An Overview of Security Aspects of the PTP Algorithm and their Effects in Networked Control Systems with TDMA under Time Discontinuity Faults

Eloy Martins de Oliveira Junior, Marcelo Lopes de Oliveira e Souza
Instituto Nacional de Pesquisas Espaciais – INPE – Av. dos Astronautas, 1758, São José dos Campos, SP, Brasil

***Abstract* — Current systems such as satellites, aircrafts, automobiles and traffic controls are becoming increasingly complex and/or highly integrated as prescribed by the SAE-ARP-4754A Standard. Such systems operate in distributed environments such as Networked Control Systems(NCS), which frequently require time synchronization among different devices, levels and granularities aiming to minimize the time de-synchronizations. However, the common approaches for this do not address the time discontinuities that occur during re-synchronizations and that may cause accidental faults. These faults may have serious consequences for safety. Even worse, these faults may be maliciously explored to cause serious consequences for security. Towards avoiding both, this paper presents an overview of security aspects of the PTP algorithm and their effects in NCS with TDMA under time discontinuity faults. It is based on a review of the literature, discussions and simulations of a NCS with TDMA which requires time synchronization using PTP. This overview show preliminarily that: 1)the effects of time discontinuities in NCS may be serious; 2)new techniques and analysis must be developed to address the time discontinuities for the improvement of security of NCS.**

***Keywords* — Cyber Security, Time Synchronization, Time Discontinuity.**

I. Introduction

Current systems such as satellites, aircrafts, automobiles, turbines, power controls and traffic controls are becoming increasingly complex and/or highly integrated as prescribed by the SAE-ARP-4754A Standard [1]. Such systems are part of a critical infrastructure of a country. The largest trend in these applications are: 1) to integrate computations, communications and real time controls in different levels of operations, using a large number of actuators, sensors and controllers implemented in intercommunicating processors; and 2) to improve the cyber security to avoid malicious attacks.

This paper emphasizes the security of the real time systems related to time triggered architectures and technologies, where the sensors, actuators, and controllers are synchronized and connected via a network to form a networked control system (NCS). These systems require predictability in the logical domain and in the temporal domain [2]. The temporal requirements may become very strict, thus demanding time synchronization among different devices, levels and granularities. Clocks have imperfections caused by the environmental fluctuations, aging, the non-linear dynamics and/or long lifetimes. The imperfections may cause accidental or intentional fluctuations beyond a tolerance in clock synchronization (in this paper, we call this fluctuation beyond a tolerance as clock de-synchronization) which can cause faults or failures in such systems.

There are many techniques to minimize the clock de-synchronization. According to [3], the state of the art clock synchronization methods use two different paradigms: 1) master/slave approach, where basically the slave clocks receive periodically a time from a real master clock and then the slave clocks compare and adjust their times; and 2) democratic approach, which basically, does not use a real master clock to synchronize the system. To achieve clock synchronization, this approach creates a global time (virtual master clock). Furthermore, there is an impression that the synchronization task is a trivial one, but this is an unfortunate misconception [4]. Among the main reasons that make the synchronization a non-trivial task, this paper addresses two:

- First, due to the complexity and/or high integration, the clock synchronization becomes critical and complex to achieve.
- Second, the common clock synchronization approaches frequently do not address the time discontinuities. Time discontinuities lead to cause accidental or intentional faults or failures in Networked Control Systems (NCS) among different devices, levels and granularities.

The first can be minimized by using known integrated architectures as the Integrated Modular Avionics (IMA) [5] which prescribes rules for incremental design and certification. The second can be minimized by using some standards as: 1) the SAE-AS6003-TTP [6], and SAE-AS6802-TTEthernet [7] which prescribes rules for deterministic communication; and 2) IEEE 1588-2008 [8] which prescribes Precision Time Protocol (PTP) and other rules to time synchronization in a distributed system over a network.

