# Safety for real-time Ethernet in IEC 61508 and IEC 61784

Ayoub Soury, Karem Hafsi, Denis Genon-Catalot, Jean-Marc Thiriet

# Safety for real-time Ethernet in IEC 61508 and IEC 61784 ⋆

**Ayoub Soury** * **Karem Hafsi** * **Denis Genon-Catalot** *
**Jean-Marc Thiriet** **

* *Univ. Grenoble Alpes, Grenoble INP* [1]*, LCIS, F-26000 Valence, France,*
*(e-mail: (ayoub.soury, denis.genon-catalot)@lcis.grenoble-inp.fr).*
** *Univ. Grenoble Alpes, Gipsa-lab, F-38000 Grenoble, France and CNRS, Gipsa-lab,F-38000 Grenoble, France,*
*(e-mail: jean-marc.thiriet@univ-grenoble-alpes.fr)*

**Abstract:** The real-time Ethernet in deterministic networks is far from being safe, it guarantees perfect synchronization among devices, and meets the real-time requirements but not the safety ones. To ensure these requirements, we need to identify the errors in the digital communication. In order to add safety measures in the communication system, this paper describes an analysis of IEC 61508 and IEC 61784. Furthermore, we propose to implement a safety layer over Real-Time Ethernet (at the top of the application layer) in the embedded real-time systems.

*Keywords:* Real-time Ethernet, safety, IEC 61508, IEC 61784, safety layer.

## 1. INTRODUCTION

Certainly the performances of industrial communication, in particular the real-time Ethernet (RTE) based protocols are necessary (e.g. response time, cycle time, etc.), but another important criterion which must be taken into consideration is the safety of data communication into these protocols. As said by Novak et al. (2007) and according to IEC 61508, safety means "absence of unacceptable risk of physical injury or damage to the health of people".

With all Ethernet applications, many variables affect the transfer of safety related data. Different devices are used to transfer data (e.g. switches) into a normal application. In addition, each application uses many internal stacks built into the disk or software. This normal application environment does not take safety related data into account.

To use existing applications, a special safety method can be built with two microcontrollers on the side of the transmitter of each device (physical redundancy) as seen in Fig. 1.

These microcontrollers (e.g. S_Controller 1 and S_Controller 2 in the safety sensor in Fig. 1) determine the sensor value in a completely independent way and monitor the functions of each sensor.

The first controller (S_Controller 1) generates the data format with the header, a consecutive number and the variable name before their transmission.

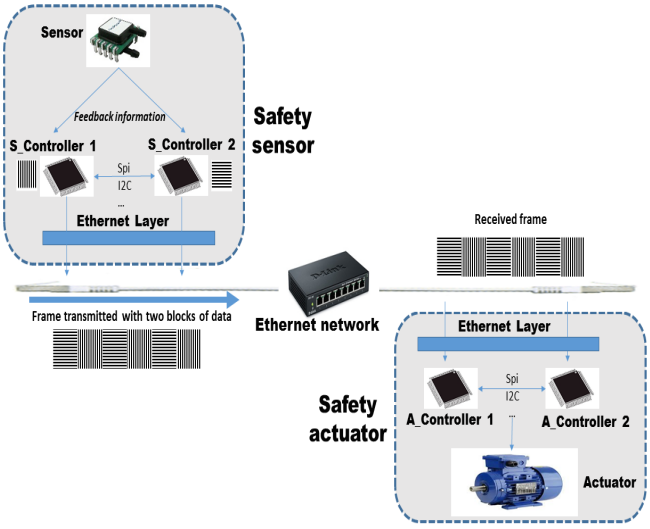


Fig. 1. Safety related data generation with physical redundancy

The second controller (S_Controller 2) inverts all its data before the transmission.

However, another option is that one of the controllers generates all the data, and the second controller then checks the content.

