

L-PTP: a Novel Clock Synchronization Protocol for Powerline Networks

L. Lo Bello, A. Raucea, G. Patti, O. Mirabella

Department of Electrical, Electronic and Computer Engineering – DIEEI

University of Catania - Italy

{lucia.lobello, antonino.raucea, orazio.mirabella}@dieei.unict.it, gaetanopatti@pgprojects.it

Abstract

This paper proposes a Lightweight version of the Precision Time Protocol, called L-PTP, for implementation on Powerline communication systems. The aim of this novel protocol is to reduce the number of messages exchanged to achieve synchronization, without penalizing its quality. L-PTP introduces several modifications to the original PTP and can be implemented on COTS devices. The paper focuses on the protocol and on the benefits of its combination with a virtual clock computation performed by a dynamic clock synchronization algorithm.

1. Introduction

Powerline networks (PLNs) [1] are natural candidates for smart-grids and energy management systems, as they allow for data transmission over already deployed power lines without the need to build new network infrastructures, thus reducing additional costs. As narrowband Powerline networks offer a low data rate, for the sake of efficiency, the packet size and the number of exchanged messages should be minimized.

PLNs to be used in smart grids and energy management systems also call for customized clock synchronization protocols. A common view of time is required, for instance, for the synchronization of voltage and current phasors of the distributed energy resources of a micro-grid working in island mode (autonomous), as a timing reference to dynamically adjust the power line frequency [2] is needed. According to the European CENELEC EN 50160 regulations, voltage frequency must be $50\text{Hz} \pm 1\%$ (i.e., from 49.5Hz to 50.5Hz) which corresponds to a clock accuracy of about 198 microseconds. Clock synchronization is also needed to support several applications as for example smart metering applications, which require that sensors values be acquired at the same time.

A suitable candidate for synchronization in PLNs is the IEEE 1588 standard, that defines a protocol known as the Precision Time Protocol (PTP) which provides precision clock synchronization [3] and it is widely used for its accuracy and simplicity. Implementations of IEEE 1588 on Powerline networks usually [4] use

special hardware to timestamp packets in order to minimize the delay introduced by the different clocks in read/store operations. This approach improves the synchronization accuracy, but it is not cost-effective as custom hardware is required.

In this work we propose a lightweight version of the Precision Time Protocol, called L-PTP, for implementation on bus networks based on Powerline communication. The deployment costs has been reduced by using existing hardware, i.e. commercially available Off-The-Shelf (COTS) narrowband embedded devices. As COTS embedded devices feature low data rate (2.4Kbps/s) and bandwidth saving is a mandatory requirement, the L-PTP here proposed aims to reduce as much as possible the number of messages exchanged to achieve synchronization, while still achieving an accurate synchronization.

Moreover, L-PTP has been combined with an approach based on Dynamic Continuous Clock Synchronization (DCCS) [5] that, through the calculation of a virtual clock, reduces the rate of synchronization cycles, thus allowing for an efficient use of the limited bandwidth the embedded devices are equipped with. The L-PTP protocol is well suited for small Powerline networks (e.g., a cluster of generators in a micro-grid) where all nodes are visible to each other, so that broadcast communication can occur.

In the following, we describe the L-PTP and its implementation on COTS devices, and present a preliminary performance evaluation in two flavors, i.e., L-PTP alone or combined with a DCCS-based virtual clock computation. Finally, we outline future work.

2. The Lightweight Precision Time Protocol

To understand the novel L-PTP here proposed, a brief overview on the original PTP as defined in the IEEE 1588 standard is needed. In PTP, the synchronization process is divided into two phases, as shown in Fig.1. In the first phase, the time difference (offset) between the master and a slave is calculated as follows. The master sends a *sync* message and takes the transmission timestamp t_1 . After reception of the *sync* message, the slave stores the timestamp of the message arrival time t_2 .

Later, the master inserts the value t_1 into a *follow_up*

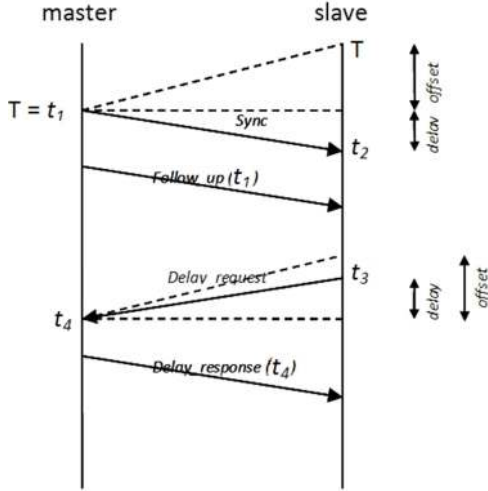


Figure 1. PTP Synchronization cycle

message and sends it to the slave.

