On the Observability in Switched Ethernet Networks in the Next Generation of Space Launchers: Problem, Challenges and Recommendations

Jérémy Robert, Jean-Philippe Georges, Thierry Divoux
Université de Lorraine, CRAN, UMR 7039,
Vandœuvre-lès-Nancy, France
CNRS, CRAN, UMR7039, France
firstname.name@univ-lorraine.fr

Philippe Miramont, Badr Rmili
CNES - Direction Des Lanceurs
Paris, France
firstname.name@cnes.fr

***Abstract*—Nowadays, many embedded systems use specific data buses to ensure the exchange of data. To reduce the financial cost, the mass and to increase performance in keeping at least the same reliability, a solution is to rely on a components off-the-shelf (COTS) technology. As switched Ethernet is a well-known solution and widely implemented, this technology is studied for the next generation of space launchers. In this paper, we focus on the observability issue defined as, not simply network management system techniques, but as the ability to monitor the satisfaction of the application quality of performance (especially in terms of time constraints and frames sequence). It consists to obtain a real picture of the communications at any given time and location. In a conventional communication technology (i.e., specific buses), it is easy to collect all exchanges on the physical wire with a dedicated device. But, it is not possible anymore on a switched network. Many monitors are therefore implemented and have to be synchronized. Hence, this paper aims at highlighting the implementation challenges that we have faced in our experimental test bench mainly in coping with online synchronization. Some recommendations on synchronisation and multi-monitoring issues are therefore submitted for the future developments.**

***Keywords*-Ethernet networks; observability; time synchronization; real-time.**

I. INTRODUCTION

Traffic monitoring can be the cornerstone for understanding communication networks. The monitoring activity aims at collecting from the various network devices a set of relevant data. This enables to characterize the network state and therefore to identify unusual network behavior. According to the application domain, the purposes of the monitoring can also be different like network management [1], network security [2], network performance analysis [3], etc. The monitoring mechanisms depend directly on the intended application and also on the nature of the observed system.

In the paper, the system to monitor is the switched Ethernet network (as shown in Figure 1), which could be embedded into the next-generation of the space launchers ([4]). This component off-the-shelf (COTS) technology is aimed at replacing the current MIL-STD-1553B [5] (for control traffic) and Controller Area Network (CAN, for telemetry traffic) buses embedded in the european (unmanned) launchers. Figure 1 gathers terminal nodes to be used in a scenario where control and telemetry traffics are performed on a single network. In this application, relevant data is (at least) the full packet capture. In general, a new technology is only considered in many applications such as space [6] or automotive [7] if (and only if) this monitoring feature is satisfied. This study is led in the framework of a "CNES french Research & Technology (R&T) activity".

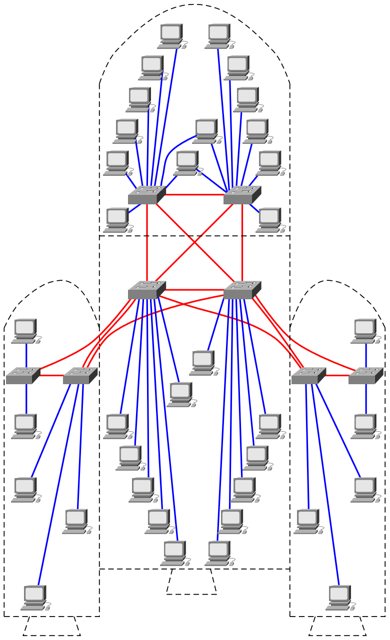

Figure 1. Switched Ethernet Architecture suggested in [4] for the next-generation of space launcher

In space applications (aircrafts, satellites, launchers), conventional communication technologies rely mainly on a specific bus, which is a unique physical medium (potentially redundant for the reliability [8]). As all end-nodes are connected to the same physical wire, each frame is observable to each of them. This is an important ability since a single dedicated device, a so called monitor, enables therefore to collect all exchanges along with a timestamp and to writte them into a trace (a real picture of the

