Tuning of a simulation model for the assessment of Functional Safety over Wi-Fi

Alberto Morato
CMZ Sistemi Elettronici s.r.l.
and *Dept. of Information Engineering*
University of Padova, Italy
alberto.morato.3@phd.unipd.it

Giovanni Peserico
Autec s.r.l.
and *Dept. of Information Engineering*
University of Padova, Italy
giovanni.peserico@phd.unipd.it

Tommaso Fedullo
Dept. of Management and Engineering
University of Padova, Italy
tommaso.fedullo@phd.unipd.it

Federico Tramarin
Dept. of Engineering "Enzo Ferrari"
University of Modena and Reggio Emilia, Italy
federico.tramarin@unimore.it

Stefano Vitturi
National Research Council of Italy
CNR-IEIIT
stefano.vitturi@ieiit.cnr.it

***Abstract*—In recent years, Factory Automation is evolving towards the so-called Industry 4.0, and the creation of a smart factory ecosystem comprising of ubiquitously interconnected objects, namely the Industrial Internet of Things (IIoT), is gaining much research interest. This paradigm aims at developing new smart technological equipment and protocols, thus providing interconnection among "factory objects" anywhere and at any time. In this context, people and machines have to safely cooperate and a high level of protection needs to be guaranteed for both operators and the surrounding environment. For this reason, safety systems, aiming at decreasing risks and failure probabilities, are nowadays of uttermost importance. Several Functional Safety communication protocols have been developed during these years pointing to increase data integrity and guarantee protection in a safety system. Popular examples are Fail Safe over EtherCAT (FSoE), ProfiSAFE, and OPC-UA Safety. These protocols, although conceived for wired networks, can be in principle adopted also by wireless communication, as they are developed by using a black channel approach. Nevertheless, the implementation of these protocols over different wireless networks is challenging as they might not ensure the required Safety Integrated Level (SIL). This paper, moving from the aforementioned observations and the need for wireless solutions in the IIoT context, focuses on proposing a possible implementation of FSOE over Wi-Fi, running UDP at the transport layer. In particular, by using suitable experimental outcomes, an OMNeT++ simulator has been calibrated, thus enabling the possibility to analyze the proposed protocol in wide industrial systems.**

***Index Terms*—Factory Automation, Wi-Fi, FSOE, Industry 4.0, Safety, Wireless, OMnet++**

I. Introduction

The Internet of Things (IoT) is spreading not only in the consumer field, but also in industrial automation, process control and distributed measurement systems. As a matter of fact, several research activities are currently in progress to extend the IoT paradigm also in the aforementioned new fields of application, thus exploiting the so called Industrial Internet of Things (IIoT) [1]. In this context, sensors/actuators, controllers and any other object within an automation system are smartly, and possibly wireless, interconnected to each other, becoming part of wide and integrated factory networks and measurement systems, where operators cooperate with machinery becoming part of the production process itself.

Analogously to what happened in the consumer IoT scenario [2], the IIoT is increasingly introducing cybersafety and cybersecurity issues into the industrial environment. A large number of new vulnerabilities and issues are introduced, that can undermine the safety of industrial plants [3]. In this scenario, functional safety systems are acquiring ever greater importance, such that new protocol proposals [4] have recently been discussed. Even more, the growing use of distributed industrial equipment and mobile robots, which often make use of wireless communications, requires that even these communication systems guarantee the same degree of safety as their wired counterparts.

Moving from wired to wireless safe communication is hence becoming a hot and challenging research topic in the industrial context, as well as in other safety critical fields, such as automotive [5] and avionics [6]. Wireless safety solutions can be derived, for example, by exploiting the ones designed with the black channel approach described in IEC 61784-3. However, industrial systems are far away from stable implementations of wireless safety networks since, at present, desired SIL levels can not be achieved by using wireless solutions. In this work, we address the widespread Fail Safe over EtherCat (FSoE) [7] functional safety protocol. In particular, moving from a prototype implementation of the FSoE protocol over a Wi-Fi network [8], we developed a realistic simulation model based on OMNet++. The model has been carefully calibrated thanks to the results obtained from the prototype so that it can be used to simulate more complex configurations. In this respect, an example is reported where a WiFi-based FSoE network comprising a master and five mobile/fixed slaves is simulated.