The time discontinuities are partially addressed in many publications as [9-12]. According to [10], it might be desirable to change clocks gradually. However, it requires a continuous clock synchronization that makes the clock synchronization function more complicated, as shown in [9-

10, 12], and harder to be implemented in high levels under high requirements of granularity and precision. So, the discrete clock synchronization approach is a reasonable solution in most systems. However, some problems appear with this approach as the time discontinuity. In [11], a clock synchronization algorithm that always uses a forward correction was proposed. The system achieves a time synchronization with the reference clock (virtual or real clock), but this can cause the loss of time, although the forward approach can increase a skew of time in relation to an external reference clock (like UTC, TAI, atomic clock). In [12], a time remapping of scheduler to avoid the time discontinuities in a computational level was proposed, but this solution only avoids the time discontinuities at the computational level. However, the [12] model assumes the possibility of the same task getting dispatched again. In this paper, we assume that when the task is dispatched, it cannot be dispatched again. This approach intends to achieve a better model of the hard real time critical tasks. The time discontinuity could cause many effects in networked control systems (NCS) in different levels as computation, communication and control. This paper intends to show the effects that the malicious time discontinuities may cause by the clock synchronization using PTP in a NCS.

II. Precison Time Protocol

The PTP algorithm follow the master/slave approach which uses a real master clock, i.e., the slave clocks receive periodically a time from master clock and then compare and adjust their times. The master clock usually is of greater quality, to establish an accurate time base. It can be: an atomic clock, an international standard time reference (UTC – International Time Coordinated or TAI – International Atomic Time), or the GPS which contains an atomic clock inside. In most cases the algorithms and techniques to establish a clock synchronization follow the IEEE 1588-2002 and now 2008 standard [3,8] which defines Precision Time Protocol (PTP) and other protocols and rules to synchronize time in a distributed system over a network [3,7,8,13]. Fig. 1 shows the PTP scheme.

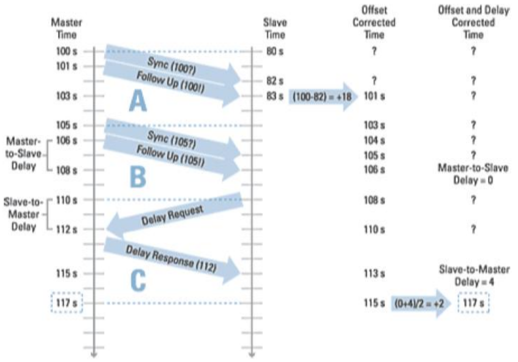

Fig. 1. PTP scheme. Source: [13].

According to [13], in the PTP, the master clock periodically sends a sync packet containing a timestamp of the time when the packet has left. The master may also, optionally, sends a follow up packet containing the timestamp for the sync packet. The follow up packet allows the master to accurately timestamp the sync packet on networks where the departure time of a packet cannot be known accurately beforehand. A slave clock receives the master's sync packet and using its own clock timestamps in the packet's arrival time. The difference of the sync packet departure timestamp and the sync packet the arrival timestamp is the slave clock's offset from the master with the network propagation delay. By adjusting its clock by the offset measured at this point, the offset between the master and slave can be reduced to the network propagation delay only. Assuming the symmetrical delay, the slave can discover, and compensate the propagation delay. It accomplishes this by issuing a delay request packet which is time tstamped on departure from the slave. The delay request message is received and time stamped by the master clock, and the arrival timestamp is sent back to the slave clock in a delay response packet. The difference in these two timestamps is the network propagation delay.

According to [13], the IEEE 1588 specification does not include any standard implementation for adjusting a clock; it merely provides a standard protocol for exchanging these messages. This approach, besides not being tolerant to byzantine errors, has the disadvantage of having a common point of failure; *i.e.*, if the master clock fails then all slave clocks lose the reference for synchronization, compromising the system. Furthermore, the IEEE 1588 assumes that the network delay is symmetrical. These disadvantages have been minimized with: 1) the use of master clocks of greater quality; 2) other efforts to minimize the problem of common point of failure, as the best master clock algorithm (BMC) [13]; 3) assuming in non-critical cases a symmetrical network delay; all at the expense of the cost of the system and increased the reliability of this approach over the years.

III. Characterization and Security Aspects of Time Discontinuity

Characterization

A common time base in distributed real-time systems enables the order of event occurrences, measures time durations and schedules real-time tasks [15]. The clock synchronization algorithms intend to achieve a consistent time base among the entities of real-time distributed systems. However, the discrete clock synchronization algorithms may cause abrupt changes in time, also known as the time discontinuity. In each re-synchronization interval, the algorithms provide the adjustment of local time. The adjustment causes a jump of time, which is the time discontinuity. There are many possible models to perform a modeling and timing analysis, however, these models usually consider only the computational or communicational levels, disregarding the interactions and interferences with the dynamics level. Unfortunately, this leads to models with limited knowledge about the system environment [16], limiting a timing system analysis, mainly about the backward and forward adjustments of time.