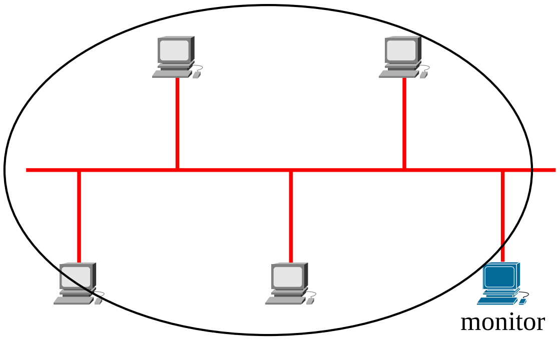

Figure 2. Global monitoring domain based on a unique observation point

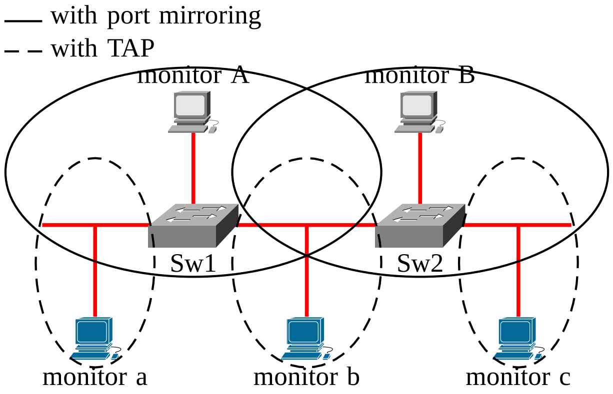

Figure 3. Distributed monitoring architecture based on multiple local monitoring domains

communications at any given time). This device constitutes a unique observation point of the network, which is necessary and sufficient to meet the monitoring traffic requirement for space applications. Accordingly, the monitoring domain covers the whole entire architecture as shown in Figure 2.

In a switched network, all end-nodes are inter-connected with several switches. Regarding the switch operating, the traffic is confined to different segments (to each link between switches) and eventually forwarded. As a consequence, to obtain a real picture of the communications as previously (on a shared medium), many monitors have to be implemented. A distributed monitoring architecture is therefore needed to cover the whole network (cf. Figure 3). Each monitor (the number and the location of these devices is discussed hereafter) generates locally a trace. The issue of distributed (monitoring) applications is to retrieve location and ordering of events (e.g., emission/reception of a frame on a device before an other one), which happens on the network architecture. Indeed, different messages in the traces have to be linked with a strict ordering relationship. However, the clocks in each monitor are initially running asynchronously and may produce significant offsets. To merge all the local traces, it needs a global reference time with synchronisation offsets have to be as small as possible. The underlying question is therefore the time synchronization method [9].

Let us remember that the objective is to obtain the highest fidelity picture of the communications in order to analyze the real network behavior. The analysis is performed offline, after tracing is finished. The aim of the paper is to highlight the implementation challenges that we face in our switched Ethernet experimental test bench and the consequences for the next generation of space launchers.

The remainder of the paper continues as follows: the section II reviews the related work and the problem overview. This is followed by a description of the proposed monitoring architecture and implementation challenges in Section III. Discussions and recommendations are given in Section IV. Section V presents the challenges to pass from the traffic monitoring to the control state observability. Finally, in section VI some conclusions and future work are given.

II. RELATED WORK AND PROBLEM OVERVIEW

In a switched Ethernet network, the monitoring architecture is distributed. The number and the location of the monitors depend on the selected concepts (and also application requirements). Indeed, there exist several different techniques to capture network traffic. A point-to-point link can be splitted with a special device, named network Test Access Point (TAP) which enables to connect a monitor on this particular link. The traffic is also copied to this monitor in a passive way. Many manufacturers suggests this type of products as NetOptics®[10] or Fluke Networks[11]. A second method, called port mirroring, consists of using a special switches function (available on the most of commercial switches), which enables to copy all traffic coming from all or part of ports to a dedicated port. Figure 3 shows these different methods on a simple example where the dashed lines represent the observation domains for the TAP technique and the solid lines those for port mirroring technique.