$$t_2 - t_1 = \text{offset} + \text{delay} \quad (1)$$

In the second phase, the slave stores the timestamp at t_3 while it is sending a *delay_request* message. After, the master sends a *delay_response* containing the arrival timestamp t_4 of the previous *delay_request*, so the slave can calculate the value in (2).

$$t_4 - t_3 = \text{delay} - \text{offset} \quad (2)$$

Finally, the slave calculates the offset and delay as in (3) and (4):

$$\text{delay} = t_2 - t_1 - \text{offset} \quad (3)$$

$$\text{offset} = \frac{t_2 - t_1 - (t_4 - t_3)}{2} \quad (4)$$

Through the second message set the slave can compute the Propagation time and take it into account during the synchronization. One main requirement of 1588 is that communication delays must be symmetrical, i.e., packets must experience the same delay in both directions. In a simple bus network with short propagation delays, like the one considered here, this assumption holds.

The PTP protocol introduces a non-negligible overhead for bandwidth-constrained networks, as it requires four messages for each synchronization cycle. In the case of PLNs with a low bit-rate (e.g., 2.4 kbps, the value for the devices used in our testbed), this represents a significant percentage of bandwidth which strongly limits the network scalability.

Another critical point of PTP is the tradeoff between the synchronization repetition rate and bandwidth consumption, as a high repetition rate provides higher accuracy, but requires more bandwidth.

The L_PTP approach proposed in this paper introduces several modifications to the PTP that aim to save bandwidth by reducing the number of messages exchanged and the synchronization rate, as shown in Fig.2.

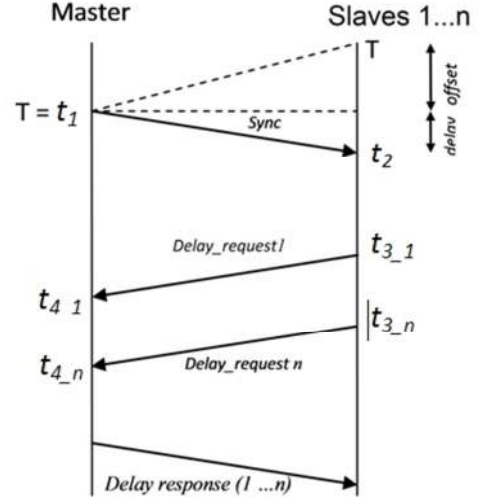


Figure 2. Sequence of messages in a multi-slave synchronization.

The first modification is the elimination of the *follow_up* message, that reduces to three the number of messages required for each synchronization cycle.

This modification is made possible by the low transmission bit rate of PLNs, which enables the master to read its clock right after the reception of the interrupt raised by the transmission of the *sync* frame preamble and insert it on-the-fly into a suitable field of the outgoing frame.

The second modification introduced in L-PTP is a multi-slave synchronization that allows us to significantly lower the number of messages required to synchronize multiple slaves with the same master.

After the transmission of the *sync* message from the master, each slave (i) in turn will send a *delay_request* message and store the time t_{3_i} when the message was sent. In order to allow slaves to transmit in an ordered fashion without collisions, a suitable access policy (e.g., a virtual token passing) can be used to solve the problem with a low overhead. Upon reception of each *delay_request* message, the master will store the time t_{4_i} when the request message was received and the corresponding source slave address. After the reception of the *delay_request* messages from all the n slaves, the master will assemble a frame containing all the $(t_{4_i}, \text{source slave}_i)$ pairs and send the *delay_response (1..n)* frame. This way, upon reception of the *delay_response (1..n)* frame, each slave will only read the relevant time of interest and will be able to calculate the offset and delay values to apply to synchronize with the master.

This feature of L-PTP is beneficial to the system scalability, as any new slave to be synchronized will only entail the transmission of a single additional frame, i.e., the *delay_request (n+1)*, while the only change in the *delay_response* frame will be one more pair $(t_{4_i}, \text{source slave}_i)$ for the new slave. The maximum number of slaves which can be synchronized by this approach depends on the bandwidth available and the maximum

frame length allowed by the specific communication protocol.

Moreover, the performance provided by L-PTP can be improved by lengthening the time interval between two consecutive synchronizations by means of a skew estimation locally performed by each slave node through the Dynamic Continuous Clock Synchronization (DCCS) protocol.

