A Brief Comparison of Security Aspects of Time Synchronization in Networked Control Systems using CSMA/CD versus TDMA Protocols

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, traffic controls, military systems and smart grids are becoming increasingly complex and/or highly integrated as prescribed by the SAE-ARP-4754 Standard. The severe constraints, complexity and/or high integration make the security of these systems itself a challenge. Such systems, usually in a form of networked control systems (NCS), require accurate time synchronization among its nodes and devices for correct operation. So, any accidental or intentional fluctuation beyond a tolerance in time synchronization can cause faults in communication and hence in the control systems. This paper presents a brief comparison of security aspects of time synchronization in NCS using CSMA/CD (Carrier Sense Multiple Access - Collision Detection) and TDMA (Time Division Multiple Access) protocols. This highlights how the NCS using CSMA/CD and TDMA networks is affected by the malicious clock de-synchronization; and this includes some simulations to illustrate them. The paper concludes by the comparison of networks and suggests some countermeasures, based on the comparisons and simulations presented.**

Palavras-Chave-**Time Synchronization, CSMA/CD, TDMA.**

I. INTRODUCTION

Currently, the largest trend in real time applications is to integrate computations, communications and real time controls in different levels of operation, using a large number of actuators, sensors and controllers implemented in intercommunicating processors. This can be done according diverse architectures. So, the nodes and devices of these distributed systems often contain real-time clocks that control their performance and coordination. Thus, in accordance with Tsang and Beznosov (2006) [1], the ability to precisely synchronize clocks among the distributed components is critical for complex and/or highly integrated systems as satellites, aircrafts, automobiles, traffic controls, military systems, wind farms and smart grids as prescribed by the SAE-ARP-4754 Standard [2]. These systems require predictability in the logical domain and in the temporal domain [3], *i.e.*, the timing is crucial to their correctness and performance.

This paper emphasizes networked control systems (NCS), where sensors, actuators, and controllers are connected via a network, as shown in Figure 1. In this scheme, the temporal requirements may become very strict, thus demanding clock synchronization among the nodes.

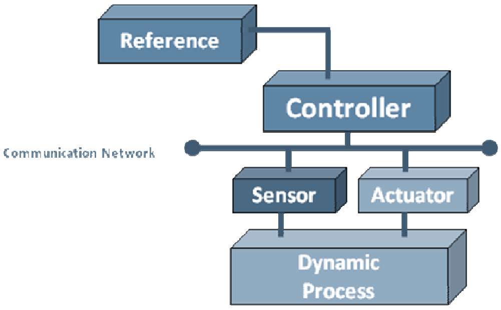

Figure 1. Networked Control System (NCS).

Clock synchronization is an important topic of conception and design. In normal operation, the goal of clock synchronization is: 1) to ensure that the operations follow in the correct order in CSMA/CD and TDMA protocols; and 2) to establish a deterministic data communication among nodes in CSMA/CD and TDMA protocols. So, all nodes need synchronization among the clocks of the entire system. Any fluctuation (de-synchronization) beyond a tolerance in this function can cause a fault in the communication and/or control system. The fluctuations of a clock may be caused by a natural cause (imperfections of clock) or by an External Malicious Agent (EMA). Consequently, there is a lot of concern, not only with the accuracy of the clocks, but with the security of the entire distributed control system.

The main imperfections of clocks are: drift, offset, fluctuation (jitter) and state error. These imperfections are caused by environmental changes such as variations in temperature and voltage, aging of crystal, in case of quartz clock, and some other reasons. However, to correctly understand and to ignore unnecessary details and focus on the essential features of the design, the abstraction of clocks is necessary. Oliveira Junior and Souza (2011) [4], show that the abstractions are often applied in a hierarchical fashion, where each layer of abstraction relies on the essential features of the abstraction level below, and hides unessential details from the lower level [5].

In this paper, we make a comparison of security aspects of time synchronization in networked control systems using CSMA/CD and TDMA protocols, only in the software abstraction layer. As described in [4], the software layer is the most susceptible to attacks by an External Malicious Agent (EMA). This paper aims to show how the clock de-synchronization affects the control of NCS with two different networks, and how this de-synchronization can be caused by an EMA. To do that, in Section II, we discuss basic concepts about clocks and its imperfections, methods, and architectures to achieve clock synchronization. In Section III we discuss the security aspects of clock synchronization, and the possible attacks to clocks by an External Malicious Agent (EMA) that can degrade networked control systems. In Section IV we show some models and simulations. In Section V we show the results and their comparisons. In Section VI we offer some conclusions.

II. BASIC CONCEPTS

In this section, we show the basic concepts about clock synchronization and networked control systems.

Clocks Fault Modes

A physical clock and, therefore, logical clocks, have some imperfections. These imperfections can be caused by environmental changes such as variations in temperature and voltage, aging of crystal, in case of quartz clock, or can be caused by an External Malicious Agent (EMA). Figure 2 shows the main imperfections of a clock.

