

Time Signals Converging within Cyber-Physical Systems*

Marc Weiss

Time and Frequency Division, NIST
Boulder, CO, USA

E-mail: mweiss@nist.gov

Sundeep Chandhoke

Research & Development
National Instruments, Austin, TX,
USA

Hugh Melvin

National University of Ireland
Galway, Ireland

Abstract— Time is central to predicting, measuring and controlling properties of the physical world, and is one of the most important constraints distinguishing Cyber-Physical Systems (CPS) from distributed computing in general. However, mixing the cyber and the physical presents a fundamental challenge, since computers and communications systems have abstracted away the physical layer and timing is fundamentally a physical signal. While such abstractions have yielded significant benefits, time has been a casualty. CPS used in industry today achieve time-awareness by making use of time-aware fieldbuses and devices with specialized proprietary software. However, this approach has proved restrictive in both the topologies achievable and the scalability of networks beyond a certain size. The new era of the Internet-of-Things and the Industrial Internet is paving the way for convergence, where time needs to be an integral part of the cyber, making integration of cyber and physical seamless. However, this requires successful research in a number of different areas.

The National Institute of Standards and Technology (NIST) has formed a CPS Public Working Group (PWG), with members from global industry, academia and government. This CPS PWG is tasked with creating a set of frameworks and reference architectures for CPS, to promote proper function and interoperability. Public documents from this effort will soon be available. We discuss the timing section of the CPS PWG document and focus on the status of challenges and efforts to integrate time-sensitive with best-effort processes in CPS nodes and the networks that connect them.

Keywords—*Cyber-physical systems, time-stamp, Internet of Things, time-sensitive networks*

I. INTRODUCTION

We stand at the advent of a revolutionary new economy fueled by the global Internet of Everything, IoE, a combination of the traditional telecom system with its growing need for wireless technology, and the emerging Internet of Things, IoT, [1] [2], including Machine-to-Machine (M2M) technology [3]. Cisco, among others, predicts that there will be a trillion endpoints connected to the Internet by 2022, with \$14.4 trillion in value at stake [4]. General Electric, GE, says “about 46% of the global economy or \$32.3 trillion in global output can

benefit from the Industrial Internet” [5]. The National Institute of Standards and Technology (NIST) has formed a Cyber-Physical Systems (CPS) Public Working Group (PWG) to bring together experts to help define and shape key aspects of CPS, and to create a framework and reference architectures to encourage interoperability and appropriate designs [6]. One fundamental enabler of this revolution will be a better marriage of timing signals and data that otherwise will limit this growth. Currently, optimal use of data in computing and networking is anathema to optimal use of timing signals. Computer hardware, software and networking all isolate timing processes, allowing the data to be processed with maximum efficiency due in part to asynchrony. Yet, coordination of processes, time-stamping of events, latency measurement and control, and optimal use of precious spectrum are enabled by timing.

Timing is critical for the future development and improvements to several current high value applications. For example, smart transportation involving the exchange of information between vehicles, highways, and perhaps civil authorities will depend on a robust ubiquitous timing system to ensure the availability and integrity of the data. Similar requirements are found in the operation of the power grid, especially now that wind farms, solar arrays and the like, which will require different control strategies, are becoming an important part of the system. Medical applications such as tele-surgery, and regulating fairness in financial systems are other important examples.

II. NIST CPS PUBLIC WORKING GROUP

In 2014, NIST convened the CPS PWG with a kick-off webinar in June and a face-to-face meeting in August. This grew out of a recognition that, while companies are already building CPS, there lacks a unified technical foundation for broad collaboration. Missing are a consensus definition and taxonomy, reference architecture, and a shared understanding of the essential roles of timing and cybersecurity. The good news in the CPS field is that there is substantial growth of applications in many sectors ranging from energy to health, disaster resilience, transportation, manufacturing, building management, and others. However, these deployments are often sector-specific and are not designed for interoperability

* Contribution of U.S. government not subject to copyright

across sectors. Further, individual communities, states, and countries are implementing their own, unique solutions that are also not designed for interoperability with their neighbors. The resulting landscape of isolated, legacy systems will only continue to grow, making solutions to create interoperability only more difficult with time, and thus limiting the potential benefits of CPS.