According to [12] and his model, the discrete clock synchronization may cause a confusion in the run-time system into making a wrong judgment on real-time tasks, that can leads a system to undesirable states and faults:

- **constraint disappearance:** is when the tasks think wrongly that they missed their deadlines and lost a chance of getting dispatched [12].
- **constraint reappearance:** is when the tasks were dispatched, and think wrongly that they need to be dispatched again [12].

This paper do not consider the constraint reappearance. In this paper, if task was dispatched, it is not dispatched again. For correct timing analysis in our model, the first step is characterizing the backward and forward corrections of time that the clock synchronization algorithms follow as shown in Fig. 2 and Fig. 3. The time discontinuity is highlighted by a red circle. In the backward correction, the local clock (in blue) runs faster than the reference clock (real or virtual, in black) due to a clock drift.

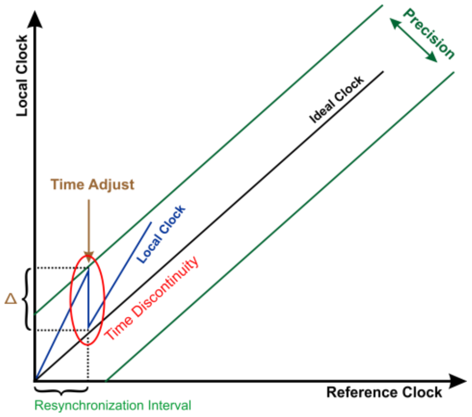

Fig. 2. Backward Correction.

So, the algorithms provide a periodic re-synchronization to keep all system clocks inside a precision. In this case (runs faster), each re-synchronization interval of the local clock needs a backward correction in time to keep the local clock inside the precision. The backward correction is calculated by the convergence function of algorithms. However, this approach needs to be used with care because the backward correction can result in negative time which invalidates the local time [9].

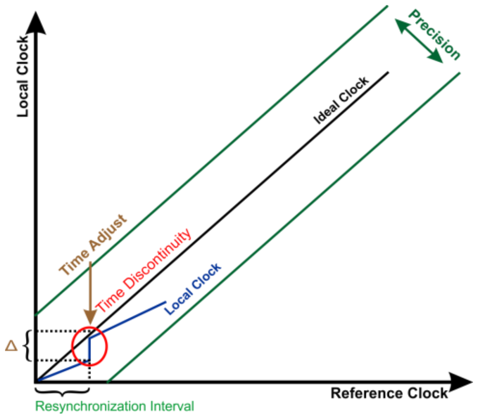

Fig. 3. Forward Correction.

In the forward correction, the local clock (in blue) runs slower than the reference clock (real or virtual, in black) due to a clock drift. So, the algorithms provide a periodic re-synchronization to keep all system clocks inside a precision. In this case (runs slower), each re-synchronization interval of the local time needs a forward correction of time to keep the local clock inside the precision. The forward correction is calculated by the convergence function of algorithms. However, the forward correction can result in a loss of time [12] and may cause an exacerbation of a drift in relation to a external clock (UTC, TAI, GPS).

Security Aspects

Currently, due to the new global situation about the cyber defense, the security becomes an important research area. In particular, at the systems that compose the principal critical infrastructure of a country. Such systems integrate more and more computations, communications and real time controls in different levels of operation, using a large number of actuators, sensors and controllers implemented in intercommunicating processors which frequently requires a time synchronization among different devices, levels and granularities. Time synchronization security is an important aspect when clock synchronization is used in commercial applications, in public networks or even in critical control applications which make use of a network. In general, the problem of time synchronization of logical clocks caused by natural imperfections is solved by algorithms. However, it is obvious that the overall functionality of those systems can be degraded or even disabled if the mechanism of synchronization of clocks is attacked [18]. The attacks to the mechanism of synchronization of clocks can cause faults and risks of accidents. In general, each system needs a security policy of time synchronization, and each network have their own vulnerability list, as shown in [18]. This paper emphasizes a time discontinuity attack. Currently, many systems use logical clocks, so, in the absence of a cyber security policy, it becomes easy to generate a jump in time, maliciously, in the logical clock degrading the time synchronization mechanism, and hence causing a time discontinuity fault or failure at the control system.