After all data has been generated by microcontrollers, Ethernet sends both data structures in the same frame. Then, the receiver performs the same process in reverse order. It transfers the two blocks to the controllers (e.g. A_Controller 1 and A_Controller 2 in the safety actuator in Fig. 1) that check and compare the data content.

⋆ This research is supported by a grant of the BGLE-ADN4SE project (Atelier de Développement et Noyau pour Systèmes Embarqués) supporting and funded by the French industry ministry and lead by Sherpa and Krono Safe companies. The authors want to thanks the members of workpackage lift.

[1] Institute of Engineering Univ. Grenoble Alpes

The separate data processing by these two microcontrollers will generate a time of delay. It makes this method not entirely cost-effective and not applicable for hypercritical systems, which require very short response times.

The OpenSAFETY protocol is designed to solve these types of problems (i.e. hardware duplication, cable wiring, transfer and processing time).

## 2. POSSIBLE ERRORS IN DIGITAL COMMUNICATION, AS IN IEC 61508 AND IEC 61784

In our case, we are using a safety layer to detect the failures in digital communication system which may be caused by:

- Wrong sequence associate with received message due to an error, fault or interference which altered the chronological order of information (IEC 61784 and IEC 61508).
- Corruption can happen by error on the transmission medium, or message interference and it will be not easy for receiver to detect non-valid information inside it (IEC 61784 and IEC 61508).
- Unintended repetition: Old not updated messages are repeated at the wrong time (IEC 61784 and IEC 61508).
- Loss : When a message is not received or not acknowledged, that causes a lack of information (IEC 61784 and IEC 61508).
- Unacceptable delay: Messages may be delayed beyond their permitted arrival time window, for example due to errors in the transmission medium, congested transmission lines, interference, in such a manner that services are delayed or denied (i.e. a message is not transmitted within the fault tolerance time) (IEC 61784 and IEC 61508).
- Insertion: A message is inserted that relates to an unexpected or unknown source entity (i.e. a fault of a bus participant that is not authorized to send message) (IEC 61784 and IEC 61508).
- Masquerading of non-safety-related message as a safety-related message: The data is inserted that relates to an apparently valid source entity, so a safety relevant participant, which then treats it as safety relevant, may receive a non-safety relevant message (IEC 61784 and IEC 61508).
- Addressing: A safety relevant message is sent to the wrong safety relevant participant, which then treats reception as correct (IEC 61784).

Dealing with this problem, Novak and Tamandl (2007) propose some safety measures for:

- Data corruption: use of CRC or data duplication with comparison;
- Loss of message: use of a watchdog;
- Insertion of message: use of a time-stamp and safe address;
- Delay, repetition, wrong sequence of messages: use of time-stamp;
- Unsafe message looks like a safe message: use of a specific header, specific addressing model.

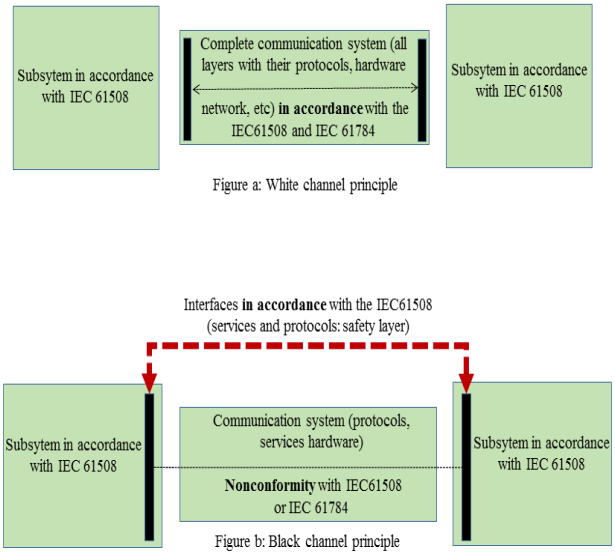


Fig. 2. The white Vs. black channel concepts according to IEC 61508

## 3. THE CONCEPT OF THE SAFETY COMMUNICATION LAYER

According to IEC 61508, we found two different approaches for safety data exchange, the black channel and the white one.