The increasing complexity of a 21st century society demands systems-of-systems solutions that require integrating CPS across domains and at multiple scales. This requires developing a common technical foundation that will enable us to work together to achieve this potential. That's the goal of the CPS Public Working Group.

Participation in the PWG is open and free to everyone, anywhere in the world. Most of the sub-group work is done in virtual meetings and using web collaboration tools allowing participation from anywhere. All of the products of the PWG will be openly available online to anyone. The output of the PWG to the public will be two documents developed in sequential phases: a CPS framework that describes best practices and options using current technology, and a CPS Technology Roadmap identifying opportunities for a coordinated effort on key technical challenges. The CPS framework will be released as a draft for public review soon, in the spring of 2015, from [7].

The PWG is organized into five subgroups each of which is led by a collaboration of three co-chairs: one from each of NIST, academia, and industry. The five subgroups are reference architecture, use cases, cyber-security, timing, and data interoperability.

The timing part of the CPS framework document consists of three major sections. First, the time-awareness section examines the components of a CPS from the perspective of the presence or absence of explicit time in the models used to describe, analyze, and design CPS and in the actual operation of the components. Next the time and latency section addresses the use of time to provide bounded latency in a CPS. Thirdly, the section on secure and resilient time addresses the special security problems associated with timing.

We focus in this paper on the time and latency section, discussing the need for and status of convergence between time-sensitive and best-effort processes in CPS nodes and interconnecting networks.

III. TIME AND LATENCY IN CPS

The aim of this section in the CPS PWG framework is to provide reference architectures/frameworks that enable building time-aware CPS to solve control and measurement applications.

Given the diversity in CPS applications and scale, it is not surprising that temporal considerations vary considerably over the range. For example, in small closed systems such as a packaging machine, the primary temporal concern is that all components respect a self-consistent timing design. In such systems, networking temporal considerations, e.g. design of a

TDMA scheme, are part of the design itself. However in large scale, and more critically, in environments characterized as "System of Systems", timing issues are more difficult. For example "smart highways" will involve many different systems, some in the vehicle, some in the infrastructure, some in a traffic management center, etc. Each will have its own temporal requirements which must be met while sharing network bandwidth and in some cases computation bandwidth on servers. Many technological challenges remain in managing the timing in such systems. The remainder of this section discusses both the general issues as well as some of the current thinking on these issues. Some of these can be applied to smaller systems. There is no doubt that the work on larger systems will result in improvements, e.g. in time-sensitive network technology, that will make small system temporal design much easier and more robust.

CPS are used in both control and measurement applications. The requirement of bounded latency is obvious in control systems where the latency from when a physical input is read to when a physical output is written has to be proven by timing and schedulability analysis. In large-scale control systems this requirement becomes even more challenging since the input, computation and output may be occurring on different nodes that are spatially distributed. The challenges of predictability in software are added to by the non-determinism provided by layers of software managing data-transfer on the network connecting these nodes. As the scale of CPS expand to Systems of Systems, the impact on timing of Cloud Computing and Networking concepts such as Software-Defined Networking (SDN) and Network Functions Virtualization (NFV) need to be carefully considered.

In CPS-based measurement systems, the deterministic relationship between acquired data (e.g. simultaneity) is of paramount importance. However, what is typically overlooked is the efficiency and complexity of transferring the acquired data from thousands of nodes to one or more aggregating units, where analytics or logging is being performed. Misaligned data can result in faulty conclusions. In many CPS-based applications, the data measurements are used for asset or structural-health monitoring and in many cases a timely response based on real-time analytics is required. Time, when applied to data-transfer can enable bandwidth reservation in networks used in these measurement applications, thereby enabling faster analytics, a smaller memory footprint, and increased efficiency in data-reduction techniques (for logging). Moreover, bounded latency is extremely useful in distributing triggers to multiple nodes inside a CPS.