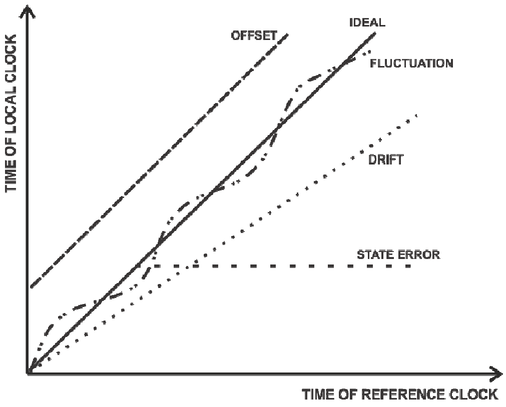

Figure 2. Main Imperfections of a Clock.

These imperfections are known as drift, initial or instantaneous offset, fluctuation and state error:

• **Clock Drift:** Clock Drift is when a local clock has a frequency of oscillation greater or less than another local clock and/or a reference clock; *i.e.*, the drift is the rate of change between the two clocks.

• **Offset:** There are two types of offset: the initial offset is the difference between the initial times of local clocks and/or of a reference clock; the instantaneous offset is the difference between the instantaneous times of local clocks and/or of a reference clock.

• **Fluctuation:** The fluctuation or jitter is the uncertainty in the measurement of the clock.

• **State Error:** The State Error is when a local clock stops the measurement of the progression of time; i.e., the local clock stops on a fixed value. The State Error can be considered a fault or a failure.

These imperfections of a clock are impossible to eliminate. Thus, there is a need to use methods to achieve the clock synchronization within a tolerance and to minimize the effects of these errors of a clock.

There are many methods for clock synchronization. These methods follow the clock synchronization architectures.

CSMA/CD Protocol

The CSMA/CD protocol is based on carrier sensing, *i.e.*, in this network each node listens the carrier and waits the communication channel to be free. When the communication channel is free, the node sends its message. However, the CSMA/CD does not generate a message priority. This may allow two or more nodes attempt to transmit data simultaneously. When this occurs, there is a collision and none of the nodes can transmit data. So, the collision is detected (CD). When the collision is detected the nodes which attempted the transmission, stop; then each wait a variable time and then each attempt to transmit again.

So, the CSMA/CD is not a deterministic protocol as a TDMA and in many real time applications the clock synchronization is required to minimize the effects of non-determinism. So, to ensure a correct order of operation, the algorithms are a necessity.

TDMA Protocol

The TDMA protocol is based on time division, *i.e.*, in this network each node has its own time to send a message. In this network, the clock synchronization is necessary to provide all nodes with an equivalent time concept which is named a global time. To establish clock synchronization, the algorithms are a necessity. Basically, each node measures the difference between the a priori known arrival time and the observed arrival time of a correct message to learn about the difference between the sender's clock and the receiver's clock. So, the algorithm converges to a global time. The algorithm needs this information to periodically calculate a correction term for the local clock so that the clock is kept in synchrony with all other clocks of the cluster. Fault-tolerant average (FTA) and fault-tolerant midpoint (FTM) algorithms [6-8] are used in many network.

Clock Synchronization Architectures

Most methods and algorithms to perform clock synchronization are based in two distinct architectures: Centralized and Distributed. Oliveira Junior e Souza (2011) [4] present a brief discussion about them. In this paper, the distributed architecture is used to establish clock synchronization in CSMA/CD and TDMA networks.

1) Centralized Architecture: The centralized architecture uses a real master clock to synchronize the system. To achieve clock synchronization, this architecture creates a real global time, compares the local times with the real global time and corrects them periodically. Since it was discussed in [4], it will not be discussed here anymore.

2) Distributed Architecture: The distributed architecture does not use a master clock to synchronize the system. To achieve clock synchronization, this architecture creates a virtual global time. The virtual global time is achieved by one mathematical equation involving a subset of the set of clocks. This architecture is frequently used in TDMA communications, where the global time must be achieved for correct operation of the network. However, this architecture is perfectly usable in other network philosophy, as in CSMA/CD to ensure the correct timeliness. Figure 3 shows this architecture over a databus.

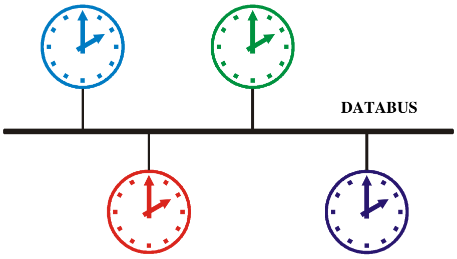

Figure 3. Distributed Architecture.

Figure 3 shows four local clocks where each of them calculates its correction. These calculations by each clock make the convergence to a global time, i.e., all clocks converge to the same time within a tolerance.

This architecture has the advantage of allowing byzantine fault tolerant clock synchronization algorithms. This increases the reliability of the system but it increases the traffic of data and adds a cost on the precision achieved among the set of clocks. Kopetz [6] presents a table of additional costs on precision due to the use of byzantine fault tolerance in an averaging clock synchronization algorithm.