IV. Case of Study

Through simulations, this paper intends to show that time discontinuities pose consistency problems to distributed real-time control applications. This paper studies the effects of malicious changes of local time that causes a exacerbate drift that causes a exacerbate backward time discontinuities by means of simulation over a networked control system using a TDMA philosophy for the communication protocol. For this study, this paper used the TrueTime/Matlab/Simulink environment [17] for simulations. The master clock, sensor, actuator and controller were connected via the network. The controller used was a digital PID (Proportional, Integral and Derivative). Fig. 4 illustrates the NCS model used in this paper.

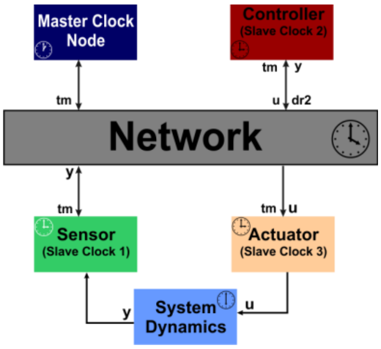

Fig. 4. NCS block diagram.

The actuator/plant was a second order marginally stable continuous time system, according to the following transfer function:

$$G(s) = \frac{1000}{s^2 + s} \qquad (1)$$

All nodes had logical clocks given by the virtual computer of the TrueTime Kernel; and they used the databus network to exchange data among them. The master time is the virtual time given by the logical clock of the TrueTime Kernel, namely master clock node.

The model of a virtual local time synchronization is:

$$t_L(k+1) = t_L(k) + \lambda DT + R(j) \qquad (2)$$

- **k -** each instant;
- **t_L -** local time at instant k;
- **λ -** Attack factor;
- **D -** Local drift;
- **T** - Sampling period;
- **j -** each instant of Re-synchronization function;
- **R -** Re-synchronization function at instant j;

In this model described by equation 2, the local drift (D) is a parameter that described how the local clock runs fast or slow in relation to other nodes at instant k.

This paper introduce the attack factor. The attack factor is a parameter that described and provide a metric to analyze how the local time is changed maliciously in each instant.

In this paper, we applied a constant attack factor 10 at Sensor 1, *i.e.*, the logical clock of sensor node virtually runs 10% faster than other nodes. For the time synchronization is used a PTP, as showed Fig. 1. To ensure that all nodes have a consistent view of time we need to re-synchronize the clocks periodically. For this, the Master Clock Node sends periodically its time to all nodes, and each node corrects its own clock and requests a delay. With this, the malicious deviations caused by the drift of the logical clock in Sensor 1 are adjusted and all system clocks are within a certain precision. The deviations is high due to the attack factor and hence the PTP causes a high time discontinuity in each resynchronization instant degrading the control and step response system.

The model of the simulated NCS is given at Fig. 4. Tables I show the nodes configuration for each node of simulation, as described in Fig. 4.

V. Results

The task configuration of simulations were:

TABLE I. Parameters Configuration for each Node

Node	T (seconds)	λ	D	Resynchronization Period (seconds)
Master	0.004	1	1	0.2
Sensor	0.004	10	1	0.2
Actuator	0.004	1	1	0.2
Controller	0.004	1	1	0.2

We analyzed the clock synchronization using PTP. First, Fig. 5 shows the time difference between the master clock and the slave clocks: Sensor (Slave Clock 1 - Red), Control (Slave Clock 2 - Yellow) and Actuator (Slave Clock 3 - Green); without synchronization. The control and actuator nodes are synchronized within a precision, because they do not have a drift effect. However, the sensor (red) has a maliciously clock drift (10%), and its difference to the master clock was increased.

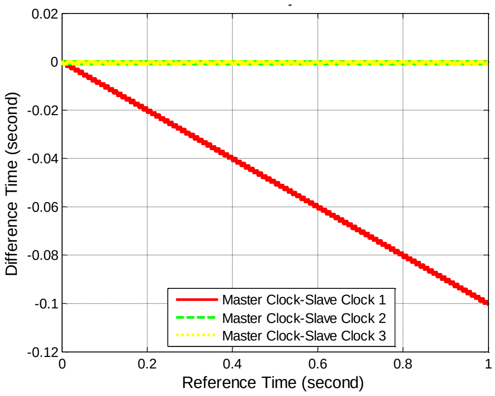

Fig. 5. Time difference of De-Synchronized Clocks.

Fig. 6 shows the time difference between the master clock and the slave clocks: Sensor (Slave Clock 1 - Red), Control (Slave Clock 2 - Yellow) and Actuator (Slave Clock 3 - Green).