In detail, the paper is organized as follows. Section II gives a brief background about the FSoE protocol. In Section III both the simulator implementations and the calibration procedure are presented. In Section IV, the calibrated model is

used to simulate a typical functional safety wireless network. Finally, section V concludes the paper, pointing out some future directions of research.

II. Theoretical Foundations

A. FailSafe Over EtherCat

FailSafe over EtherCat (FSoE), is a safety communication protocol conceived to work in conjunction with EtherCat. It is referred to as Functional Safety Communication Profile 12–1 by IEC 61784–3. All the protocols defined by IEC 61784–3 are designed with the black channel approach. This means that the safety communication protocol is not aware of the transmission medium nor of the protocols used to encapsulate the safety data. Indeed, the protocol itself encompasses all the countermeasures to detect possible communication errors. In particular, the errors that FSoE can detect and the correspondent countermeasures are briefly reported in Table I.

Table I
FSOE Safety countermeasures

Fault	Safety Measures			
	Seq. Nr	Watchdog	Con.Id	CRC
Corruption				X
Repetition	X			
Incor.Seq.	X			
Loss	X			
Delay		X		
Insertion	X	X	X	
Masquerade		X	X	X
Addressing		X	X	

As can be seen, these countermeasures are

- Packet Sequencing. Each FSoE node implements an incremental counter, referred to as Virtual Sequence Number, that allows checking the correct sequence of the exchanged Safety Protocol Data Unit (SPDU). It is defined as virtual since it is not directly included in the SPDU. Nonetheless, it is used in the calculation of the CRC.
- Time Expectation. All the devices use watchdog timers to check whether SPDUs violated time expectations due to delays on the network. The actual value of the watchdog depends on the network cycle time and the intrinsic dynamic of the process to control.
- Connection Authentication. Each device has a unique Connection Id. This ensures that data exchange takes place exclusively among identified partners.
- Data Integrity Assurance. This countermeasure is implemented by the CRC which ensures the integrity SPDUs. It is designed in such a way that subsequent SPDUs, even containing the same data, are completely distinguishable.

FSoE is a Master–Slave protocol, with a unique FSoE master, and several FSoE slaves. During normal operation, the FSoE master cyclically polls all the FSoE slaves. The data exchange takes place over FSoE connections, which are virtual communication channels established during the initialization phase. The FSoE Safety Protocol Data Unit (SPDU), which corresponds to the safety message, has two possible formats depending on the amount of safe data bytes that have to be exchanged. The simplest format is used to transfer a single byte of safety data from master to slave and vice versa and it is presented in Fig. 1.

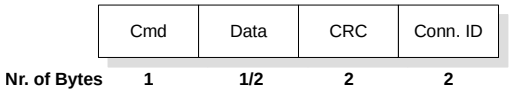

Figure 1. Basic FSoE frame

The first byte which is referred to as *Command*, contains the specific state of the FSoE connection, allowing to determine the meaning of the safety data. This is followed by the *Data* field which carries at most 2 bytes of safety information, by the 16-bit (2 bytes) *Cyclic Redundancy Check* (CRC) and then by the *Connection Id* (2 bytes) which is a unique identifier of the current specific FSoE connection. To ensure an adequate level of safety, when one or more of the countermeasures detects an error, the FSoE internal machine state is reset, i.e. the connection is forced to a safe state. In agreement with the required Safety Integrated Level, FSoE defines the maximum number of the machine-state reset. With an adequate robust channel, it is possible to guarantee the transmission with up to a SIL 3.

B. Safety over Wi–Fi

All protocols described by IEC61784-3 are implemented according to the black channel approach. In this approach, the safety nodes are neither aware of the characteristics of the transmission medium nor of the transport protocol used; therefore, the safety protocol must contain all the necessary countermeasures to protect the SPDU from possible communication errors introduced by the transmission medium. According to this principle, the safety communication layer is placed above the application layer. All layers below the safety one, are considered black channels and therefore are not encompassed in the considerations regarding safety. An example of the application of this principle is shown in Fig. 2.