Whatever is the solution retained for traffic monitoring, all monitors must have the same reference time to be able to make conclusions and recommendations on the network behavior (usual and unusual events). However, the clocks of each monitor produce time-varying offsets (because of clock drift), which are different from one another. This clock drift can be limited by using a synchronisation protocol as Network Time Protocol (NTP) or IEEE1588 - Precision Time Protocol (PTP). Some work (mainly, in a operating system tracing) suggest to rely on offline synchronisation by using a post-processing algorithm. These algorithms are mainly based on regression analysis (linear, least-squares, convex hull, etc.) [12] or linear programming [13]. The choice of the concept depends on the required performance which will be discussed in the following.

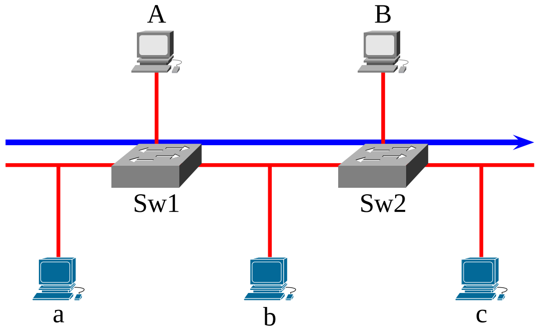

Figure 4. Illustration of synchronisation constraints

Let us remember that the ordering of events have to be retrieved from the analysis of traces. For example, Figure 4 shows a flow crossing respectively two switches Sw1 and Sw2. The flow must be captured on the monitor A (or a) before being it on the monitor B (or b and c) in the case of port mirroring technique (or in the TAP technique). As a consequence, a synchronisation performance constraint has to be defined in order to be sure that this ordering relationship can be observed. This constraint corresponds to the maximum offset between two monitors off_{max} (A and B, or a and b or b and c) and depends on the network parameters: the transmission time τ and the propagation delay δ (which can be negligible on the short Ethernet links). It can be expressed as $off_{max} < \tau + \delta$ with $\tau = \frac{min(L_{frame})}{C}$ where C corresponds to the link capacity and L_{frame} to the length of the Ethernet frames. The impact of the network parameters is discussed hereafter.

III. MONITORING ARCHITECTURES AND IMPLEMENTATION ISSUES

Our research laboratory collaborates closely with the CNES to lead R&T activities. In this framework, a certain level maturity of switched Ethernet technology has to be reached for the next generation of space launchers. This level can be assessed according to the Technology Readiness Level (TRL) [14]. In this collaboration, the objective is to reach the TRL4. Here, the aim is to constitute a ”proof-of-concept” on the ability to monitor all traffic.

For this purpose, an experimental test bench has been implemented in a laboratory environment (i.e., without being in an operational environment and without space components, but with a set of launcher representative data) as shown in Figure 5. It is composed of 8 switches Cisco IE3000 [15] and 100 Raspberry PI as end-nodes. On this topology, it has been deployed our monitoring architecture consisting of 8 computers (1 per switch) with Linux as operating system. Each monitor implements a special hardware card for time synchronisation: a Meinberg PTP card (PTP 270 PEX model) [16] . This card has been designed to add precise timestamping capabilities to data acquisition and measurement applications. The trafic is captured using the tcpdump[17] library. And the port mirroring technique has been chosen to minimize the number of additional devices.

