layer IC. On the one hand, synchronization is independent of the system around the PHY IC; on the other hand, synchronization becomes dependent on the physical media. While this might be seen as a possible leap of synchronization between different physical transmission standards, it opens new possibilities in terms of accuracy and simplicity.

SYSTEM CONCEPT

The proposed system is an extension of standard Ethernet communication. All the normal Ethernet data passes over a standard PHY, whereas clock synchronization data are transferred over a different channel on the same media by an orthogonal encoding scheme. This structure differs from the PTP structure by the fact that the standard physical layer device is no longer used for clock synchronization (as depicted in the figure below). This implies that the normal data traffic is not interrupted by clock synchronization packets nor is it necessary to run the protocol stack in the operating system as a background process, as the synchronization is completely shifted to the hardware clock core. The final vision is to include the clock synchronization logic into the PHY (like products from National Semiconductor and Zarlink, who included PTP timestamping into their PHYs).



Physical layer clock synchronization within an Ethernet network.

The stability of the local oscillator of each node in a PTP network is limiting the attainable accuracy, because standard crystal oscillators may drift away multiple nanoseconds between the synchronization packets. One valid solution is to either increase the synchronization rate or by supplying a common clock to all nodes. The latter approach is commonly used in synchronous networks like Synchronous Data Hierarchy (SDH) or by Synchronous Ethernet (SyncE). The objectives of SyncE and PTP differ in that SyncE's aim is to provide frequency lock to all devices, whereas IEEE 1588 attempts to maintain phase lock (minimizing the clock offset) between the master and its slaves.

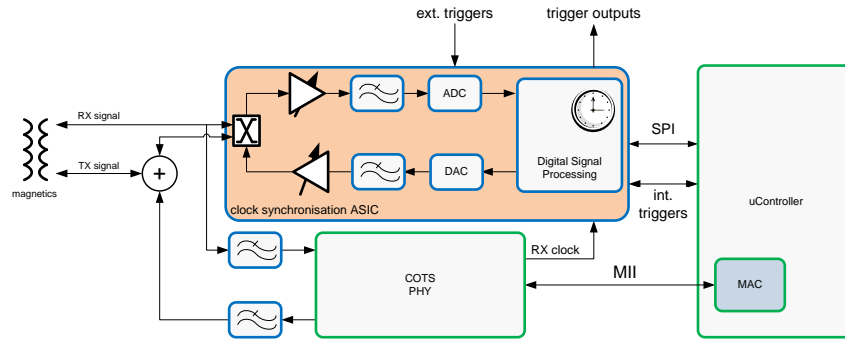
When Adder-Based Clocks (ABCs) are used within a PTP system, slaves can adjust their local clock by changing their rate, i.e., the relative speed of the clock compared to a virtual perfect time. One drawback of such a virtual phase-locked loop is that a regeneration of the master's clock frequency (as needed in telecom applications) is difficult and requires long averaging periods and a high synchronization packet

rate. Since the frequency is the derivative of the phase, the frequency can be regenerated by means of a numerically controlled oscillator (NCO). However, the quantized nature of the phase register within an NCO creates a significant amount of phase noise, making the signal unusable as a clock source for analog circuitry. The proposed system combines the approaches from a frequency-locked system like SyncE with the phase-locked system like PTP: The frequency distribution will be maintained by the physical layer (as in SyncE) based on clock recovery of the data, whereas the synchronization (the phase alignment) will be established by accurate link delay measurements based on synchronization frames. If the frequency $f(t)$ is shared among all slaves, bringing the phase to all nodes results in estimating the initial phase $\phi(t_0)$ as $\phi_{est}(t) = \phi(t_0) + \sum_{n=0} f(t_0 + n\Delta t)\Delta t$.

From the observer's point of view, estimating the phase offset is simple. At one arbitrary instant, the observer fetches the actual clock reading from the master and the slaves and tells the slaves to advance their clocks once by the difference of the readings with respect to the master. Given that the transmission delays never changes and the local PLL stays locked with the frequency provided from the receive clock of the PHY, the system stays synchronized forever. Within the system, though, the estimation of $\phi(t_0)$ requires the knowledge of the sum of the delays between the master and the slave in both directions individually, as asymmetry is unavoidable. Even in the case that the initial phase was set correctly (i.e., master and slave had exactly the same notion of time), a real-world system does not necessarily stay unbiased forever. As cables are affected by environmental effects, like diurnal temperature variation and aging, these effects may change the channel impulse response and propagation delay. The receiving PHY will accommodate for the altered channel, e.g., by using a higher amplification and different equalizer coefficients. This shifts the phase estimation, as amplifiers having a limited gain-bandwidth product create more phase shift for higher amplification. Hence, the delay measurements between master and slave and reverse have to be repeated periodically.