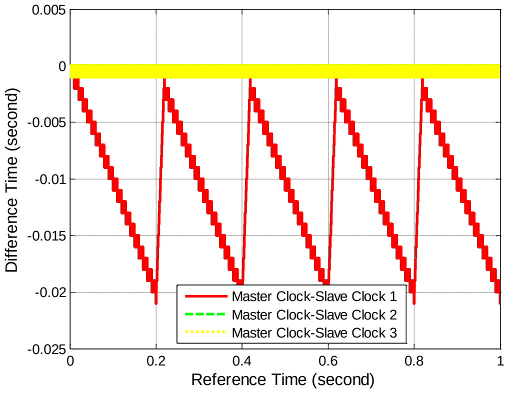

Fig. 6. Time Difference of Synchronized Clocks.

The sensor has a drift, and it is synchronized periodically, showing that PTP fulfilled its role, all clocks are synchronized with the master clock within a 20 ms of tolerance. The excessive value is due to the excessive drift (10%) caused maliciously.

Fig. 7 shows the successive step responses of the system, and Fig. 8 shows the control law of system.

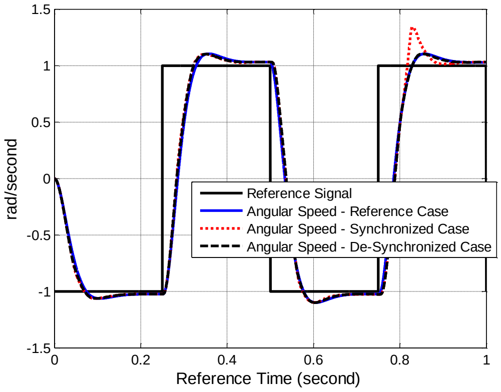

Fig. 7. Step Response of Dynamics.

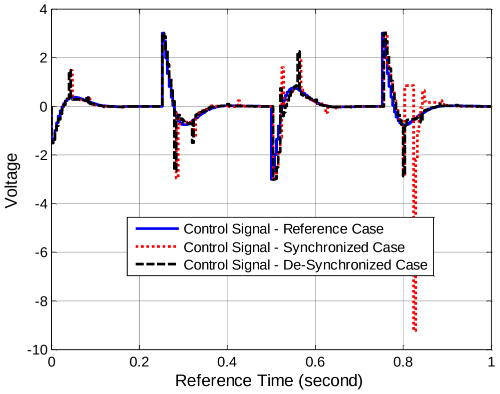

Fig. 8. Control Law.

In Figs. 7-8, the continuous black line is a reference signal (step). The continuous blue line represents the step response of dynamics (Fig. 7), and the control law (Fig. 8) without clock faults and synchronization. The dashed black line represents the step response of dynamics (Fig. 7), and the control law (Fig. 8) with clock fault (in sensor) and without synchronization. The dot red line represents the step response of dynamics (Fig. 7), and the control law (Fig. 8) with clock fault (in sensor) and synchronization. In order, it is possible to observe in Fig. 5-8, that the clock synchronization fulfilled its role, but the malicious excessive local drift cause an excessive time discontinuity (backward correction) caused by PTP that degraded the control law and in consequence the step response of dynamics, especially during resynchronizations and transitions of the system.

VI. CONCLUSIONS

The PTP, prescribed by IEEE 1588, provided a good technique to achieve clock synchronization and proved efficient to achieve the synchronization by reducing the local drift. However, this showed a backward time discontinuity effect over a networked control system. All nodes of the NCS achieve clock synchronization within a precision. However, to achieve this time synchronization, the control law and dynamic response we degraded. These faults may be maliciously explored to cause serious consequences for security. More results need to be analyzed, mainly, using the attack factor, as the forward case.

The results obtained for this case suggest that: 1) Due to the time discontinuity caused by the clock synchronization, the control and response of dynamic systems are degraded; 2) new analytical techniques are needed to avoid the time discontinuity effects over NCS; 3) The attack factor parameter helps the analysis of attack; and 4) The countermeasures and a cyber security policy are important to avoid the maliciously attacks at the local clocks.