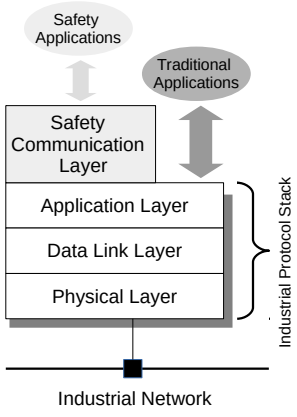

Figure 2. Protocol Stack of a Functional Safety Node

In recent years, research has been conducted to bring existing safety protocols designed for wired networks to wireless

networks. Indeed this is a mandatory step to open up to a completely new category of industrial equipment used in manufacturing that will make use of IIoT and require strict functional safety features, as Collaborative Robots (Cobots) and Autonomous Guided Vehicles (AGVs) [9]. In this perspective, new emerging technologies such as Time Sensitive Networking (TSN) and OPC-UA safety have been proposed in [10] to design self-organizing safety systems able to automatically generate a suitable safety configuration based on the working conditions. However, despite the increasing adoption of the new technologies proposed by Industry 4.0, several plants still use the traditional, perhaps legacy, fieldbus protocols, and their safety extensions.

In fact, in [11] the authors presented a proof-of-concept for the use of WirelessHART in safety-critical communications based on ProfiSafe. They successfully exploited the black channel to integrate into the ProfiSafe network of wireless gateways to extend the wired network. Essentially, they only changed the transmission system without changing anything in the upper layers of the protocol stack. In [8] we proposed a prototype FSoE implementation over WiFi. In practice, we developed an application in which safety PDUs are encapsulated in UDP messages that were delivered via IEEE802.11. We kept only the safety layer and completely replaced the underlying transport protocols for which FSoE was intended for. The main issue that emerged from this work was the relatively high packet loss that caused the FSoE connection to be reset more frequently than allowed by the standard. Nonetheless, the achieved performance was interesting, suggesting that the implementation could be suitable for certain applications.

III. OMNet++ Simulation Model

The aim of this study is the implementation of an accurate simulation model, able to reproduce realistic representations of the industrial wireless environment behavior, to provide with a tool for the performance assessment of wide wireless industrial safety networks.

To this purpose, the FSoE over Wi-Fi protocol has been implemented using the widespread, discrete event OMNet++ simulator. In particular, OMNet++ is widely adopted to simulate communication networks and to model the surrounding electromagnetic environment, allowing to properly build protocols exploiting existing modules. In our case, this feature reveals particularly useful since we implemented FSoE using UDP and the validated WiFi stack made available by OMNet++ [12].

However, these models often implement generic calibrations that can simulate a wide range of scenarios but are unlikely to reproduce specific use cases. Therefore, to be able to carefully simulate a WiFi-based FSoE network, it becomes imperative to set up a precise calibration phase of both the channel errors models and the polling time, which is representative of the time necessary to complete the communication cycle between two devices.

A. Calibration of the channel error model

To the aim of the IEEE802.11g OFDM error model calibration, we have at first determined the Packet Error Rate–Signal Noise Ratio (PER-SNR) relationship through experimental measurements on the channel. We then exploited a feature of the OMNet++ framework, which allows to feed the simulator with suitable lookup tables representing the PER-SNR relationship. In this way, we directly employed data from the field to reproduce a very accurate calibration of the channel model within the simulator.

To this extent, we have reproduced the experimental setup proposed in [13] to obtain measurements of the PER-SNR function. With an approach similar to [14], we then carried out a fine tuning of the main parameters of the OMNet++ Wi-Fi channel model. For repeatability purposes, in the simulation and experimental setups several tests for each specific SNR value have been executed, and the corresponding PER evaluated. The results are shown in Fig. 3 where the PER-SNR curves obtained an OMNet++ are compared with the experimental ones. Table II shows the comparison of the PERs. As can be seen, the values are rather similar, thus demonstrating the quality of the calibration of the error model.

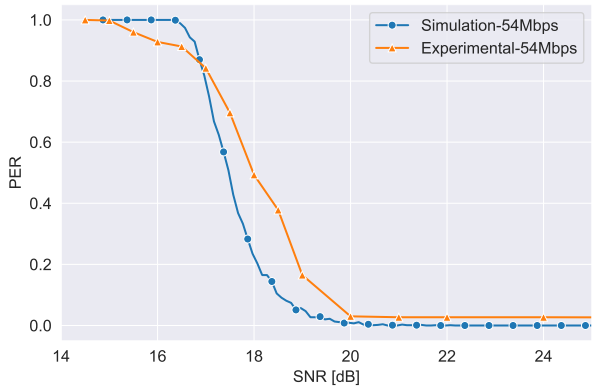

Figure 3. Experimental and simulated PER–SNR curves after calibration.