The white channel consists to verify integrity of all layers in the OSI model and in the hardware implied in transmission data as shown in Fig. 2.a.

The black channel concept is based on the safety layer (i.e. top of application layer) that is performed by additional safety transmission functions as can be seen in Fig. 2.b.

In fact, it abstracts the lower layers independently how the transmission system is built without having detailed knowledge about it.

In our case, the transmission system from one safe node to another (i.e. safety related communication) is based on the black channel principle, an analogy to a black box. A safety related communication in our networked system requires that:

- Information has to be corrected at the correct place and within the correct order;
- Information has to be available at the correct point in time;
- Safety-related and non-safety-related communications on the same channel must be independent, allowing the contemporary use of safety devices and standard devices.

## 4. REALIZATION OF SAFETY NODE AND THE CORRESPONDING MEASURES TO DETECT ERRORS

In order to remedy the errors described above, our safety stack implements the measures in order to detect and avoid falls into unsafe state (unsafe state means a situation that bears an unacceptable risk level according to IEC 61508).

*4.1 Safety measures according to IEC 61784 and IEC 61508*

To enrich the safety measures proposed by Novak and Tamandl (2007) in section 2, we detailed an additional safety measures according to IEC 61784 and IEC 61508.

- Sequence number: A sequence number is integrated into messages exchanged between message source and message sink,
  - ☐ detected errors={retransmission, loss, insertion wrong sequence};
- Time-stamp: In most cases the content of message is only valid at a particular point in time. The time-stamp may be a time, or time and date, included in a message by the sender,
  - ☐ detected errors = {repetition, wrong sequence, delay errors};
- Time expectation: During the transmission, the message checks whether the delay between two consecutively received messages exceeds a predetermined value. In this case, an error has to be assumed,
  - ☐ detected errors={transmission delay errors};
- Connection authentication: Messages may have a unique source and/or destination identifier that describes the logical address of the safety relevant participant,
  - ☐ detected errors={insertion into a message by a non-authorized sender};
- Feedback message or acknowledgement via an echo: The message sink returns a feedback message to the source to confirm reception of the original message. This feedback message has to be processed by the safety communication layers,
  - ☐ detected errors={loss, insertion, data corruption, data masquerading};
- Redundancy with cross checking: In safety-related fieldbus applications, the safety data may be sent twice, within one or two separate messages, using identical or different integrity measures, independent from the underlying fieldbus. If a difference is detected, an error shall have taken place during the transmission, in the processing unit of the source or the processing unit of the sink,
  - ☐ detected errors={repetition, loss, insertion, wrong sequence, data corruption};
- Data integrity assurance: The safety-related application process shall not trust the data integrity assurance methods if they are not designed from the point of view of functional safety. Therefore, redundant data is included in a message to permit data corruptions to be detected by redundancy checks. In Storey (1996) and Zammali (2016), the authors defined data integrity as the ability of system to detect or correct data errors during an exchange via network;
- Distinction between safety related (SR) and non-safety related (NSR) messages;
- Data protection.

In Soury et al. (2015a), we analyzed the classification of real-time Ethernet (RTE) solutions in order to choose the adopted approach for our system (i.e. lift control system that should be conformed with IEC 61508). We are interested in communication system and its safety on