The FTM (Fault-Tolerant Mid-Point) Algorithm, created by Lundelius and Lynch [7], is an example of a byzantine fault tolerant algorithm for a distributed architecture. To ensure that all nodes have a consistent view of time, the re-synchronization of clocks is needed regularly (periodically). The consistent view of time by all clocks is called a (virtual) global clock. For this, the algorithm follows a logical sequence. Each node applies this sequence with the aim of reaching a correction term.

Networked Control System

Current systems such as satellites, aircrafts, traffic controls, military systems and smart grids are becoming increasingly complex and/or highly integrated as prescribed by the SAE-ARP-4754 Standard [2]. Such systems integrate computations, communications and real time controls via networks among other key architectures and technologies to form networked control systems (NCS). Such architectures and technologies usually require accurate clock synchronization among its nodes and devices for correct operation. The introduction of communication lines brings a lot of advantages, and disadvantages. This paper uses a NCS with a CSMA/CD (Carrier Sense Multiple Access / Collision Detection) and a TDMA (Time Triggered Multiple Access) protocols to discuss and compare the security aspects of clock synchronization, as shown below.

III. SECURITY ASPECTS

Security becomes an importance aspect when clock synchronization is used in commercial applications, in public networks or even in critical control applications which make use of a network.

The security aspects are becoming an important research area. In general, the problem of clock synchronization of logical clocks (Software Abstraction Layer) caused by natural imperfections is solved by algorithms [4]. 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 [9-12]. For example, according Wolf et al. [12], the current car communication networks assure safety against several technical interferences, but they are mostly unprotected against malicious attacks. The attacks to the mechanism of synchronization of clocks can cause faults and risks of accidents. These attacks depend of a clock synchronization mechanism and the network used. In general, each system needs a policy of synchronization and a network that have their own vulnerability list.

Clock Attacks

The CSMA/CD is widely used in many systems. The Ethernet is a protocol which uses a CSMA/CD philosophy. There are some researches in the literature about the security aspects of systems using the IEEE 1588-2008 Standard [13], such as [9]-[11].

Table I, extracted from [9], exemplify a vulnerability list of attacks to clock synchronization in IEEE 1588-2008 Standard over Ethernet protocol, and the results of these attacks.

TABLE I – LIST OF ATTACKS TO IEEE 1588 - CLOCK SYNCHRONIZATION

	Attack	Result of Attack
1	Denial of service	no service available
2	Byzantine master	complete loss of control
3	Interruption of control loop	deviation determined by precision of local clock
4	Removal of packets from control loop	deviation determined by precision of local clock
5	Packet manipulation	complete loss of control
6	Packet insertion	offset up to sync cycle depending on implementation
7	Selective packet delay	offset up to sync cycle

Source: [9].

In the list of Table 1, the attacks that result in a complete loss of control or instability are: 2) byzantine master, and 5) packet manipulation, that can cause it directly; and the 3) interruption of control loop, and 4) removal of packets from control loop, that can cause it indirectly. Let us discuss them below briefly.

However, the applications that make use of the distributed architecture are growing. This architecture does not use a master clock in the clock synchronization process. Examples of protocols used in these architectures are the Time

Triggered Protocol (TTP) [14] and the FlexRay [12]. There, clock synchronization is assured by the Byzantine Theorem [15] expressed by Equation (1):

$$n \geq 3f + 1 \qquad (1)$$

where, n is the number of clocks in the system; and f the number of clocks with errors. Through this rule, it is possible to achieve clock synchronization even in the presence of (*f*) errors.

This distributed architecture has its own vulnerable list of clock attacks. However, Oliveira Junior and Souza (2011) [4] show that the distributed architecture can suffer some of the same attacks as the centralized architecture, such as: 1) byzantine attack, 2) packet manipulation, 3) interruption of control loop, and 4) removal of packets of control loop. The difference is that, in the distributed architecture, the attacks do not focus on the master clock, but focus on the hypotheses of the Byzantine Theorem (15); *i.e.*, they aim to make them invalid. If this happens, then the clock synchronization process is impaired, and the virtual global time is not maintained. So, if the global time is lost the communication is degraded and hence the control system.

Furthermore, Table I shows the attacks and its effects over the Ethernet network. However, in practice, the same attacks have different effects over different networks. For each network, an investigation on the means to reach higher security is necessary.

IV. MODELS AND SIMULATIONS

In networked control systems, mainly in real time ones, clock synchronization is crucial to have a good timeliness of system. To exemplify how the imperfections (caused by Nature or by an External Malicious Agent (EMA)) of clock can cause a degradation of control over different networks, we studied by means of modeling and simulation the case of a networked control system using CSMA/CD and TDMA protocols for the communication network.