B. Calibration of the polling time

We subsequently focused on the FSoE prototype implementation, specifically on the evaluation of the Polling Time (t_P) between a FSoE Master and the FSoE Slave. t_P is defined, in this context, as the time elapsed from the generation of a FSoE frame transmission request by the master and the reception of the confirm primitive from the slave. As pointed out in [8], it includes the time necessary to execute both the FSoE and the underlying protocol stacks in both master and slave, and the time to transmit the safety frame and the slave's answer message. The latter time, assuming no collision during the transmission [13], may be considered deterministic. Conversely, the former time usually depend to the characteristic of the device where protocols and applications are implemented, such as the computational capabilities, as well as operating system calls, memory management, etc., which may introduce

random latencies and jitter. OMNet++ in some way tries to take into account these uncertainties by introducing some uniform random delays, but obviously they cannot correspond to real use cases.

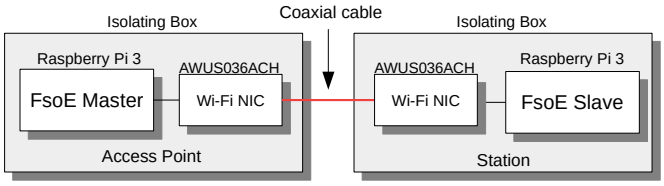

Figure 4. Experimental set-up

Therefore, a further set of tests were relevant to the calibration of the simulator with respect to experimental values. To this goal, a suitable measurement setup has been designed, as shown in Fig. 4.

All the experiments have been carried out on Raspberry Pi Model 3B boards which run a general-purpose operating system Raspbian OS (Kernel version 5.4.83). Each device have been equipped with an external Alfa AWUS036ACH USB Wireless Network Interface Controller (WNIC) set to operate in the 2.4 GHz band with IEEE802.11g modulation standard and output power of 30 dBm. Moreover, the rate adaptation features have been disabled, thus using a fixed 54 MBps rate. Tests have been conducted in an industrial area, whereby it is necessary to minimize as far as possible the influence of external factors, for example, other WiFi stations or background noise. For this reason, the WNICs antennas have been connected via a coaxial cable, thus simulating an ideal transmission medium. To further increase the robustness to external noise, the Raspberry Pis and the WNICs have been embedded into separate shielding boxes, and finally, a variable RF attenuator, with an attenuation range 50-110 dBm, has been inserted in the coaxial transmission line to properly control the attenuation of the real transmission medium.

The FSoE Master and FSoE Slave have been implemented on two different Raspberry Pi boards, respectively configured as Wi-Fi Access Point (AP) and Station (STA). The communication between the Master and the Slave is hence managed without intermediate devices. The simplest FSoE SPDU (7 bytes) has been encapsulated into a UDP frame, to carry safety data. It is worth noting that the FSoE stack runs on the Raspberry Pi as normal non-prioritized processes. Moreover, SPDUs are sent continuously in a non-scheduled way. In particular, after the reception of the safety frame SF_i from the master, the slave answers with the frame AF_i. Then, the master sends a new safety frame SF_{i+1} immediately after the reception of the answer AF_i from the slave. In each device, a watchdog of 250 ms monitors the FSoE connection cycle to detect possible delays on the network. If a device does not receive any answer from the communication partner within the specified timeout, the frame is marked as lost and the FSoE connection is re-initialized.

Several experimental sessions have been conducted, to test the medium with different attenuation values, that have been varied in 1dB steps. Correspondingly, for each measurement session, the acquisition of more than 50000 unique values of the polling time t_P has been acquired.

For the calibration of the polling time model, we used the probability densities obtained from the experimental measurements. Each of these was obtained for different channel attenuation values. Using the Inverse Transform Sampling method, the delays to be used in the transmission are sampled from the experimental densities according to the received signal strength. In practice, when the Master issues the transmission of an SPDU, the simulator calculates what will be the channel attenuation and therefore the power of the received signal. Based on this, the simulator chooses the density from which to sample, and schedules the response message from the slave after the delay generated by the sampling. The outcome of the polling time calibration is shown in Fig. 5 while a more detailed statistic is reported in Table II. As can be seen in Fig. 5 the trend of the mean, minimum and maximum values of the polling time is rather similar for both the experimental and simulated sessions.