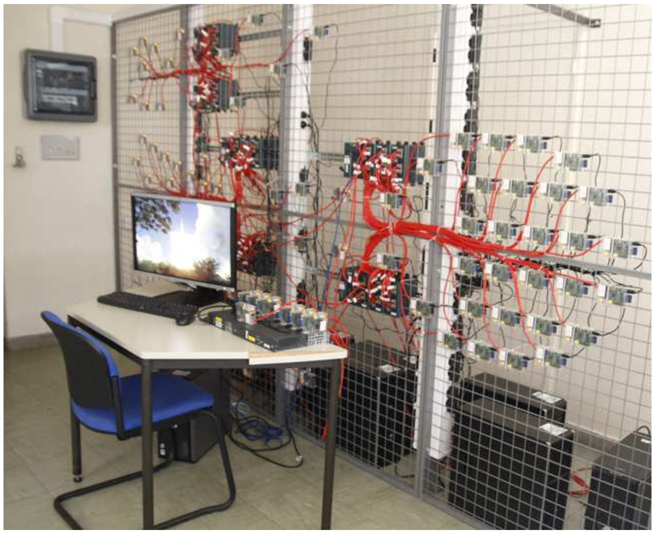

Figure 5. Our experimental test bench

In this framework, we face in many technical constraints to implement the monitoring architecture. The first one is that the PTP card can not be used as a standard network interface card. As a consequence, a second Ethernet link has to be used to monitor the traffic sent by the switch (via the port mirroring). On the other hand, the port mirroring can transmit only the copies of sent and received traffic for all monitored source ports. It therefore could not have been used to synchronise the monitor. The monitors are connected to a switch by two Ethernet links.

The second constraint concerns the timestamping of the captured frames. Indeed, the timestamping uses the date of the kernel clock and not the one of the PTP card (cf. tcpdump operation) as shown in Figure 6. As a consequence, a local synchronization is needed to enables to synchronize

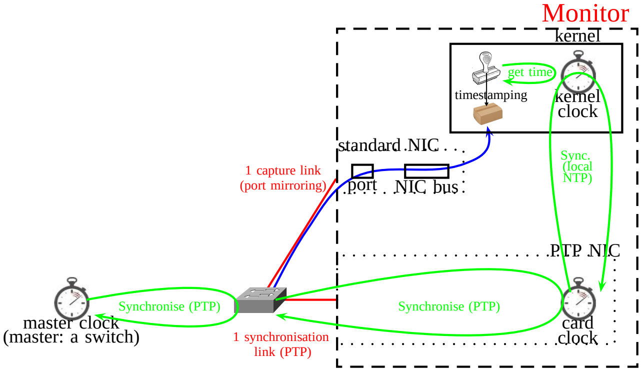

Figure 6. Experimental timestamping mechanism

ISBN: 978-1-61208-397-1

the kernel clock with the PTP card. To do that, NTP at stratum 0 is therefore used (to our knowledge, PTP can not be implemented locally yet). NTP is a protocol initially suggested by [18] for synchronizing the clocks of computer systems over packet-switched data network. It is based on a client-server model. To synchronize its clock with a server, the client computes the round-trip time and the offset from several measured timestamps (server's/client's timestamps of request/response packet transmission and reception). The timestamping remains on NTP at the high level. Hence, it is not related to specific hardwares like in PTP. The performances of the two control loops (as shown in Figure 6) have been measured on each monitor. Figures 7 and 8 represent the variation of the offset from master measured on a given day (without experimentations) for the PTP and NTP loop.

In brief, the PTP offsets are ranged between $-300\ ns$ and $300\ ns$ and those of NTP between $-40\ \mu s$ and $40\ \mu s$. These graphs highlight that the offsets of NTP are greater than those of PTP. For NTP, the variations are all the more important as the Central Processing Unit (CPU) load increases (e.g., when tcpdump is used).

In this network, all links are configured with a $100\ Mbits/s$ capacity. As a consequence, to be sure to detect the ordering of events with a minimum Ethernet frame ($72\ octets$), the offset between two monitors must be inferior to $5.76\ \mu s$. In our case, the offsets between two monitors can be $80\ \mu s$ at worst ($40\ \mu s$ from the master for monitor 1 and $-40\ \mu s$ from the master for monitor 2). As a consequence, it is clear that this is not possible to detect the ordering of events in a consistent manner. However, some temporal results have already been achieved with this monitoring architecture. Indeed, if the observation of events are not linked to many monitors, then this monitoring architecture is suitable for that. For exemple, the temporal respect of the events sequence (to a single destination and crossing a unique switch) has been verified.

Although this study shows that this implementation is not currently and directly applicable to traffic monitoring in space applications (because of NTP loop only), it is nevertheless possible to submit many recommendations to the future developments.