PHY ARCHITECTURE

The proposed physical layer architecture is based on the concept that there are two orthogonal communication channels on the same media: One for the standard Ethernet data traffic, and one for clock synchronization only. This concept is favored, as it is not economically reasonable to build a complete 100 Base-TX PHY from scratch and modify it in a way to still remain standard compliant while adapting it to the needs of clock synchronization. The proposed layout, consisting of a commercial-off-the-shelf PHY together with the dedicated clock synchronization ASIC, is shown below. While the COTS PHY deals with everything that is related to standard Ethernet traffic, the ASIC is the timekeeping device. It is able to timestamp every received frame and, in addition to that, perform true one-way delay measurements. Apart from the communication blocks, the ASIC holds a microcontroller which processes the clock synchronization protocol exchanged between the ASICs, either via Ethernet or via a serial interface in the case of multiple ASICs on a switch PCB. Given that SyncE clock distribution concept is employed, the data bandwidth requirement of synchronization and delay measurement is very low, as it is just necessary to compensate for environmental effects, but not for the instability of local clocks.



Proposed physical layer architecture using DSSS.

All synchronization information (like link delays, status) is transmitted using a modulation method which is orthogonal to the MLT-3 encoding of Ethernet, reducing the cross-interference level. Besides this requirement, the modulation should allow for drawing timestamps with a high accuracy; hence, the bandwidth of the signal should be high. It is proposed to use Direct Sequence Spread Spectrum (DSSS) modulation, which spreads its signal energy over a wide spectrum, making it appear as white noise to other receivers like COTS PHYs. The DSSS modulation “xors” each data bit by a pseudorandom noise (PRN) sequence clocked with the much higher chip rate, making it immune against narrowband interference. For this application, the interferer is the MLT-3 code of Ethernet, and vice versa. The degree of interference depends on the ratio of the spectral power of the DSSS signal with respect to the MLT-3 signal of Ethernet. As transmission and reception within the same media may happen at the same time, the DSSS signal power must be low enough, taking the near-far problem (cable attenuation) of Code Division Multiple Access (CDMA) transmission into account. One option is to reduce the spectral power of the DSSS by a very long spread codes to a level where interference is low. As the required data bandwidth is very low (a few bits per second), very long spread codes impose no limitation. The PRNs for such a spread code can be generated by linear feedback shift registers (LFSRs). The second option is to shift the DSSS signal to a higher frequency band, building a frequency division multiplex (FDM) system. In the latter case, the MLT-3 and DSSS signal do not suffer the near-far problem, as the signal can be separated by filters.

TIMESTAMPING

The use of DSSS modulation has one inherent advantage over narrowband data communication with the same data rate, namely its pulse compression abilities. When the transmitted signal is correlated with the locally replicated PRN spread code, its autocorrelation function has significant peaks and low sidelobes which enable the estimation of the time delay between the received signal and the local replica with an accuracy below the chip rate. This fact is used by many positioning systems like GPS, Galileo, or radar applications [6]. In Ethernet, the pulse compression ability can be used in a similar way to create highly accurate timestamps. Consider the following example: The clock master transmits a frame with known format using DSSS modulation, and the slave needs to timestamp the frame at a predefined point within the frame (epoch). For the DSSS receiver in the slave, there is some degree of freedom as to how to implement the receiver logic. One approach is called hybrid receiver, where the receiver adjusts the phase of its sampling clock in a way that it exactly coincides with the transmitted chips of the DSSS [7]. Hence, the sampling clock is a regenerated version of the transmitter’s clock, which is typically maintained via a Costas loop or squaring loop. Note that the hybrid receiver architecture is also used by 100 MBit/s Ethernet PHYs with MII, where the MII’s receive clock is a replica of transmitter’s clock. As discussed in the previous section, the presence of two clocks (receive and local clock) within a PHY

requires at least a clock transition from the receive clock to the local clock, which in turn is a source of timestamp jitter.