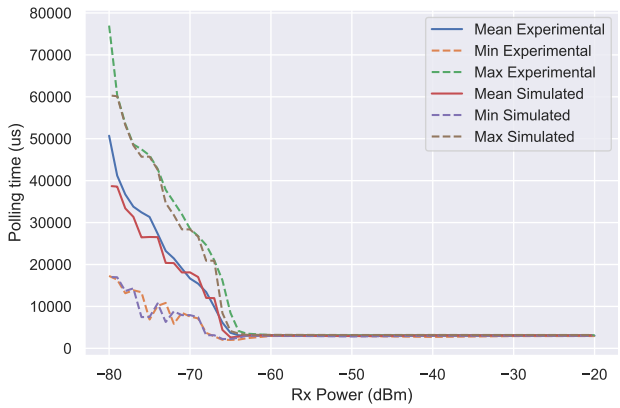

Figure 5. Comparison of mean, maximum and minimum polling time on the experimental and simulated setup

Indeed, for high reception powers the trends are practically identical. For powers lower than -60 dBm the polling time values follow the same trend but with a definitely higher error. This observation is also confirmed by Fig. 6 which shows the Mean Square Error calculated on the average polling time.

As can be seen, the error is almost zero for relatively high receiving powers, while it tends to increase as the intensity of the received signal decreases. This phenomenon is due to the model of path loss used in the simulator which is the *LogNormalShadowing*. The path loss model not only takes care of simulating the attenuation of the signal and calculating the probability that the signal can be received, but also it adds a certain degree of variability to the signal reception time to simulate the effects of attenuation. Therefore, the uncertainties introduced by the path loss model overlap with the delays introduced by the polling time model causing it to deviate slightly from the experimental trend. The solution to this problem will require the implementation of a completely

Table II
STATISTICS OF THE POLLING TIME AND PER

rxPower (dBm)	Experimental					Simulated					MSE on mean (%)
	Polling time (µs)				PER	Polling time (µs)				PER	
	Mean	Std	Min	Max		Mean	Std	Min	Max		
-80.0	50720.27	11510.14	17271.40	76988.70	0.030	41473.87	8602.50	23940.54	59851.54	0.027	18.23
-76.0	32437.22	6756.42	13383.80	47484.70	0	31377.75	6506.41	17041.85	45395.85	0	3.26
-73.0	23225.14	6050.65	10836.40	37790.80	0	21427.64	4985.96	13059.43	34340.43	0	7.73
-70.0	16663.57	4124.12	7597.51	28525.20	0	16684.58	3954.34	11635.10	27807.10	0	0.12
-67.0	9870.57	3460.87	2855.14	21021.10	0	9877.29	3260.08	4049.86	20378.86	0	0.06
-64.0	3116.34	220.08	2076.29	4144.56	0	3106.29	82.38	3035.66	3535.66	0	0.32
-50.0	3010.49	23.01	2931.70	3139.15	0	3084.09	21.70	3014.20	3173.20	0	2.44
-20.0	3010.20	21.32	2930.30	3130.82	0	3085.52	19.16	3048.02	3170.02	0	2.50