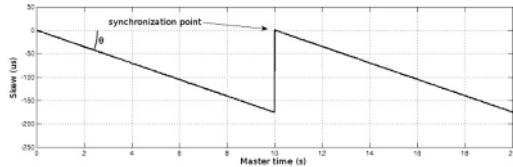


Figure 3. Growth of the skew between two consecutive synchronizations.

- The rationale for introducing the DCCS protocol comes from Fig.3, that shows how the skew between the master and a generic slave node varies over time. After each synchronization, the slave starts drifting again, but with an almost constant angle θ . The measurement of this slope can be effectively exploited by a slave to infer the master clock value and correct the skew accordingly. Using two consecutive pairs of these values, slave S can compute the cotangent of angle θ as a drift correction coefficient. Applying the methods of the right triangles geometry, from the knowledge of the amount of time elapsed since the last synchronization (i.e. the horizontal cathetus $C_s(t) - C_s(t_i)$), it is possible to calculate the theoretical difference between the master clock and the slave clock at any time, and correct it. This way, any slave can locally estimate, at any time, the skew between its clock and the master clock and thus may provide the local application with an instantaneous *virtual clock* value.

3. Testbed Implementation Issues

The behavior of the synchronization algorithm is influenced by several conditions which include the network topology and the hardware used. Concerning the topology, a bidirectional bus structure was used, where each node is visible to all the others. In a real system, this condition is achieved by limiting the length of the line and avoiding the presence of low impedance loads. As regards the hardware and the physical layer used here, our testbed consists of some Powerline devices based on the Konnex standard [6]

Each node is implemented on the STEVAL-IHP003V1 board, which integrates within a single board one ST7540 power line modem, with configurable bit rates up to 4800 bps, and one STM32F103C8 microprocessor based on an ARM Cortex-M3 architecture, with a 72MHz clock, 64 kB of flash memory and 20 kB of RAM.

These are open devices, so it has been possible to implement several changes in the original software. For example, we have defined a physical layer frame in which the master clock value is entered right after being read, before shipping the outgoing frame to the medium.

The accuracy of the synchronization algorithm in the testbed considered in this paper is limited by the resolution of the microcontroller clock, the encoding technique and the transmission rate. The resolution of the clock produces an error between two distinct readings that is at least equal to the resolution itself, that is 10 μ s in this case.

The encoding and the transmission rate are correlated: The ST7540 modem uses an FSK (frequency shift keying) encoding with two reference frequencies given by the relation $132\text{kHz} \pm \Delta/2$, where Δ is the baud rate, set to 2400 bps, which imposes a bit time on the channel of about 415 μ s. The transmission starts with a preamble, alternating a sequence of 1 and 0 for a total of 4 bytes, this way giving rise to a square wave with a period that is twice the bit time. This wave represents both a wake-up signal and a synchronization reference for the receiving modem. Each modem follows this signal, waiting for the sequence indicating the end of the preamble and the beginning of the payload. Since the synchronization quality is influenced by the precision of the time reference acquisition, the evaluation of these references is very important. To this aim, we measured the latency between two specific points in time, i.e. the detection of the end of the frame header generated by the master and the reception of such a header by two close slaves. The end of the frame header during both transmission and reception generates in the microcontrollers, both at the master and the slave side, an interrupt that represents the synchronization signal in the proposed algorithm. After the reception of the interrupt, the master stores the clock and sends its value on the channel.

4. Preliminary Performance evaluation

All measurements were made with a master and two slaves located at short distance (one meter apart from each other) in order to minimize the noise effect, so as to find the performance that the approach is able to provide in the most favorable case. In all measurements the L-PTP was adopted to synchronize two slaves through a single message sequence, as we described in Sect.2.

Fig.4 shows the behavior of L-PTP when the synchronization period is fixed to 10 s. The Y axis shows the skew between the master and the two slaves.

We can observe that slave 2 has a higher drift than slave 1 and that in the interval between two synchronizations the skew can approach 300 μ s, whereas for slave 1 the skew approaches is almost 200 μ s at maximum. The difference between the two slaves depends only on the different drift of the quartz oscillators, but after each synchronization we observe

that the skew practically goes to zero in both slaves and this confirms the good behaviour of the synchronization algorithm. In the case a lower value of the maximum skew has to be provided, the synchronization interval should be reduced (e.g., to 5 s or less), but at the cost of a higher bandwidth usage.