For this modeling and simulation, we used the TrueTime/Matlab/Simulink environment [16]. We simulated two sets of controls, *i.e.*, a system with two control loops connected by a common databus (CSMA/CD and TDMA). The sensors, actuators/plants and controllers were connected via the databus. The controller used was a PID (Proportional, Integral and Derivative). The actuator/plant was a second order marginally stable continuous time system, according to the following transfer function:

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

The controller and sensor nodes had logical clocks given by the virtual computer of the TrueTime Kernel; and they used the databus to exchange data and time measurements between them. The actuators/plant used the databus only to receive the control data. Each control node implemented a periodic control task and a periodic clock synchronization task. Each sensor node implemented a periodic task for sending the measured data to the controller; and a periodic task for synchronizing the clock. Each actuator/plant was activated by events when the control task arrived in the actuator by the databus. All nodes had an interruption caused by data arrived from the databus. The reference time is the virtual time given by the logical clock of the Matlab/Simulink environment. In the TrueTime Kernel it is possible to manage the virtual clocks. So, it is possible to insert faults (imperfections) and correct them. The EMA (External Malicious Agent) node implemented one task for changing, maliciously, the virtual clock in Sensor 1. The model of the simulated NCS is given at Figure 4:

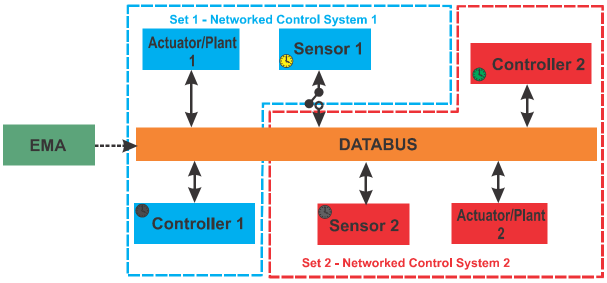

Figure 4. Networked Control System Model.

For the clock synchronization, we used a FTM (Fault-Tolerant Mid-Point) algorithm, also known as Welch-Lynch algorithm [7]. To ensure that all nodes have a consistent view of time we need to re-synchronize the clocks regularly (periodically). For this, the algorithm follows a logical sequence. Each node applies this sequence with the objective of reaching a correction term. With this correction term, the deviations caused by the drift of the clocks are adjusted so that all system clocks are within a certain precision.

V. RESULTS AND COMPARISONS

We simulated two control subsystems sharing the same databus as shown in Figure 4. In blue, we have control loop set 1 and in red we have control loop set 2. The sensors, actuators/plants and controllers are connected via the databus.

The EMA, in green in Figure 4, represents the External Malicious Agent, which was responsible by inserting a sum of 0.5 second in the time measurement of sensor 1 at instant 0. We call this attack as local clock manipulation. The objective of the simulation is to show how this attack: 1) directly degrades the synchronization between Controller 1, Sensor 1, Controller 2 and Sensor 2 even with all using a FTM algorithm in CSMA/CD and TDMA protocols; and 2) The differences of control responses of the two networks.

Table II shows the ideal and two worst cases simulated.

TABLE II – SIMULATED CASES

Case	Network	Error in Clock
Ideal	CSMA/CD or TDMA	No Error
Case 1	CSMA/CD	Initial Offset = 0.5
Case 2	TDMA	Initial Offset = 0.5

Ideal Case

In this case, we do not have an external attack in clocks and the result is the same in CSMA/CD and TDMA networks.

Figure 5 shows step responses of the system and Figure 6 shows their control laws. In blue, we have control loop set 1 and in red, we have control loop set 2. It is possible to observe that the step response of control loop set 1 in blue and control loop set 2 in red are very close.

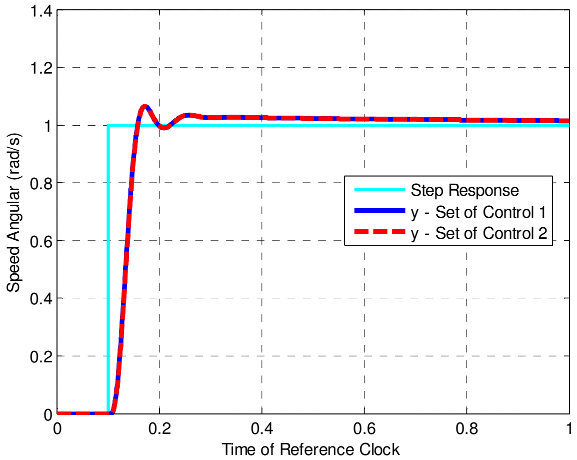

Figure 5. Ideal Case - Step Responses.

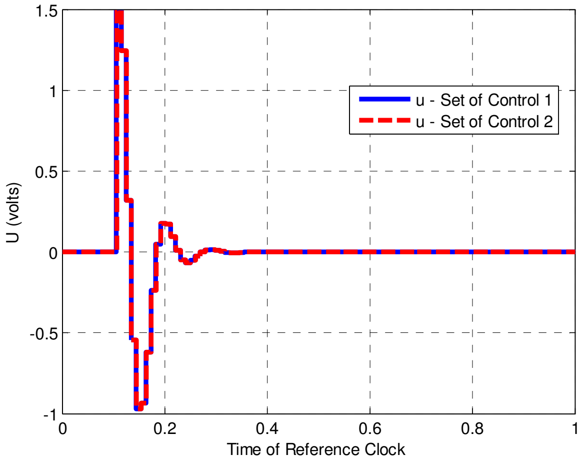

Figure 6. Ideal Case - Control Laws.