IV. DISCUSSIONS / RECOMMENDATIONS

In this work, the set of tools are turnkey solutions, this means that no specific development have been done. A monitor and the function ”port mirroring” in the switch constitutes here a prototype of the function ”trafic monitoring”.

The aim of this section is therefore to present some possible evolutions and/or recommendations for the future development.

To validate in a definitive manner our ”proof-of-concept”, here are some obvious evolutions, which could be applied in our experimental test bench:

- other type of switch with timestamping capabilities (at the mirroring port) could also be used (e.g., Cisco Nexus). At the time of the choice, these devices were not available yet.
- to avoid the NTP loop on each monitor, a homemade tcpdump could be developed to timestamp directly all collected frames with the PTP card clock. It is important to note that this solution is really feasible.

The devices used in the experimental test bench will be not embedded as is in the space launcher. But, if we look at the space news, we can see that many Ethernet switches begin to be used in space program (e.g., Hewlett Packard switches on-board ISS (International Space Station) [19]) or begin to be rugged for space environment in the launch vehicle (e.g., Cisco IE 3000 switches for the Atlas and Delta IV [20]). All devices are based on COTS, and industrials refer to a R-COTS (Rugged-COTS) or M-COTS (Modified-COTS).

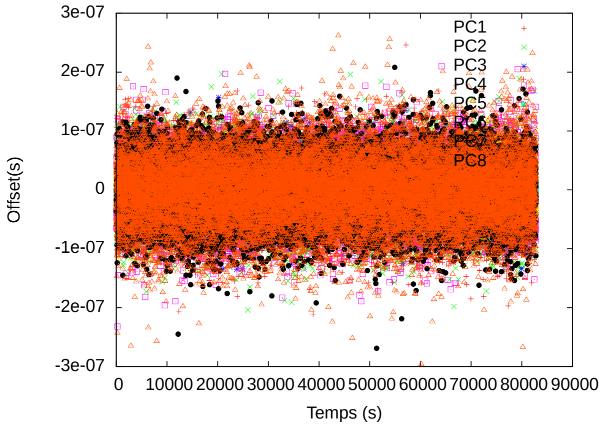

Figure 7. *Offsets* PTP measured on a day

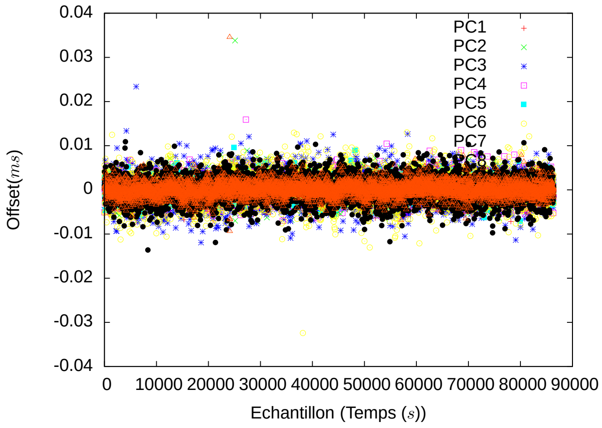

Figure 8. *Offsets* NTP measured on a day

ISBN: 978-1-61208-397-1

In any case, it seems that all switches still implement the port mirroring function (e.g., TTEthernet switches, Aitech S750 Radiation Tolerant switches, etc.). Developing a homemade switch rather than relying on commercial products might be long and expensive, especially regarding memories for mechanisms like mirroring. For instance, it takes 3 years for HP to develop new switches for the International Space Station.

As a consequence, in a short term, it is clear that it will be interesting to develop quickly a solution for timestamping directly the frame with the PTP card clock. Then, in design phase, it will be necessary to study the total quantity of traffic which is copied from all monitored source ports to the port mirroring. Indeed, the bandwidth of this port is limited and it can become congested.

On the other hand, we can see that the synchronisation constraint is all the more when the network capacity increases. Consequently, it is possible that the mere use of synchronisation to satisfy the trafic monitoring requirement is not sufficient. Others methods need to be designed to face this limitation. A track will be to consider offline synchronization by using the knowledge of network events (since that space applications are often deterministic).