Similar to CPUs, computer networking has traditionally been optimized for "best effort delivery", and that has worked extremely well in the past and will continue to do so in the future for many uses. However, a challenge exists when the same networking technology is used for time-sensitive applications that are served by CPS. There is much work being done for enabling time-based CPS, using standard Ethernet technologies to enable seamless integration with the Internet. This "Time-Awareness" in standard Ethernet is paving the way to enable time-sensitive (bounded latency) traffic to coexist on the same network as traditional best-effort (no latency guarantees) traffic. Further details of this work relating to

networks, FPGAs and computers can be found in [8] [9] [10] [11] [12] [13] [14].

A. CPS Domain and Network Managers

A time-aware CPS should guarantee bounds on latency of data delivery and guarantees on synchronization accuracy as it applies to timing correlation of physical I/O. To build such large-scale systems with these guarantees the following two concepts of CPS Domain and CPS Network Manager (CNM) are defined.

CPS Domain: A CPS domain is a logical group of CPS nodes and bridges which form a network with their own timing master. The master may synchronize to a globally traceable time source (e.g. GPS). Each CPS domain has its own primary (or self-consistent as described earlier) time-scale. This time-scale provides a strong monotonically increasing clock to applications for performing input/output functions and time-based scheduling. The timing master of a CPS domain should not produce a discontinuity of time once time-sensitive data transfer within the domain has commenced, even if the master loses connectivity to its global source (e.g. GPS) sporadically.

If a global traceable time is required inside a CPS node, then the node can implement a second time-scale called the Global Traceable Time-Scale. This time-scale can be managed independent of the CPSs primary Time-scale. To correlate the CPS's primary time-scale to the Global Traceable Time-Scale, the offset of the primary time-scale from the Global Traceable Time-Scale can be maintained at all times by the CPS node. The Global Traceable Time-Scale can be used to correlate CPS Time-Scales from multiple CPS domains. This is illustrated in Fig. 1.

Many CPS will be small enough that they don't need an external time-scale and the primary time-scale will suffice. However, significant benefits can accrue from such systems being, and some level of traceable timing may be available, though perhaps not at the needed stability or accuracy.

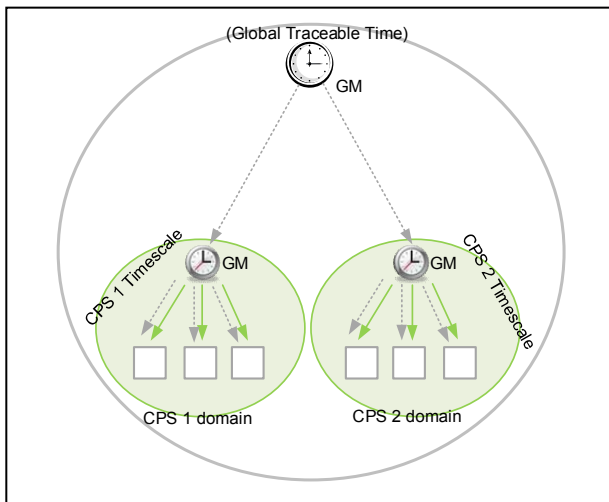


Fig. 1. Domains and Multiple Time-scales in Time-aware CPSs

The functions of a CNM vary depending on the size of the system. These functions include:

- Control and manage the state of all CPS nodes in a CPS domain.
- Coordinate with a centralized network controller to configure bridges in a CPS domain.
- Configure transmission schedules on CPS nodes
- Monitor the health of the CPS domain (for handling errors, changing schedules and bringing new CPS nodes online, etc.).
- Configure application and I/O timing on each CPS node
- Configure any static timing requirements for time-based synchronization