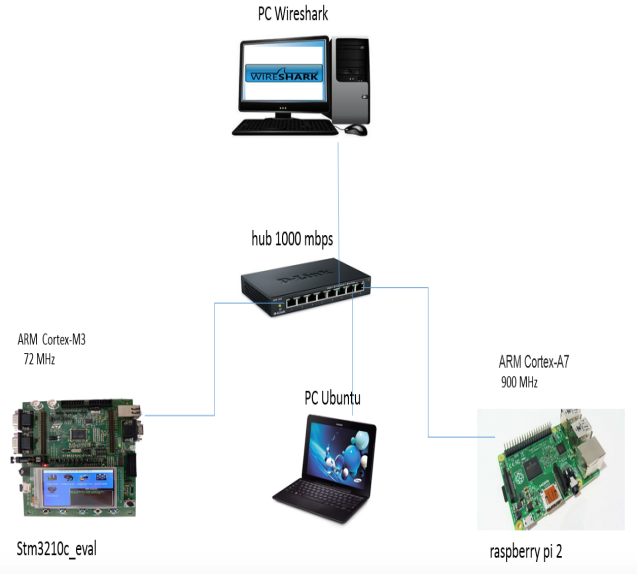


Fig. 3. The EPL network.

industrial cards (e.g. Industrial Communication Engine (ICE) AM3359, STM3210c eval, Rasberry pi 2 card, etc.).

In this paper we described the safety measures in the communication system which is capable to guarantee the SIL 3 according to IEC 61508. For implementation we used two protocols for data transfer:

- The Ethernet PowerLink (EPL) as RTE based protocol and we implemented functional safety to achieve the safety level required for one master and 2 slaves as shown in Fig. 3.
- The Ethernet for Control Automation (EtherCAT) as RTE protocol in simulation model (OMNet++ model) and we implemented a safety networked system with one master (PC) and one slave (ICE AM3359).

*4.2 The Ethernet PowerLink: EPL*

In Soury et al. (2015a) we realized a networked industrial system for the safety chain in lift control system. Communication among components (STM3210c eval cards and PC) was ensured through EPL protocol. In this paper, we continue with the same approach (using EPL in the safety chain). However, we changed the target networked system as shown in Fig. 3.

Actually, we are using two slave nodes (controlled node - CN) and a master node (MN):

- STM3210c eval card like CN-1,
- Raspberry pi 2 like CN-2,
- PC master.

In this system, the STM32 card (CN-1) has to send a feedback information to the MN by the network. PC (MN) sends an order to the second Raspberry card (CN-2). The temporal performance of the EPL networked system was evaluated in Soury et al. (2015a) as shown in equation 1, which calculates the cycle time for EPL network. This equation makes it possible to express the cycle time as a function of:

- The size of EPL message (i.e. $SoC$, $PReq$, $PRes$, $SoA$, $ASnd$);
- The number of slaves in the network ($n$).

$$Cycle\ time = t_{SoC} + \sum_{k=1}^{n}(t_{Preq_k} + t_{Pres_k}) + t_{Pres} + t_{SoA} + t_{ASnd} + 2(n+2) \times IFG \quad (1)$$

Theoretically, the EPL cycle has to reach = 886 $\mu s$ as shown in equation 2 with the following configuration:

- two slaves and one master, i.e. $n = 2$,
- without safety functions,
- with fast Ethernet (theoretical speed is 100Mbps),

$$Cycle\ time = t_{SoC} + \sum_{k=1}^{2}(t_{Preq_k} + t_{Pres_k}) + t_{Pres} + t_{SoA} + t_{ASnd} + 8 \times IFG$$

$$Cycle\ time = \frac{64 + \sum_{k=1}^{2}(1490 + 1490) + 1490 + 64 + 318}{9 \times 10^6} + 8 \times 10^{-6} \quad (2)$$

$$Cycle\ time = 886\ \mu s$$

In our EPL implementation (with STM32 card an rasberry as CNs), we reach a cycle time equal to 20 ms.

### *4.3 The Ethernet for Control Automation: EtherCAT*

A typical EtherCAT network consists of one master connected to one or more slaves, where the master controls the slaves and the slaves in turn can be in control of some functions that need to be managed, as shown in Fig. 4.

The principle is that all slaves nodes in an EtherCAT network can read/write data from/to the EtherCAT telegram as it passes (on the fly) by with only a short constant delay in each slave device (independent of the packet size). The constant node delay is typically below 500 ns. The telegrams are reflected at the end of each network segment (last slave node) and sent back to the master as shown in Fig. 5.