V. FROM NETWORK TRAFFIC MONITORING TO THE CONTROL STATE OBSERVABILITY: CHALLENGES

With traditional buses for launchers, the network testing mainly relies on traffic monitoring. It mainly consists in capturing all frames from a single capture point. It is useful to check if packets losses occur (network QoS) and also to know the current static launcher control state (application quality of performance like the information promptness and the arrival ordering). The on-board controller manages this control state by sending specific data, called *control words*, to the sensors/actuators. By analyzing the content of the packet, it is hence possible to retrieve the *control word* values measured by the sensors and those sent by the controller to the actuators. Hence, the network acts as an observer of the control state.

All those control words are related to the different dynamics of the launcher control. A control step is defined by a sequence $\mathcal{S}$ of application control words w_i with $\mathcal{S} = \{w_1, w_2 \dots w_n\}$. The key point is now to develop strategies to monitor how a given sequence (and not only a frame) will be served in time by the next generation of networks. For each word w_i, the control application will define a target sending date t_i (relative to a reference time) with a tolerance δ_i. From the network point of view, each word corresponds to a single frame that has to be sent to a destination (not necessary the same for all words even if several may belong to the same transfert). The departure time of these frames may also be not periodic. Hence, the traffic monitoring should be able to observe these times and next, to check that all these requirements (order and tolerance) are satisfied. Next generation network, and in particular, switched Ethernet network, may however face two important issues:

Compared to buses, switched architectures do not permit anymore to capture from one single point the whole traffic (see Figure 3). To achieve this objective, it requires to add several capture points (based on TAP on each link or on port mirroring mechanisms on each switch). The synchronisation of these multiple captures have to be solved in order to test if the application sequence order and tolerance are satisfied. This first issue only deals for switched Ethernet network (like in native IEEE 802.1D or AFDX) and may not occur for Ethernet protocols that will be used on a bus.

The second issue that Ethernet protocols may introduce is related to the medium access policy. Even if at the MAC level, IEEE 802.3 defines a specific method, a lot of solutions add a middleware that change the access. For instance, with Modbus/TCP, it may corresponds to a Master/Slaves policy where only one frame is sent at a given time on the network. For legacy switched architectures, it means that several frames may be simultaneously forwarded around the network. As a consequence, the frames order may change and a given frame may be captured at different dates and locations by several monitors.

We define here the observability as the ability to determine dynamically how the sequence requirements are satisfied. A question might be *is the word* w_j *successfully forwarded by the network at the time* $t_j \pm \delta_j$. Even if multiple (network) observers are used, a centralized overview of the current frames exchanged by the network has to be determined (this centralized overview is important for launchers where the control state information have to be transmitted to the ground via the telemetry channel). In the following, we will develop such challenges for two example of space solutions: AFDX [21] and TTEthernet [22].

Avionics Full-Duplex Switched Ethernet (AFDX) relies on the exact bandwidth regulated traffic control to guarantee a determinist service. Thanks to the notion of Virtual Links (ARINC 664, part 7), a channel is opened between a source and a destination and is characterized by a minimal time between two consecutive frames (Bandwidth Allocation Gap). As the name suggests, this technology relies on a switched topology. As a consequence, AFDX solution has to face to the synchronisation issue of the multiple captures (obtained on several monitors) as seen previously. On the other hand, many frames may be sent on the network on the same time. The middleware enables to guarantee only the bandwidth for a given flow and not its order relatively to an other.

TTEthernet is a time-triggered Ethernet solution. It relies on time division multiple access (TDMA) for time-triggered communication (according to SAE AS 6802). The aim is to ensure predictable transmission delays without queuing, and therefore low latency and jitter. In this way, a unique frame is a priori on the network at a given time. However, this frame will be captured by several monitors at different dates as the topology is a switched one. Although the TDMA mechanism may guarantee the order (if these traffic flows are considered as time-triggered communication), it will be important to check the respect of tolerance. Indeed, as the target sending date is calculated during the flight (relatively to several events), it is possible that a sender has no access to the medium at this date (slot allocated to an other sender) and has to wait the next cycle. On the other hand, TTEthernet enables to use two others traffic classes: rate-constrained (ARINC 664, part 7), and COTS Ethernet (IEEE 802.3) traffic flows. Some sequences could be sent by using several frames belonging to these others traffic classes. As a consequence, no guarantees are given by the middleware and the same previous issues remain to handle.