This clock transition can be avoided if the sampler is driven by the local clock and the extraction of the chips is done digitally by interpolation of the sampled analog input data. Such a receiver is termed digital receiver, as the chip synchronization is not done in an analog way by adjusting the frequency of the sampler's clock. The squaring synchronizer is one of many possible realizations of a synchronizer estimating the relationship between the received signal and the local sampling clock. It belongs to the class of non-data-aided synchronizers generating a spectral line at the symbol rate and multiples of it. It can be shown that the transitions within the DSSS signal generate a cyclostationary process at the output of the squarer [7]. The spectral line at the chip frequency contains an unbiased estimate for the chip timing ε (with respect to the sampling clock). It can be extracted by calculating the argument of the Fourier coefficient c_1 by

$$\varepsilon = -\frac{1}{2\pi} \arg \left(\sum_{m=0}^{LN-1} |x_{m+1}|^2 e^{-j2\pi m/N} \right),$$

with N the number of samples per chip, and L the length of the averaging frame and x the sampled data. As the chip timing is available as a numerical value, the uncertainty connected to clock transitions is avoided. Theoretically, this would allow for timestamps with infinite resolution if the averaging window is extended to infinity and the noise is zero-mean. Given that the probability density function (pdf) of sampled data is known, the timestamping variance can be calculated by the Cramér-Rao Lower Bound, the inverse of the Fisher matrix [8]. In practice, the sample's pdf is not zero-mean and, therefore, the timestamps are biased.

ASYMMETRY COMPENSATION

Asymmetry caused by cables and PHY ICs has been identified as one of shortcomings of Ethernet synchronization, as it is impossible to measure the same cable pair in both directions. In case of the proposed layer 1 clock synchronization scheme, true one-way measurements are possible. If the DSSS spread code is chosen long enough, the required transmission power is so low that transmission and reception can be performed simultaneously without the requirement to use complex echo cancellation methods. Hence, each transmission pair can be measured individually and the asymmetry can be fully compensated. As the frequency is distributed independently even during the delay measurement, it does not matter how long the measurement takes, as there is no oscillator that may drift away during the measurement.

SIMULATED PERFORMANCE

The design of a layer 1 clock synchronization system depends on the communication parameters of the system. This section provides simulated results on the basis of the specification of the channel (the Ethernet cabling) with the known characteristics of DSSS. The IEEE 802.3 Clause 25 (known as 100 Base-TX) standard transmits its signal with a symbol rate of 125 MS/s. Using the three level MLT-3 encoding, the effective bandwidth is reduced to 31.25 MHz. 100 Base-TX requires at least Category 5 cables which have a specified frequency range of 100 MHz, a maximum propagation delay of 548 ns, a delay skew between transmission pairs of 50 ns, and a maximum attenuation of 24 dB. Besides these communication parameters, bounds for the near-end cross-talk, power-sum cross-talk, attenuation-to-

cross-talk ratio, and so on are defined as well.

DSSS AND MLT-3 WITHIN THE SAME FREQUENCY BAND

If the DSSS-modulated clock synchronization channel is put into the same frequency band up to 31.25 MHz as MLT-3, the signals interfere with each other. Considering that the maximum attenuation is defined by 24 dB in one direction, in case of simultaneous DSSS transmission and MLT-3 reception, the transmitted DSSS signal power must be lower than the received MLT-3 signal at -24 dB. If one assumes that the COTS PHY needs an additional Signal-to-Noise-plus-Interference Ratio (SNIR) of 10 dB, the DSSS spectral transmit power must be -34 dB lower than the MLT-3 transmit power. On the other hand, if the DSSS receiver requires 10 dB SNIR and the cable attenuates the signal by 24 dB in the other direction, the DSSS must have a total process gain of 68 dB. As the process gain of DSSS is a linear function of the length of the PRN sequence, it must be at least $10^{68/20} = 2511$ chips long in order to achieve the required process gain under the assumption that MLT-3 and PRN sequence are fully orthogonal. As LFSRs can only generate sequences of length $2^n - 1$ (with n the length of the shift register), a 12-bit shift register can be used for generating a 4095-chip-long PRN sequence. The length of the sequence can be increased to even higher values. However, for very long PRN spread codes, the acquisition and synchronization is time and resource consuming. For the presented numbers, 3815 bit/s are transmitted and 4095 possible lock-in positions are available. Using a single correlator (with a $\frac{1}{2}$ chip spacing), it will take less than 2 seconds to acquire lock to the spread code.

With 31.25 MHz of allocated spectrum, the length of a chip is 64 ns. Although the timestamp estimation using a digital receiver can be arbitrarily accurate in case of a zero-mean noise, a timestamp accuracy of $\pm 1/10$ of the chip period (± 6.4 ns) can be expected if the channel impulse response (CIR) is subject to distortions and dispersion. These are estimated values, as the variability of the CIR was not under investigation for this paper. Given that the CIR is taken into account, the multipath mitigation methods (e.g., narrow strobe correlators, as presented in [9]) can be applied, reducing the timestamp bias. Considering that Category 5/5E and 6 cables are specified with a delay skew of 50 ns, a reduction of the bias down to 6 ns already mitigates the majority of the asymmetry.