The EtherCAT node (master and slave) uses the standard physical layer defined by the standard Ethernet at 100Mbps and hence any PC equipped with a NIC (Network Interface Card) can run as an EtherCAT master. However, in practice an EtherCAT slave is implemented with special hardware (e.g. ICE AM3359 in our case) to facilitate very short packet forwarding delays (some nanoseconds) in the slave devices. In fact, the master node is normally implemented with standard components. This master generates and sends the EtherCAT telegrams using full duplex transmission.

EtherCAT supports many network configurations. But taking advantage of the one-frame-many-slaves concept, the topology has to be reducible to logical line, which can be just a simple line (in our case, the daisy chain topology is used as shown Fig. 4).

The temporal performance of the EtherCAT networked system was evaluated in Robert et al. (2014) as shown in

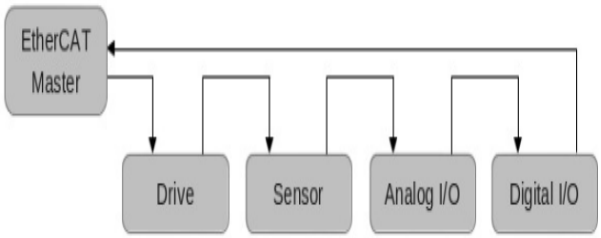


Fig. 4. The EtherCAT daisy chain topology.

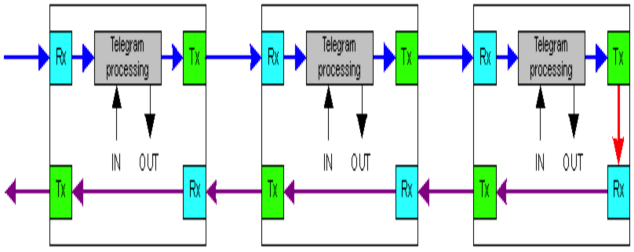


Fig. 5. The EtherCAT processing data.

equation 3 which calculates the cycle time for EtherCAT network. This equation makes it possible to express the cycle time as a function of:

- The network device latencies $l$;
- The payload size per device;
- The number of slaves in the network ($n$).

$$Cycle\ time = 2(n-1) \times l + 2n \times propagation\ delay + n \times t_{Payload} \quad (3)$$

For EtherCAT realization, we used the Texas instruments-Sitara AM335X ARM Cortex-A8 based card (ICE AM3359) as slave node. This ICE integrated the Programmable Real-time Unit (PRU) for real-time and critical tasks like communication. This technique allows us to save the cost of using an FPGA or an external ASIC. The temporal performance of the EtherCAT was evaluated in Prytz (2008), Cereia et al. (2010) and Robert et al. (2014).

Theoretically, the EtherCAT cycle has to reach = 15, 7 $\mu s$ as shown in equation 4 with the following configuration:

- one slave and one master, i.e. $n = 1$,
- without safety functions,
- with fast Ethernet (theoretical speed is 100Mbps),
- propagation delay = 50 nanosecond,
- latency = 1,35 micro second,
- payload = 100 bytes per slave + constant header size (40 bytes).

$$Cycle\ time = 2 \times propagation\ delay + t_{Payload}$$

$$Cycle\ time = 10 \times 10^{-9} + \frac{140}{9 \times 10^6} \quad (4)$$

$$Cycle\ time = 15,7\ \mu s$$

In our EtherCAT implementation (with the ICE AM3359 card), we reach a cycle time equal to 5 ms.

*4.4 Synthesis*

These theoretical values could not be achieved without using a specific hardware in the network components (master or slave node) as FPGA (Field Programmable Gate Array) or ASIC (Application Specific Integrated circuit) allowing very fast data processing.

We implemented following safety functions on RTE based networked system:

- sequential number,
- time-stamp,
- time expectation (only over EtherCAT protocol),
- identification,
- distinction between SR and NSR messages,
- data protection.