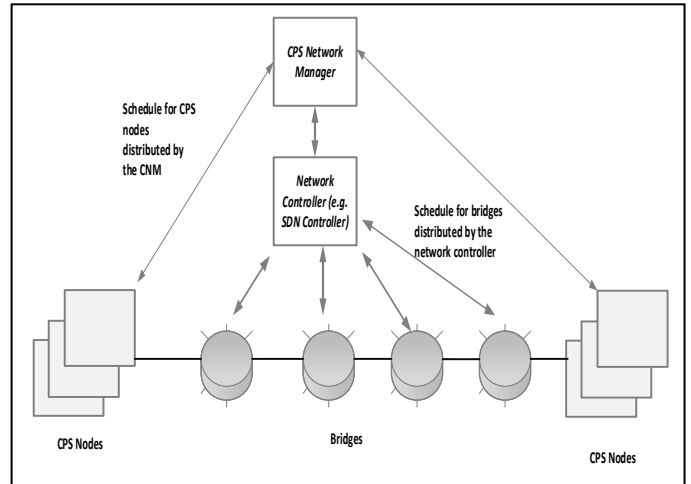


Fig. 2. CPS Network Manager configuring a CPS

Either the CNM or the centralized network controller has to gather performance metrics and determine the topology of CPS nodes in a CPS domain in order to create a schedule. The relevant performance metrics include Bridge Delays, Propagation Delays, and Forwarding/Transmission delays. There are multiple ways to detect topology. For example, one approach to Software Defined Networking (SDN) defines a “Packet-In” “Packet-Out” protocol which uses Openflow [14] with Link Layer Discovery Protocol (LLDP) [15]. Some other protocols like PROFINET [16] use Simple Network Management Protocol (SNMP) [17] along with LLDP. The Centralized Network Manager computes the topology for the CPS domain using these mechanisms, and determines the bandwidth requirements for each time-sensitive stream based on application requirements. The bandwidth can be specified by the period and the size of the frame. Optionally the application can also specify a range $\langle \text{min}, \text{max} \rangle$ for the offset from start of a period. This information is provided to the Centralized Network Controller. The Centralized Network Controller computes the path for the streams and gathers performance metrics for the stream (latency through the path and through the bridges). This information is then used to compute the schedule for the transmission time of each time-sensitive stream and the bridge shaper/gate events to ensure that each time-sensitive stream has guaranteed latency through each bridge. Additionally, queues in bridges are reserved for each stream to guarantee bandwidth for zero congestion loss. It should be noted that schedulability analysis and computation

is the subject of continuing research as the problem becomes intractable for large systems.

It should also be noted that there is considerable activity in the IEEE 802.1 and other standards communities in providing additional tools for controlling network temporal properties.

B. Converging Time-Sensitive and Best-Effort Processes

Many CPS nodes will need to combine time-sensitive with best-effort processes. Such a time-aware node will have separate streams for the two types of data and applications. An illustration of a possible device model for a time-aware CPS node is shown in Fig. 3.