VI. CONCLUSION

In this work, we face implementation issue in terms of synchronisation. However, the paper highlights that it is possible to lead quickly a proof-of-concept of traffic monitoring in switched Ethernet networks in the next generation of space launchers.

It is also important to note that the presented problem will be the same for any switched Ethernet technology (TTEthernet, AFDX, etc.), which could be retained for the next generation of space launchers. As a consequence, all solutions could benefit from the recommendations established in this paper.

REFERENCES

[1] K. G. Anagnostakis, S. Ioannidis, S. Miltchev, M. Greenwald, J. M. Smith, and J. Ioannidis, “Efficient packet monitoring for network management,” IEEE/IFIP Network Operations and Management Symposium (NOMS), 2002, pp. 423–436.

[2] R. Bejtlich, “The Practice of Network Security Monitoring: Understanding Incident Detection and Response,“ No Starch Press, 2013.

[3] R. Hofstede, I. Drago, A. Sperotto, and A. Pras, “Flow monitoring experiences at the ethernet-layer,” EUNICE Workshop on Energy-Aware Communications, 2011, pp. 129–140.

[4] J. Robert, J.-P. Georges, T. Divoux , P. Miramont, B. Rmili, “Toward self-reconfiguration of switched ethernet architectures in the next generation of space launchers,” International Conference on Advances in Satellite and Space Communications (SPACOMM), 2013, pp. 17–22.

[5] Department of Defense, “Military standard aircraft internal time division command/response multiplex data bus,” MIL-STD 1553B, 1978.

[6] J. Robert, J. Georges, T. Divoux, P. Miramont, B. Rmili, “Ethernet networks for real-time systems: application to launchers,” International Space System Engineering Conference (DASIA), 2011.

[7] A. Kern, H. Zhang, T. Streichert, and J. Teich, “Testing switched ethernet networks in automotive embedded systems,” IEEE International Symposium on Industrial Embedded Systems (SIES), 2011, pp. 150–155.

[8] D. Monchaux, P. Gast, and J. Sangare, “Avionic-x: A demonstrator for the next generation launcher avionics,” Embedded Real-Time Software and Systems (ERTS), 2012.

[9] L. Lamport, “Time, clocks, and the ordering of events in a distributed system,” Communications of the ACM, vol. 21, no. 7, 1978, pp. 558–565.

[10] http://www.netoptics.com, 2015.

[11] http://www.flukenetworks.com, 2015.

[12] A. Duda, G. Harrus, Y. Haddad, and G. Bernard, “Estimating global time in distributed systems.” International Conference on Distributed Computing Systems (ICDCS), vol. 87, 1987, pp. 299–306.

[13] S. B. Moon, P. Skelly, and D. Towsley, “Estimation and removal of clock skew from network delay measurements,” Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, vol. 1., 1999, pp. 227–234.

[14] “Iso/fdis 16290 - space systems – definition of the technology readiness levels (trls) and their criteria of assessment.”

[15] http://www.cisco.com/c/en/us/products/switches/industrial-ethernet-3000-series-switches/index.html, 2015.

[16] https://www.meinbergglobal.com/english/products/pci-express-ieee-1588-ptp-slave.htm, 2015.

[17] http://www.tcpdump.org, 2015.

[18] D. L. Mills, “Internet time synchronization: the network time protocol,” IEEE Transactions on Communications, vol. 39, no. 10, 1991, pp. 1482–1493.

[19] R. Schmidhuber, “International space station welcomes aboard procurve networking,”EADS Astrium Space Transportation White Paper, 2008.

[20] “Curtiss-wright awarded contract by united launch alliance,” Press-releases, April 10, 2014.

[21] “Arinc 664, aircraft data network, part 7: Deterministic networks.” 2003.

[22] “SAE AS6802: Time-triggered ethernet,” 2011.