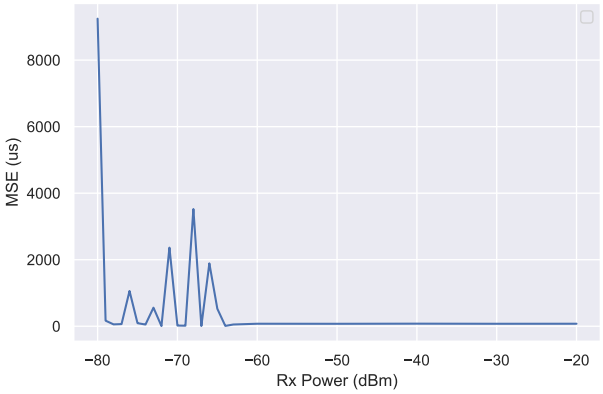

Figure 6. MSE

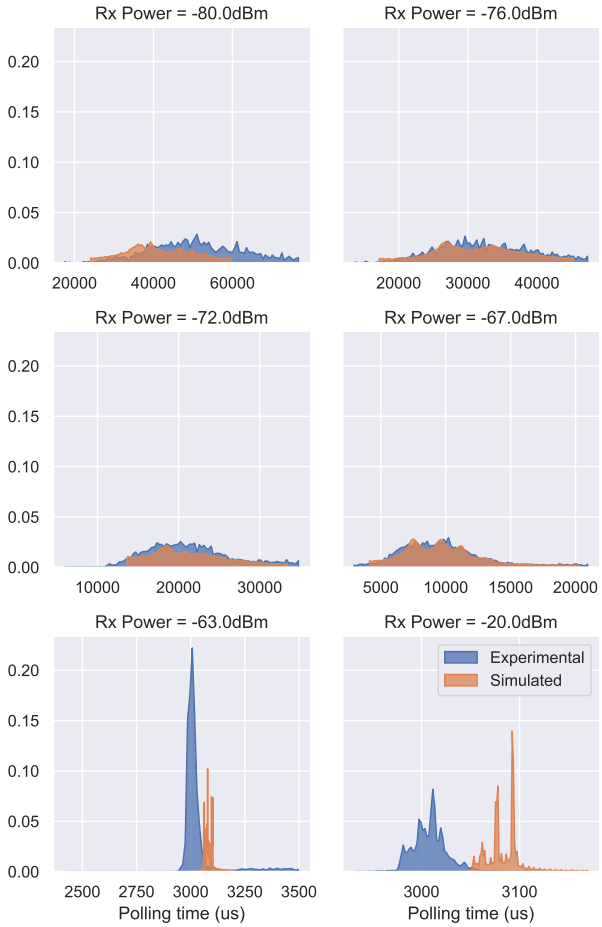

Figure 7. Comparison of the probability density function of the polling time in the experimental and simulated setup

custom path loss model, which will be left for future work. In the current state of the simulator, the path loss model has been calibrated to minimize these superimposition effects.

In general, the accuracy of the calibration can be confirmed by the MSE which, except for some isolated cases, remains less than 4%. This is also confirmed by by Fig. 7 which shows the comparison of the probability density function of the polling time. As can be seen, they are definitely very similar.

C. Simulation with multiple slaves

The simulation of a realistic multi-node network needs particular care and requires the execution of several tests. In this Section, we analyze the outcomes of some simulation sessions we have carried out towards the assessment of the proposed simulation model. We refer to the prototype network, described in Fig. 8, which is composed of one FSoE master and five FSoE slaves. The position of the nodes with respect to each other has been carefully selected to reproduce and test three different communication scenarios. Firstly, aiming at analyzing the protocol behavior in absence of mutual interference, Slaves 1 and 2 are placed relatively distant to each other, thus reproducing an ideal situation. On the contrary, slaves 3 and 4 have been placed close together, to study how possible interference can affect the polling time and the PER. Finally, slave 5 has been placed far away from the master, to analyze the impact of a great attenuation, i.e. a relatively low SNR, on the polling time and the PER.

Simulation statistics of both the polling time and exchanged packets for each node are reported in Table III.

Comparing the behavior of Slaves 1 and 2, it is possible

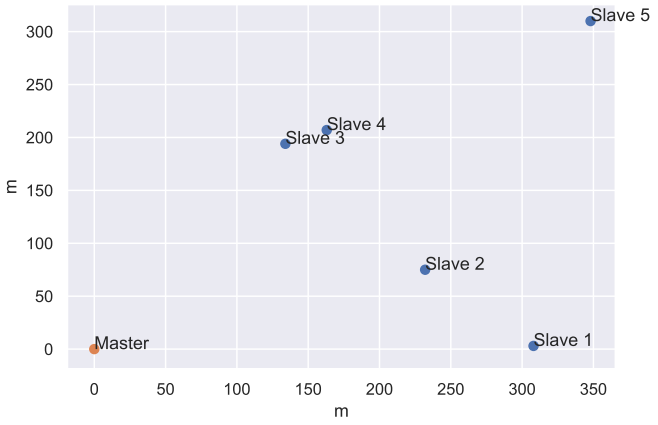

Figure 8. Positions of the nodes

Table III
STATISTICS OF THE POLLING TIME IN SIMULATION WITH MULTIPLE SLAVES

	Polling time (µs)		**Packets**			
Slave	Mean	Std	Sent	Acknowledged	Lost	PER
1	33932.20	6581.77	71997	71997	0	0.000000
2	27618.30	6540.25	71997	71997	0	0.000000
3	27390.94	6506.25	71999	71999	0	0.000000
4	31431.96	6545.19	72000	71998	2	0.000028
5	105864.38	52708.85	71998	52683	19315	0.268271

to underline that the latter introduces a slightly lower polling time, while both do not experience any packet loss. This is reasonable as they do not have other nodes nearby, but the distance between Slave 1 and the Master is higher than the Slave 2 one. Conversely, Slaves 3 and 4 introduce mutual interference, as can be noted from both the polling time (t_p) and the Packet Error Rate (PER). Indeed, the polling times of Slaves 3 and 4 are quite similar to those of Slave 1 and 2, although the latter ones have a greater distance from the master. Furthermore, Slave 4 also experiences some packet loss. Finally, Slave 5 introduces both higher t_p and PER, thus underlying the impact of the attenuation on the communication.