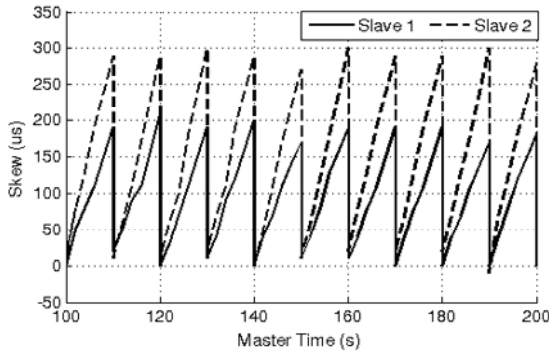


Figure 4. Measurement of the skew for two slaves with different clock drift

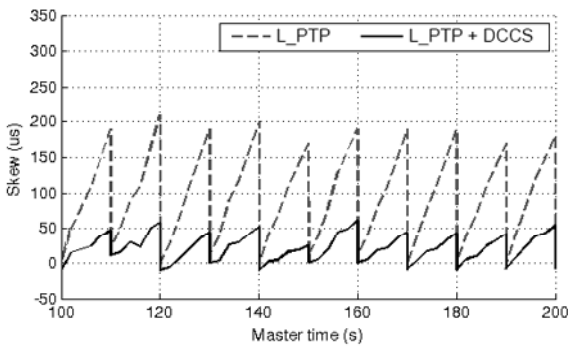


Figure 5. Comparison between the skew obtained using either the L-PTP alone or L-PTP with DCCS

A solution to this problem comes from the combination of the L-PTP with the DCCS algorithm. The results obtained with this combined version are shown in Fig.5, where we can see that the skew between the master and the slave (in this case we have considered only slave 1) is much lower if compared to the values obtained using the L-PTP alone. Even if the synchronization period is maintained to 10s, the skew is always lower than 50 μ s.

This result depends on the DCCS algorithm, which is able to compensate for the clock drift, provided it is constant in time. As this condition normally holds, the DCCS virtual clock is able to provide an accurate clock even in the case of long synchronization intervals.

These results are further confirmed by the test shown in Fig.6, where the synchronization interval has been increased up to 200s. The maximum skew for slave 1 is still very low (about 50 μ s) even if the synchronization intervals are very long. Similar results have been

obtained for slave 2. Using the L-PTP alone, according to the trend shown in Fig.3, the skew would reach the value of several milliseconds.

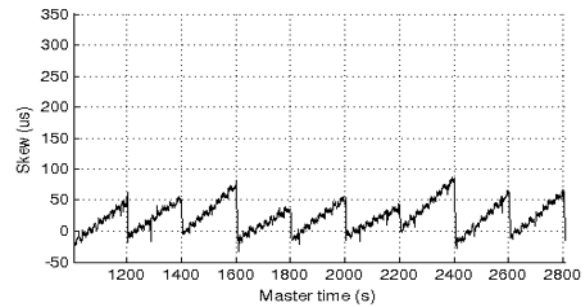


Figure 6. Skew in slave 1 with a 200s synchronization interval

5. Conclusions and Future Work

The results obtained are promising as, even with very long synchronization repetition intervals, the synchronization quality obtained by L-PTP is very good. Future work will deal with a more extended performance evaluation, with a greater number of slaves, in order to assess the possible influence of their number on the quality of the synchronization obtained. In addition, the case when not all the slaves are directly visible from the master and it is necessary to use multi-hop communication techniques will also be investigated. In this case, the influence of relevant parameters will be analyzed and an accurate tuning will be performed in order to limit the performance degradation.

References

- [1] Crolla, P., Tonello, A., Weiss, S., Burt, G., "Evaluation of narrowband power line communications on a smart grid testbed", *2nd IEEE PES ISGT Europe*, pp. 1-5, Manchester (UK), 5-7 December 2011.
- [2] Costabeber A., Tenti P., Mattavelli P., "Surround Control of Distributed Energy Resources in Micro-Grids", *Proc. of the IEEE Int. Conf. on Sustainable Energy Technologies*, pp. 1-6, Kandy, December 2010.
- [3] IEEE. Std. 1588-2008, "Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control System", 2008.
- [4] Gaderer, G., Sauter, T. and Bumiller, G., "Clock synchronization in powerline networks", *Proc. of the 2005 IEEE Int. Symp. on Power Line Communications and Its Applications*, pp. 71 – 75, 6-8 April 2005.
- [5] Mirabella, O., Brischetto, M., Rauceca, A., Sindoni, P., "Dynamic Continuous clock Synchronization for IEEE 802.15.4 based sensor networks", *Proc. of IEEE IECON 2008*, 2438-2444, Orlando, FL., Nov. 2008.
- [6] CENELEC EN 50090 "Konnex: European Standard for Home and Building Electronic Systems (HBES)", 19