Suggestions of Countermeasures

Based on these conclusions, to reach the required security goals, we suggest installing various countermeasures on various levels as:

- Firewalls to protect the transmission media;
- FDIR means to: Detect an error in the master clock, Isolate and Identify an error in the master clock, and Reconfigure the architecture by choosing another master clock;
- New time synchronization algorithms to identify and prevent any abrupt change in time at instant, and that minimize a time discontinuity;
- The attack factor may be help the designers to analyze the weakness of time synchronization;
- Security measures by hardware timestamp, i.e., include a new layer in the clock synchronization process;
- Introduction of real time clock, as a GPS, to provide a real global time to increase a time synchronization reliability;

ACKNOWLEDGMENTS

Both authors thank the INPE, its Course and Division of Space Mechanics and Control for supporting them during the writing of this paper. The first author acknowledges CNPQ financial support during his Ph.D. Program and the writing of this paper.

REFERENCES

[1] SAE, “Certification Considerations for Highly-Integrated or Complex Aircraft Systems,” Aerospace Recommended Practice ARP-4754a, SAE, Dec. 2010.

[2] J. A. Stankovic, "Misconceptions about Real-Time Computing: a Serious Problem For Next-Generation Systems," IEEE Computer Society, vol.21, no.10, pp.10,19, Oct. 1988, doi: 10.1109/2.7053.

[3] G. Gaderer, S. Rinaldi, N. Kero, "Master Failures in the Precision Time Protocol," Precision Clock Synchronization for Measurement, Control and Communication, 2008. ISPCS 2008. IEEE International Symposium on , vol., no., pp.59,64, 22-26 Sept. 2008, doi: 10.1109/ISPCS.2008.4659214.

[4] H. Meyr, G. Ascheid, "Synchronization in Digital Communications: V1-Phase-, Frequency-Locked Loops, and Amplitude Control", vol. 1, USA: John Wiley & Sons, 1990, ISBN --471-50193-X (v. 1).

[5] G. B. S. Tagawa, M. L. O. Souza, "A Discussion on the Use of an Integrated Modular Avionics (IMA) Architecture to Simulate an Aerospace Control System," SAE Technical Paper 2011-36-0182, 2011.

[6] SAE, “TTP Communication Protocol", AS6003, SAE, Feb. 2011.

[7] SAE, “Time-Triggered Ethernet", AS6802, SAE, Nov. 2011.

[8] 1588-2008 - IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems.

[9] H. Kopetz, W. Ochsenreiter, "Clock Synchronization in Distributed Real-Time Systems," Computers, IEEE Transactions on , vol.C-36, no.8, pp.933,940, Aug. 1987.

[10] L. Lamport, P. M. Melliar-Smith, "Synchronizing Clocks in the Presence of Faults.", Journal of the ACM, 32(1):52–78, Jan. 1985.

[11] T. K. Srikanth, S. Toueg, "Optimal Clock Synchronization", Journal of ACM, 34(3):626–645, July 1987.

[12] M. Ryu, J. Park, S. Hong, "Timing Constraint Remapping to Avoid Time Discontinuities in Distributed Real-Time Systems," Real-Time Technology and Applications Symposium, 1999. Proceedings of the Fifth IEEE , vol., no., pp.89,98, 1999.

[13] National Instruments White Paper, "Introduction to Distributed Clock Synchronization and the IEEE 1588 PTP", http://www.ni.com/white-paper/2822/en, Fev. 2013, Access on 01/05/2013.

[14] J. Lundelius, N. Lynch, "A New Fault-Tolerant Algorithm for Clock Synchronization", In Proceedings of the Third Annual ACM Symposium on Principles of Distributed Computing. , USA, 1984.

[15] H. Kopetz, "Sparse Time Versus Dense Time in Distributed Real-Time Systems", Distributed Computing Systems, 1992., Proceedings of the 12th International Conference on , vol., no., pp.460,467, 9-12 Jun 1992.

[16] R. Wilhelm, "Timing Analysis and Timing Predictability", In Proceedings of the Third international Conference on Formal Methods for Components and Objects (FMCO'04), Springer-Verlag, Berlin, Heidelberg, 317-323. 2004.

[17] M. Ohlin, D. Henriksson, A. Cervin, “TrueTime 1.5 – Reference Manual”, Department of Automatic Control, Lund University, Sweden, Jan. 2007.

[18] E. M. Oliveira Junior, M. L. O. Souza, " A Brief Comparison of Security Aspects of Time Synchronization in Networked Control Systems using CSMA/CD versus TDMA Protocols", Proceedings of XIV SIGE, São José dos Campos, Brazil, Sept. 2012.