The safety functions implementation over RTE protocols (i.e. EPL and EtherCAT protocols) duplicates the cycle time previously calculated. Adding the safety function, will push the cycle time up to 40 ms for EPL protocol and 10 ms for EtherCAT protocol. These results meet the temporal requirements of the safety standard for lift control system PESSRAL (derived from IEC 61508, detailed in Soury et al. (2015b)) (Programmable Electronic components and Systems in Safety Related Applications for Lifts).

## 5. CONCLUSION

The communication system defines the technique for medium access, transmitting mechanism, telegram, etc. For safety applications, communication protocol must support safety requirements like IEC 61508 with safety functions and methods to detect transmission errors. But IEC 61508 did not specify the communication technologies.

This paper identified the possible errors in industrial communication system according to IEC 61508 and IEC 61784. We introduced and described safety functions used in digital communication protocols to meet the real-time and safety requirements of embedded critical systems. We propose to use the black channel concept to guarantee the safety in the communication system.

In this collaborative research project, funded by the French industry ministry, we use a deterministic real-time scheduling Krono-Safe technology patent (Chabrol et al. (2014)). For the ADN4SE project lift demonstrator, our contribution was to propose electronic modules supporting safety connected by real-time networked system instead of the original electromechanical safety chain switch of the lift control system.

The RTE protocols choice is driven by performance classification (detailed in Soury et al. (2015a)) excluding TT Ethernet solution which requires licensing -(ADN4SE project agreement).

Our contribution was to implement all the functional safety over real-time Ethernet-based protocol for embedded system with real-time and safety requirements all.

## REFERENCES


Cereia, M., Bertolotti, I.C., and Scanzio, S. (2010). Performance evaluation of an EtherCAT master using Linux and the RT Patch. In *Industrial Electronics (ISIE), 2010 IEEE International Symposium on*, 1748–1753. IEEE.

Chabrol, D., Barbot, A., and David, V. (2014). Dependable real-time system and mixed criticality: Seeking safety, flexibility and efficiency with kron-os. *Ada User Journal*, 35(04), 259–265.

Novak, T. and Tamandl, T. (2007). Architecture of a safe node for a fieldbus system. In *Industrial Informatics, 2007 5th IEEE International Conference on*, volume 1, 101–106. IEEE.

Novak, T., Treytl, A., and Palensky, P. (2007). Common Approach to Functional Safety and System Security in Building Automation and Control Systems. In *Emerging Technologies and Factory Automation, 2007. ETFA'07. IEEE Conference on*, 1141–1148. IEEE.

Prytz, G. (2008). A performance analysis of EtherCAT and PROFINET IRT Conference paper by Gunnar Prytz A performance analysis of EtherCAT and PROFINET IRT Gunnar Prytz Gunnar.Prytz@no.abb.com Abstract. In *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference*, 1396, 408–415.

Robert, J., Georges, J.P., Rondeau, E., and Divoux, T. (2014). Minimum cycle time analysis of ethernet-based real-time protocols. *International Journal of Computers Communications & Control*, 7(4), 744–758.

Soury, A., Charfi, M., Genon-Catalot, D., and Thiriet, J.M. (2015a). Performance analysis of ethernet powerlink protocol: Application to a new lift system generation. In *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, 1–6. IEEE.

Soury, A., Genon-Catalot, D., and Thiriet, J.M. (2015b). New lift safety architecture to meet pessral requirements. In *Web Applications and Networking (WSWAN), 2015 2nd World Symposium on*, 1–5. IEEE.

Storey, N.R. (1996). *Safety critical computer systems*. Addison-Wesley Longman Publishing Co., Inc.

Zammali, A. (2016). *Approche d'intégrité bout en bout pour les communications dans les systèmes embarqués critiques: application aux systèmes de commande de vol d'hélicoptères*. Ph.D. thesis, Universite Toulouse III Paul Sabatier.