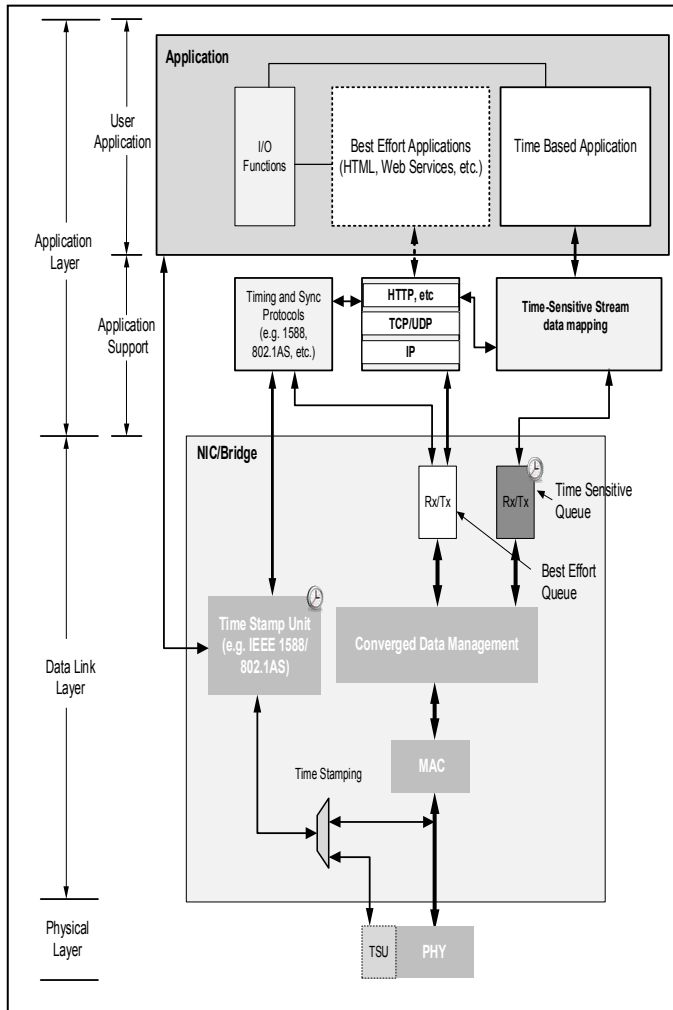


Fig. 3. Time-Aware CPS Device Model

The physical layer receives data units from the data link layer and encodes the bits into signals and transmits the resulting physical signals to the transmission medium connected to the CPS node. If the physical layer supports a time stamp unit (TSU) then its management interface should be connected to the data link layer so that a time stamp can be retrieved as and when required by the timing and synchronization protocol (e.g. IEEE Std. 1588TM [10]).

The data link layer provides time-sensitive data communication among devices in a CPS domain. The data link layer implements a set of dedicated buffer pairs (Tx and Rx queues) for time-sensitive data. At a minimum two pairs of buffers are required so that time sensitive data can be managed independently from best effort data. The time-sensitive transmit buffer is connected to a scheduled (time-triggered) transmit unit. This unit uses a schedule provided by the CPS Network Manager and reads data from the application and copies it into the time sensitive transmit frame and transmits the frame on to the CPS domain.

- The application layer consists of two parts:
- Application-support protocols: These are the protocols that support the conveyance of time sensitive data at the user's application level.
- Time-Sensitive Data Mapping: Protocol to manage the mapping of application data to time sensitive data exchange frames between devices. An example can be CANopen [18] which is used as a data-mapping protocol by multiple industrial protocols.
- Best-Effort protocols: Used for standard internet access, non- time-sensitive streams.
- Timing and Sync Protocols: These include protocols which propagate synchronized time from the network to the application (including I/O functions). Some examples of such protocols are IEEE 1588, IEEE 802.1AS [19], etc.
- User application: User defined applications accessing time sensitive and best effort data, and time-sensitive I/O interfaces to allow decoupling of logical and physical time with enforcement only at the boundary to physics. An example of a realization of this capability is inherent in the design of the Texas Instruments DP83630 Ethernet PHY¹.

Currently time in a CPU is implemented via time-stamp counters (TSC) that increment time using the local clock driving the CPU. This clock does not maintain network time. The TSC can be disciplined via software to slave it to network time. However this leads to significant loss of precision and accuracy. For CPS nodes that synchronize to a single external clock source, it may be desirable to have the TSC driven directly by the network time. This may be implemented by linking the registers of the TSC with the timekeeper in the network interface or by providing a common time-base which can be atomically captured by the network interface before propagating the network time to the CPU or any peripheral device. More generally, CPS applications may choose to maintain offset/PPM state for each derived clock and translate on-the-fly as needed without physically disciplining the TSC. This is especially useful in cases where the applications care about multiple time sources.