DSSS AND MLT-3 IN DIFFERENT FREQUENCY BANDS

When the DSSS signal is not within the same frequency range as the MLT-3 encoding of Ethernet, the FDM enables the separation of DSSS and MLT-3 by the use of sharp filters. Hence, either the transmission power of the DSSS signal can be decreased or the spread code can be shortened in order to increase the data rate. In addition, if a wider bandwidth is available, the DSSS spread factor can be increased, resulting in lower chip periods and practically better timestamp accuracy. Yet, covering a broad spectrum requires equalization to accommodate to the channel impulse response, which increases complexity in the receiving ASIC. Covering the complete 250 MHz band of a Category 6 cable (while using the lower band for MLT-3, the higher one for DSSS) would yield a chip period of about 9 ns and, therefore, a practical timestamp accuracy of even below 1 ns.

CONCLUSION AND FURTHER WORK

This paper shows the difficulties of reaching a 1-nanosecond-accurate clock synchronization in Ethernet, as the time offsets are difficult to remove. Even state-of-the-art highly accurate PTP Ethernet equipment cannot reach the nanosecond bound, as factors like timestamp granularity, oscillator stability, complexity, and the inability to measure asymmetric delays hold it back. The communication theory, on the other hand, confirms that this bound is within reach, if a synchronization system takes advantage of the

parameters of the physical layer. It is proposed to attach a dedicated clock synchronization ASIC in parallel to a COTS PHY and let it perform all clock synchronization-related tasks. Using a DSSS modulation for the low data rate synchronization channel, it is possible to compensate for asymmetries with true one-way measurements, while appearing as white Gaussian noise to the PHY. While the theoretical bound is a matter of averaging, the practical bound can be found in about $\pm 1/10$ of the chip period. Depending on the bandwidth used, simulation shows that two nodes can be synchronized so as to have an offset of below 1 ns.

On the other hand, one has to be aware that such accuracy can only be reached if, in a synchronization system, these enhanced PHYs are in use. Nevertheless, it is easy to obtain, for example, in connection with commercially available IEEE 1588 hardware, standard-compliant synchronization, with reduced accuracy. For the case of the usage of the proposed approach, it is believed that, for niche applications, the solution has significant benefits in terms of accuracy and complexity. The next steps are to prove the concept by prototype hardware and to design a protocol extension to IEEE 1588 to use all features of the ASIC. The final goal can be seen as the integration of clock synchronization circuit into a standard Ethernet PHY IC.

REFERENCES

- [1] P. Loschmidt, R. Exel, A. Nagy, and G. Gaderer, 2008, "*Limits of Synchronization Accuracy Using Hardware Support in IEEE 1588*," in Proceedings of the 2008 IEEE Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS), 24-26 September 2008, Ann Arbor, Michigan, USA (IEEE), pp. 12-16.
- [2] R. Exel and G. Gaderer, 2008, "*Boundaries of Ethernet Layer 2 Hardware Timestamping*," in Proceedings of the 2008 IEEE International Workshop on Factory Communication Systems, 20-23 May 2008, Dresden, Germany (IEEE), pp. 255-258.
- [3] R. Szplet, J. Kalisz, and R. Szymanowski, 2000, "*Interpolating time counter with 100 ps resolution on a single FPGA device*," **IEEE Transactions on Instrumentation and Measurements**, **49**, 879-883
- [4] R. Exel and P. Loschmidt, 2009, "*High Accurate Timestamping by Phase and Frequency Estimation*," in Proceedings of the 2009 IEEE Symposium on Precision Clock Synchronization for Measurement, Control, and Communication, 12-16 October 2009, Brescia, Italy (IEEE), pp. 126-131.
- [5] D. Rosselot, 2006, "*DP83848 and DP83849 100Mb Data Latency*," Application Note AN-1507 (National Semiconductor), pp. 1-13.
- [6] E. Kaplan and C. Hegarty (editors), 2006, **Understanding GPS: Principles and Applications** (2nd edition; Artech House, Norwood, Massachusetts).
- [7] H. Meyr, R. Moeneclaey, and S. Fechtel, 1998, **Digital Communication Receivers** (J. G. Proakis, editor; John Wiley & Sons, New York).
- [8] J. G. Proakis, 2006, **Digital Communications** (4th edition; McGraw-Hill Science, New York).

- [9] J.-M. Sleewaegen and F. Boon, 2001, “*Mitigating Short-Delay Multipath: A Promising New Technique,*” in Proceedings of the ION-GPS 2001 Meeting, 11-14 September 2001, Salt Lake City, Utah, USA (Institute of Navigation, Alexandria, Virginia), pp. 204-213.