However, Figure 7 shows the difference between the step responses of both control loop sets. This difference is expected due to a delay caused by a network.

Figure 8 shows the times in nodes: Controller 1, Sensor 1, Controller 2 and Sensor 2. These nodes are used to achieve a clock synchronization using a FTM Algorithm. So, in Figure 8 we observe a consistent time view among 4 nodes in the network due to clock synchronization.

Now, in the control loop sets 1 and 2, we put an error in a Sensor 1 node and we can observe the degradation in step responses and controls.

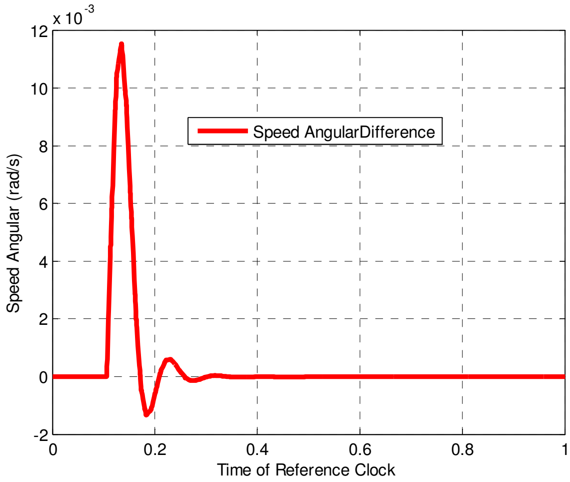

Figure 7. Ideal Case - Step Responses Difference.

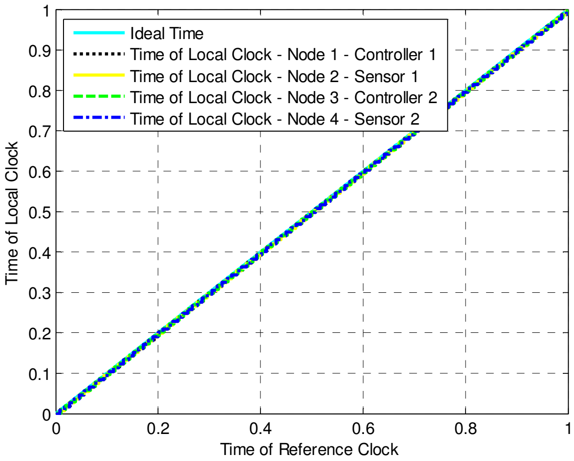

Figure 8. Ideal Case - Timelines.

Case 1 - CSMA/CD

In this case, we simulated an external attack by EMA in the clock of Sensor 1. The EMA was responsible by inserting a sum of 0.5 second in the time measurement of sensor 1 at instant 0 second. The objective of the simulation is to show how this attack: 1) directly degrades the synchronization between Controller 1, Sensor 1, Controller 2 and Sensor 2 using a FTM algorithm in CSMA/CD network; and 2) The control response due to this attack.

Figure 9 shows step responses of the system and Figure 10 shows their control laws in blue and red. Figure 11 shows the difference between the step responses of both control loop sets.

It is possible to observe that the step response of control loop set 1 in blue is delayed in relation to that of control loop set 2 in red. This occurred because after all nodes were synchronized together, the sensor 1 had already sent its tasks. So, sensor 1 is waiting a new task. Due to this, the control loop is temporarily opened. So, until a new task of measurement is executed, the system remains marginally stable. When a new task is executed, the control loop is closed again and the system becomes asymptotically stable.

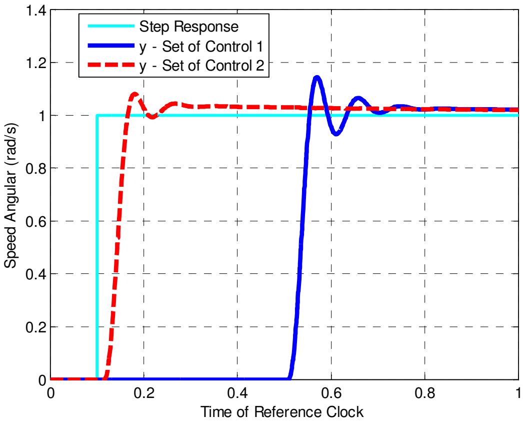

Figure 9. CSMA/CD Case - Step Responses.