Languages used for modeling and programming of time-aware CPS need time as a fundamental programming semantic. Time in the language is required when interfacing to physical I/O and the network. Functions that take future time events to read physical inputs and write physical outputs can enable coordination of physical I/O with scheduled data on the network. Additionally, time-triggered loops can enable

coordination of logic execution with schedule of transmission of data. PTIDES¹ [20] and LabVIEW¹ [21] are two examples of system design tools which implement these time-based programming semantics.

CPS can employ operating systems with a wide range of complexities, from a simple application-level infinite loop (e.g. the Arduino platform) to a virtual machine hypervisor running several instances of virtualized systems on a multi-blade, multi-core hardware platform. The issues that arise throughout these systems with respect to time-awareness are how to get time to the application with a bounded latency and with accuracy, and how to schedule tasks with time accuracy and bounded latency.

At the application layer, the introduction of explicit time will have a profound impact on the conception, design, execution, and robustness of CPS applications. This is a very active area of research, but there are hints of things to come. For example the concept of decoupling of logical and physical time with enforcement only at the boundary to physics mentioned above has yet to be fully exploited. In some cases, tradeoffs can be exploited by applications between message passing, which consumes network bandwidth, and reasoning about timestamps, which can in some cases eliminate some of the messages. An example of this in database management is the Google Spanner system¹ [22].

Building CPS using the above mentioned techniques will make it easier to characterize systems, which is a key requirement of safety-critical systems. CPS with scheduled converged networks built with FPGAs and time-aware CPUs will provide static guarantees and always satisfy timing requirements for their time-sensitive traffic. Architecture-specific analysis tools can derive these guarantees in the form of upper and sometimes also lower bounds on all execution times, since time is foundational in all elements of the CPS.

C. Needed research

We identify a number of areas where research on timing is needed to ensure that the full potential of CPS is realized.

Further research in languages used for modeling and programming time-aware CPS is desired that will allow an application written on a CPS node to be represented as one or more timed-functional modules which can be shifted in time by a Schedule Generator to align production and consumption of time-sensitive data on a converged network. Methodologies which allow coordinating these timed-functional units with respect to each other would enable aligning inputs and outputs across disparate CPS nodes. New techniques that enable harvesting timing information of software functions during the design-phase will allow for better characterization of applications and thereby enable a CPS to be built correct by design.

Increasing precision of timestamps will not only improve application timing but allow better utilization of bandwidth on the network. Currently, asymmetry of delay in networks and phase errors due to asynchronous clocks driving the transmissions on a network are the major causes of inaccuracy in time transfer. Research aimed at increasing precision of

timestamps by enabling new hardware and software methods to correct these asymmetries and phase errors will improve clock accuracy by an order of magnitude.

Better precision time-based synchronization [23] in IEEE 802.11 will enable time-awareness in wireless access points and stations. Research into mechanisms that use these synchronized clocks to create a TDMA-based scheme that can coexist with best-effort traffic similar to wired Ethernet (802.1Q) will enable reduced cost of infrastructure for many CPS applications.

WANs currently offer QoS over dedicated connections for all forms of real time communication (RTC) such as audio and video streaming. Developing synchronized time and the methods described in this paper, the same QoS would be possible over standard networks, thereby reducing costs and increasing accessibility.

IV. CONCLUSIONS

The expected massive growth in the new Internet of Things, encompassing Cyber-Physical Systems and the Industrial Internet, will require a convergence of time-sensitive systems with best-effort systems. Much work is already underway, though new research remains, which will require collaboration among different fields. The extent to which these timing challenges are met and surpassed will dictate the success of emerging CPS applications and others as yet unheard of.