IV. Conclusions and Future Directions

In this paper, we addressed the design of an OMNet++ simulation model for a wireless implementation of the FailSafe over EtherCat safety communication protocol. In particular, we focused on the calibration of such simulator, tuning both the channel error and path loss models, and the polling time. To this purpose, we exploited measurement results obtained using suitable experimental setups, specifically designed for this task. The results of the assessment carried out on a multi–node network underline that in most cases the error between simulated and experimental results is very low.

While simulations carried out under different interference conditions demonstrated the goodness of the calibration, a completely custom path loss model needs to be developed to limit the effects of the superimposition of multiple delays. Hence, extensive session tests on both real networks and their simulated counterparts are necessary to come to the effective validation of the simulator. Specifically, scenarios with multiple nodes, possibly mobile, will be tested to verify that, even in case of external agents interference the simulator gives realistic results.

Finally, we will expand the simulator capabilities by implementing an Application Programming Interface (API) that can easily simulate different types of nodes (from electric drives to mobile robots) and interface with external tools such as Robotic Operating System (ROS).

References

[1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, Nov 2018.

[2] X. Yu and H. Guo, "A survey on iiot security," in *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*, 2019, pp. 1–5.

[3] S. Munirathinam, "Industry 4.0: Industrial internet of things (iiot)," in *Advances in computers*. Elsevier, 2020, vol. 117, no. 1, pp. 129–164.

[4] X. Zhou, A. Xu, B. Yan, Y. Sun, X. Han, and J. Wang, "Design and implementation of functional safety fieldbus communication protocol," in *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 5, 2021, pp. 1910–1917.

[5] G. Xie, Y. Li, Y. Han, Y. Xie, G. Zeng, and R. Li, "Recent advances and future trends for automotive functional safety design methodologies," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 5629–5642, 2020.

[6] A. Allouch, A. Koubâa, M. Khalgui, and T. Abbes, "Qualitative and quantitative risk analysis and safety assessment of unmanned aerial vehicles missions over the internet," *IEEE Access*, vol. 7, pp. 53 392–53 410, 2019.

[7] "Industrial communication networks - Profiles - Part 3-12: Functional safety fieldbuses - Additional specifications for CPF 12," International Electrotechnical Commission, Standard, 2019.

[8] A. Morato, S. Vitturi, A. Cenedese, G. Fadel, and F. Tramarin, "The Fail Safe over EtherCAT (FSoE) protocol implemented on the IEEE 802.11 WLAN," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019, pp. 1163–1170.

[9] S. H. Robinson, "Living with the challenges to functional safety in the industrial internet of things," 2019.

[10] D. Etz, T. Frühwirth, and W. Kastner, "Flexible safety systems for smart manufacturing," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2020, pp. 1123–1126.

[11] J. Åkerberg, F. Reichenbach, and M. Björkman, "Enabling safety-critical wireless communication using WirelessHART and PROFIsafe," in *2010 IEEE 15th Conference on Emerging Technologies Factory Automation (ETFA 2010)*, Sep. 2010, pp. 1–8, iSSN: 1946-0759.

[12] M. Bredel and M. Bergner, "On the accuracy of ieee 802.11g wireless lan simulations using omnet++," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, ser. Simutools '09. Brussels, BEL: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009. [Online]. Available: https://doi.org/10.4108/ICST.SIMUTOOLS2009.5585

[13] F. Tramarin, S. Vitturi, M. Luvisotto, and A. Zanella, "On the use of ieee 802.11n for industrial communications," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1877–1886, 2016.

[14] G. Peserico, T. Fedullo, A. Morato, F. Tramarin, L. Rovati, and S. Vitturi, "SNR-based Reinforcement Learning Rate Adaptation for Time Critical Wi-Fi Networks: Assessment through a Calibrated Simulator," in *2021 IEEE Instrumentation and Measurement Technology Conference*, 2021, pp. 1–6. Accepted.