Figure 10 shows the effect in the control laws and Figure 12 shows the timelines, where it is observed that the correction of the initial offset error with the FTM algorithm affects temporarily the clock of sensor 1. The correction generates a state error, *i.e.*, the clock of sensor 1 stopped temporarily, to recover the synchronization. However, in this case with CSMA/CD the system recovers the synchronization before the transition of the step response. After all nodes were synchronized together, sensor 1 had already sent its tasks. So, sensor 1 is waiting a new task. The new task entered after the step transition. Due to this, the step response is delayed as shown in Figure 9 and Figure 11.

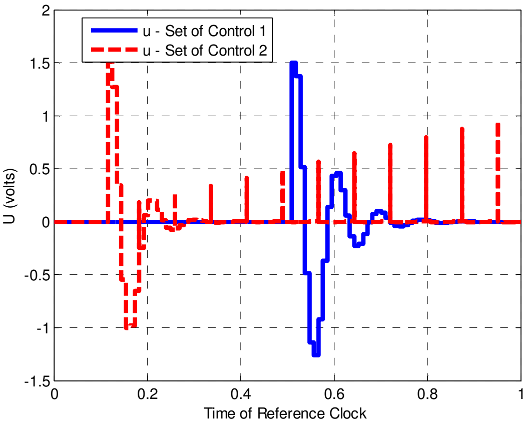

Figure 10. CSMA/CD Case - Control Laws.

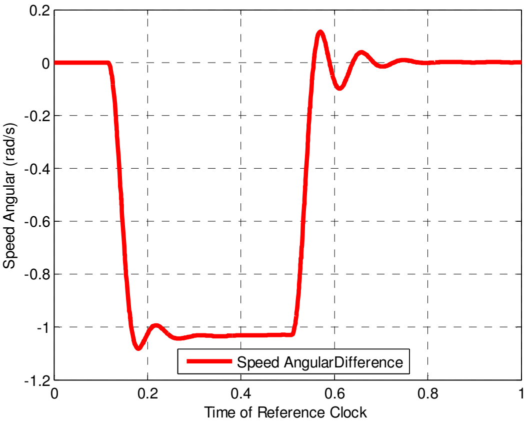

Figure 11. CSMA/CD Case - Step Responses Difference.

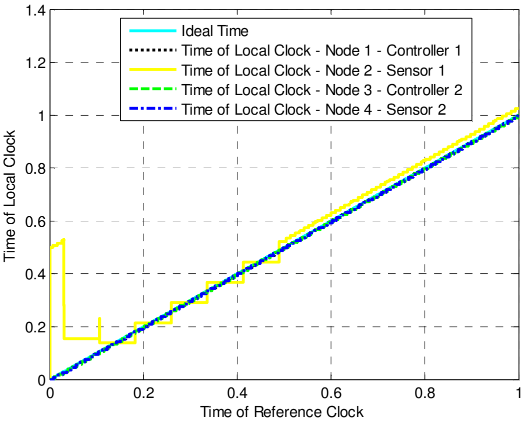

Figure 12. CSMA/CD Case - Timelines.

Case 2 - TDMA

In this case, we repeat the previous attack simulation of case 1, with the difference that we simulated the system with a TDMA network. The objective of the simulation is to show how this attack: 1) directly degrades the synchronization between Controller 1, Sensor 1, Controller 2 and Sensor 2 using a FTM algorithm in TDMA network; and 2) The control response due to this attack.

Figure 13 shows step responses of the system and Figure 14 shows their control laws in blue and red. Figure 15 shows the difference between the step responses of both control loop sets.

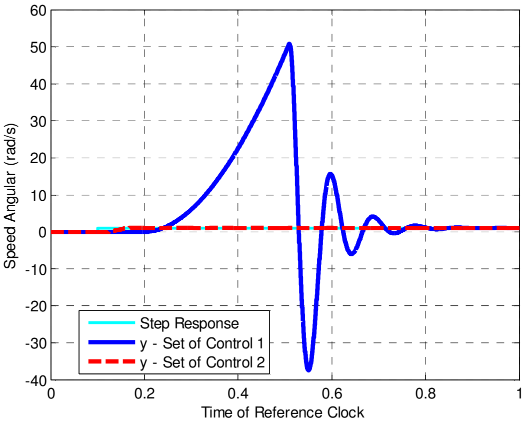

Figure 13. TDMA Case - Step Responses.

After all nodes were synchronized together, sensor 1 had already sent its tasks. So, sensor 1 is waiting a new task. Due to this, the control loop is temporarily opened. So, until a new task of measurement is executed, the system remains marginally stable. When a new task is executed, the control loop is closed again and the system becomes asymptotically stable.

Figure 14 shows the effects in the control laws and Figure 16 shows the timelines, where it is observed that the correction of the initial offset error with the FTM algorithm affects temporarily the clock of sensor 1. The correction generates a state error, *i.e.*, the clock of sensor 1 stopped temporarily, to recover the sync. However, in this case with TDMA, the system recovers synchronization after the transition of step response. After all nodes were synchronized together, sensor 1 had already sent its tasks. So, sensor 1 is waiting a new task. Due to this, the control loop is temporarily opened as shown in Figure 13. This occurs because of the time spent in synchronization of TDMA is greater than in CSMA/CD.