REFERENCES

- [1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [2] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's intranet of things to a future internet of things: a wireless-and mobility-related view," *Wireless Communications, IEEE*, vol. 17, no. 6, pp. 44–51, 2010.
- [3] J. O. Smith, "Standardization in machine-to-machine (M2M) communications," 2012. [Online]. Available: http://connectedvehicle.itsa.wikispaces.net/file/view/ITSA_Webinar_February_15_2012_Jeff_Smith_Numerex_GSCMSTF.pdf
- [4] J. Bradley, J. Barbier, and D. Handler, "Embracing the internet of everything to capture your share of \$14.4 trillion," 2013. [Online]. Available: http://www.cisco.com/web/about/ac79/docs/innov/loE_Economy.pdf
- [5] P. C. Evans and M. Annunziata, "Industrial internet: Pushing the boundaries of minds and machines," 2012. [Online]. Available: http://www.ge.com/docs/chapters/Industrial_Internet.pdf
- [6] NIST Cyber-Physical Systems Homepage, [Online]. Available: <http://www.nist.gov/cps/>
- [7] NIST, "Cyber-physical systems public working group," 2014. [Online]. Available: <http://www.cpspwg.org/>
- [8] H. Kopetz and G. Bauer, "The time-triggered architecture," *Proceedings of the IEEE*, 91(1):112–126, 2003.
- [9] J. Jasperneite, J. Feld, "PROFINET: an integration platform for heterogeneous industrial communication systems," *Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on*, vol. 1, no., pp. 8–22, 19–22 Sept. 2005
- [10] IEEE Instrumentation and Measurement Society, "1588: IEEE standard for a precision clock synchronization protocol for networked measurement and control systems," IEEE, Standard Specification, July

¹ Commercial equipment is identified in order to describe a concept adequately. It is not intended to imply recommendation or endorsement, nor is it intended to imply that these entities, materials, or equipment are necessarily the best available for the purpose.

- 24, 2008
- [11] G. C. Buttazzo: Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications, Third Edition. Real-Time Systems Series 24, Springer 2011, ISBN 978-1-4614-0675-4, pp. 1-521
- [12] P. Axer, R. Ernst, H. Falk, A. Girault, D. Grund, N. Guan, B. Jonsson, P. Marwedel, J. Reineke, C. Rochange, M. Sebastian, R. von Hanxleden, R. Wilhelm, W. Yi, "Building timing predictable embedded systems," ACM Trans. Embedded Comput. Syst. 13(4): 82 (2014)
- [13] R. Wilhelm, J. Engblom, A. Ermedahl, N. Holsti, S. Thesing, D.B. Whalley, G. Bernat, C. Ferdinand, R. Heckmann, T. Mitra, F. Mueller, I. Puaut, P. Puschner, J. Staschulat, P. Stenström, "The worst-case execution-time problem - overview of methods and survey of tools," ACM Trans. Embedded Comput. Syst. 7(3) (2008)
- [14] Open Networking Foundation, "OpenFlow Switch Specification," October 14, 2013. [Online] Available from: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf>
- [15] P. Congdon, "Link Layer Discovery Protocol Overview (LLDP)," March 8, 2003. [Online] Available from: <http://www.ieee802.org/1/files/public/docs2002/LLDP%20Overview.pdf>
- [16] PROFINET. [Online] presentation at: <http://www.profibus.com/technology/profinet/>
- [17] IETF Network Working Group, "Simple Network Management Protocol (SNMP)". [Online] available at: <https://www.ietf.org/rfc/rfc1157.txt>
- [18] CANopen. [Online] available from: <http://www.can-cia.org/index.php?id=canopen>
- [19] 802.1AS - Timing and Synchronization. [Online] available from: <http://www.ieee802.org/1/pages/802.1as.html>
- [20] Center for Hybrid and Embedded Software (CHESS), UC Berkeley, "PTIDES". [Online] see: <http://chess.eecs.berkeley.edu/ptides/>
- [21] National Instruments, "LabVIEW System Design Software". [Online] see: <http://www.ni.com/labview/>
- [22] J. C. Corbett, et al. "Spanner: Google's globally distributed database," ACM Transactions on Computer Systems (TOCS) 31.3 (2013): 8.
- [23] P802.11Rev-mc , "Fine Timing Measurement"