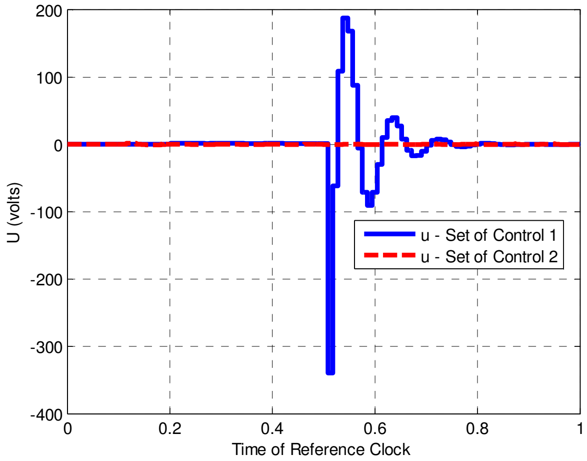

Figure 14. TDMA Case - Control Laws.

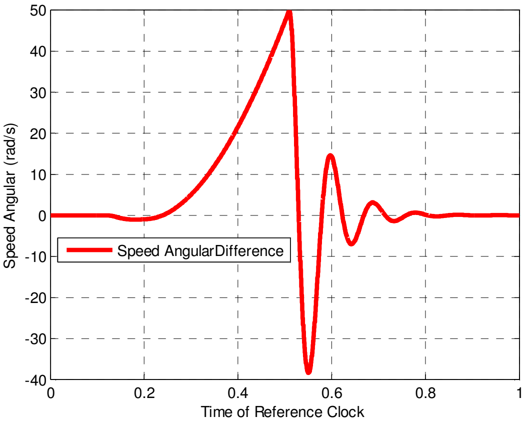

Figure 15. TDMA Case - Step Responses Difference.

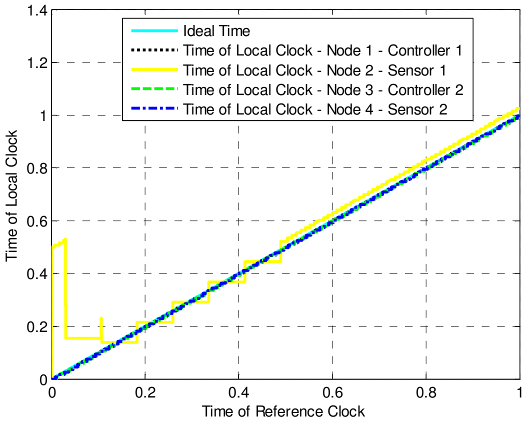

Figure 16. TDMA Case - Timelines.

TDMA VERSUS CSMA/CD

Cases 1 and 2 are the most relevant results found during the simulations. Figures 9-10 and 13-14 show that even synchronizing the clocks with the FTM algorithm, the external attack on the clock can affect the step responses and control laws of the system.

As shown in Figure 4, the task of a sensor 1 measures and send the data via the communication network to a control 1. In the case 1 with a CSMA / CD, due to clock attack in Sensor 1, the tasks are sending in advance but, because of network characteristics, it does not suffer task suspension. So, the delay is small and the system recovers a sync before the step transition. Thus, the system does not recognize the step transition but, since sensor 1 stops to recover a sync, the step response is delayed.

In the case 2 with a TDMA, due to clock attack in Sensor 1, the tasks are sending in advance but, because of network characteristics, the tasks of a sensor 1 will be suspended when its time is exceeded. This suspension task generates a large delay. So, the delay is large and the system recovers a sync after the step transition. Thus, the system recognizes the step transition but, since sensor 1 stops to recover a sync, the control system operates in open loop and thereby cause the overshoot of Figure 13.

From this brief comparison of cases, it can be concluded that: a) the TDMA communication network is much more susceptible to errors of the clocks than a communication bus with CSMA / CD, 2) Since the CSMA / CD is not be activated by time, it has a better allocation of the clock uncertainties, 3) the FTM algorithm does not satisfactorily correct errors with the initial offset.

VI. CONCLUSIONS

The security aspects of clock synchronization are becoming important for research and development of complex and/or integrated systems. Technically, the problem of clock synchronization caused by its natural imperfections can be solved within a tolerance by algorithms and methods. But socially, the overall functionality of those systems can be degraded or even disabled if the synchronization of the clocks is attacked.

The attacks to clocks can be made by numerous ways, such as byzantine attack, packet manipulation, interruption of control loop, removal of packets of control loop, local clock manipulation and others. These attacks aim to manipulate the times of clocks of systems, and trick them. The attacks to clocks depend of the clock synchronization architecture, methods and network philosophy used. So, during design, each system shall be analyzed: 1) to establish its own vulnerability list; and 2) to choose its security mechanism to be implemented to minimize these vulnerabilities.

In a NCS with:

- CSMA/CD network, the clock synchronization is not crucial to achieve a communication. However, if the system is designed to establish a deterministic communication and it is attacked, then the control is degraded, but the communication is not dependent of time. The degradation observed can cause a delayed step response.

• TDMA network, the clock synchronization is crucial to achieve a deterministic communication. So, if the temporal correctness is attacked, then the communication and control is degraded. The degradation observed can cause an accident due to the big overshoot in step response.

The preliminary results based on the case studied here suggest that: 1) the local clock manipulation attack to a sensor clock in a NCS affects the clock synchronization, and then, the communication and control; 2) the TDMA protocol is more deterministic, but an attack to the time management and/or clock synchronization can cause an accident due to the opening of the loop temporarily; 3) the CSMA/CD protocol is non-deterministic, but it is more robust to an attack in the time management and/or clock synchronization; 4) both cases need countermeasures for attacks to the clock synchronization to prevent accidents; 4) The FTM algorithm used prevents the worst effects due to clock drift but not due to initial clock offset.

Suggestions of Countermeasures

Based on these conclusions, Table III summarizes the main effects of attacks to clocks of CSMA/CD AND TDMA networks:

TABLE III – MAIN EFFECTS OF ATTACKS TO CLOCKS OF CSMA/CD AND TDMA NETWORKS

Protocols	CSMA/CD	TDMA
Examples	Ethernet	TTP and FlexRay
Exposure	Big	Acute
Possible Harms	Degrade and Delayed the Control	Degrade the control and Risk of accidents

To reach the required security goals, we suggest installing various countermeasures on various levels [5, 7], as:

• Cryptography of time messages transmitted, *i.e.,* introduction of cryptographed measures of time in the clock synchronization process;

• 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;

• Algorithms to identify and prevent any abrupt change in time;

• Security measures by hardware timestamp, *i.e.*, include a new layer in the clock synchronization process;

• Authentication of external messages;

• Introduction of real time clock, as a GPS, to provide a real global time to increase a clock synchronization reliability;

REFERENCES

[1] J. Tsang, K. Beznosov, "A Security Analysis of the Precise Time Protocol (short paper)", In Information and Communications Security, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2006, vol. 4307, no.10, p..50-59.

[2] SAE, "Certification Considerations for Highly-Integrated or Complex Aircraft Systems," Aerospace Recommend Practice ARP-4754, SAE, Nov. 1996.

[3] J. A. Stankovic, "Misconceptions about Real-Time Computing – a Serious Problem for Next-Generation Systems", IEEE Computer, vol. 21, no.10, pp.10-19, October 1988.

[4] E. M. Oliveira Junior, M. L. O. Souza, " A Brief Discussion of Security Aspects of Clock Synchronization in Networked Control Systems", III Simpósio de Aplicações Operacionais em Áreas de Defesa, ISSN 1983-7402, São José dos Campos, Brazil, 2011.

[5] D.G , Messerschmitt, "Synchronization in Digital System Design", IEEE Journal on Selected Areas in Communications, vol. 8, p. 1404, October 1990.

[6] H. Kopetz, "Real-Time Systems: Design for Distributed Embedded Applications", 1st. ed., Kluwer Academic Publishers, 1997.

[7] J. Lundelius, and N. Lynch, "A New Fault-Tolerant Algorithm for Clock Synchronization", Third Annual ACM Symposium on Principles of Distributed Computing, Vancouver, Canada, 1984.

[8] E. M. Oliveira Junior, M. L. O. Souza, "The Effects of Initial Offset Errors on Clock Synchronization of Networked Control Systems." In: XVII SAE Brasil Fair and Congress, São Paulo. SAE Brasil, 2010.

[9] G. Gaderer, A. Treytl, T. Sauter, "Security Aspects For IEEE 1588 Based Clock Synchronization Protocols", IEEE International Workshop on Factory Communication Systems - WFCS 2006, Torino, Italy, September 2006.

[10] J.-C. Tournier, O. Goerlitz, "Strategies to Secure the IEEE 1588 Protocol in Digital Substation Automation", Fourth International Conference on Critical Infrastructures, CRIS 2009, Sweden, Linköping, April 2009.

[11] A. Treytl, G. Gaderer, P. Loschmidt, N. Kerö, "Investigations on Security Aspects in Clock Synchronized Industrial Ethernet", 38th. Annual Precise Time and Time Interval (PTTI) Application and Planning Meeting, Washington, USA, December, 2006.

[12] M. Wolf, A. Weimerskirch, and C. Paar. "Security in Automotive Bus Systems." In Proceedings of the Workshop on Embedded Security in Cars 2004, 2004.

[13] 1588-2008 - IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, c1 – 269, July, 2008.

[14] TTTech Computertechnik. "Time-Triggered Protocol TTP/C High-Level Specification Document Protocol", version 1.1. Ed. 1.4.3. Vienna, 2003. (D–032–S–10–028).

[15] L. Lamport, R. Shostak, M., Pease. "The Byzantine Generals Problem", ACM Transactions on Programming Languages and Systems, Vol. 4, No. 3, July 1982, Pages 382-401, 1982.

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