*Sensors & Transducers, Vol. 265, Issue 2, May 2024, pp. 127-138*

www.sensorsportal.com

**Sensors & Transducers**

Published by IFSA Publishing, S. L., 2024
http://www.sensorsportal.com

# Adaptive Communication Protocols – Steps Towards a Grey-channel Approach

**Michael H. SCHWARZ and Josef BÖRCSÖK**
University of Kassel, Department Computer Architecture and System Programming (ICAS),
Wilhelmshöher Allee 71, 34121 Kassel, Germany
E-mail: m.schwarz@uni-kassel.de

*Received: 8 April 2024 / Accepted: 10 May 2024 / Published: 30 May 2024*

**Abstract:** Wireless communication is a key issue in many industrial and research areas. On one hand cables are not necessary anymore, and this reduces costs while it is getting much more flexible and information can be selected from any device. On the other hand, safety and especially security are becoming even more important as data can be read by everyone and can be more easily perturbed or intentionally falsified. This article defines protocols that changes its safety attributes according to the continually calculated probability of error per hour value. A grey (gray) channel is defined that is not ignoring any information from the communication channel (black-channel) and uses the observed information to adapt the safety attributes. The protocols use short data-lengths and several different cyclic redundancy checks (CRCs) which are changed depending on the current safety integrity level, the observed channel characteristics and the amount of data to be sent.

**Keywords:** Adaptive communication protocol, Probability of failure on demand, Safety integrity level, Black-channel, Grey-channel.

## 1. Introduction

On one side, wireless communication is getting popular in automation [1], process control [2], robotics [3] and many other industrial areas. Investigations on 6G are in full process [4]. On the other side manufacturers, end users and academia [5] have some doubts about safety and especially security: Wireless communication is non-deterministic and more interference-prone as non-wireless communication [6]. The interference can be caused by noise of other devices, signal reflections or disturbances as a form of intrusion [7, 8]. Safety and security were strictly separated in the past, but have to be combined in safety analysis [9], as a safety issue can result in a critical situation, but also, a security issue can cause safety problems. However, new methods arise to detect intruders, which can be identified with its radio frequency fingerprints [10, 11]. If the fingerprints are known, then data can be trusted, otherwise, if the fingerprints are unknown the data is rejected or marked as suspicious. Fig. 1 shows some scenarios that might occur [12]. Scenario 1 shows a transmission without any problems. Scenario 2 shows a situation where the receiver is blocked. This is a situation that is not possible using wired communication. Scenario 3 shows some interferences. This scenario can also happen in wired communication, if the cable is badly shielded or a (fieldbus)-devise picks up some noise. Scenario 4 is also new for wired communication, as a device tries to "sneak into" a communication.

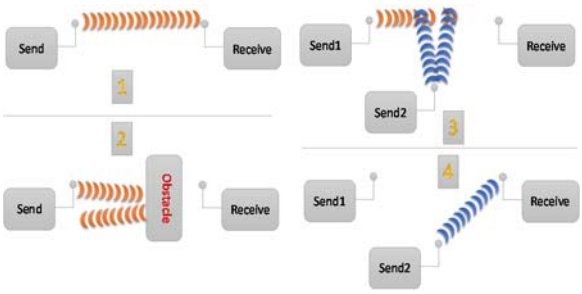


**Fig. 1.** Various communication scenarios.

In functional safety [13], the *Probability of Failure per Hour (PFH)* is the value which defines in which *Safety Integrity Level* (SIL) the system is operating as shown in Table 1.

**Table 1.** Relation of PFH and SIL [13].

| SIL | SIL/PFH | |
|---|---|---|
| | ***PFH of the safety function*** | ***PFH-of safety communication channel*** |
| 4 | $< 10^{-8}$ | $< 10^{-10}$ |
| 3 | $< 10^{-7}$ | $< 10^{-9}$ |
| 2 | $< 10^{-6}$ | $< 10^{-8}$ |
| 1 | $< 10^{-5}$ | $< 10^{-7}$ |

This value states how large the possibility is that an error is not detected and this leads to a safety-critical situation. Wired communication is allowed to contribute up to 1% of the overall SIL of the entire system [13]. Enough methods exist to countervail increasing errors [14]. In wireless communication a temporal increase can occur and Fig 2. shows the behavior of the PFH value of the communications using different CRCs. The first CRC (blue) is No.5 with a Hamming distance of 6, while the second CRC (red) is No.0 with a Hamming distance of 3, both listed in Table 5, respectively. This demonstrates that the PFH in communication should not be seen as a fixed number but as a variable that can change its value. Therefore, it is important to monitor continuously the PFH value, as an increase of the errors of the communication channel can lead to an increase of the overall PFH value which can finally lead to a different overall SIL, that means the system is operating in a safety level that is not allowed.

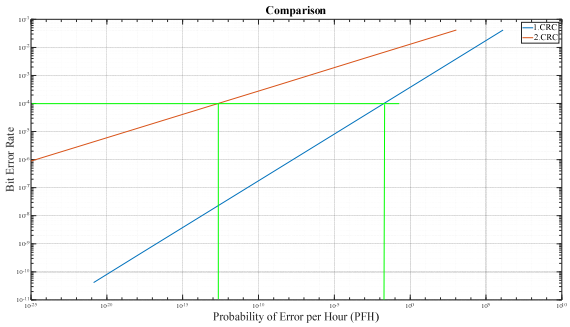


**Fig. 2.** PFH behavior for two different CRCs.

The remaining article is organized as follows: Section 2 describes different channel models in function safety. Section 3 summarizes error detection methods used in industrial protocols. Section 4 discusses the mathematical background, while Section 5 defines the structure of the protocols. Afterwards in Section 6, data errors are discussed. The decision maker is presented in Section 7, followed by short results in Section 8. Conclusions and future work are described in the final section 9 followed by the acknowledgment.

## 2. Different Channels

In communication different channel approaches exists, which are the black-channel, white-channel and the grey-channel, sometimes as stated as gray-channel.

In the safety communication standard IEC 61784 part 3 [15] a white-channel is defined as follows:

"*3.1.1.44 white channel: communication channel in which all relevant hardware and software components are designed, implemented and validated according to IEC 61508.*"

This means, all parts of the communication have to go through all the same steps of certification as the remaining system. This also means that for industrial systems such as Programmable Logic Controllers (PLC) different devices have to exist for safe and non-safe communication, and also the question arises what restrictions or conditions has the actual communication channel to fulfil. An interesting alternative for the white-channel approach is the black-channel [15]:

*"3.1.1.3 black channel: communication channel without available evidence of design or validation according to IEC 61508."*

Fig. 3 [15] illustrates the black-channel approach, where all safety requirements are performed in the safety layer, everything which is below the safety layer is defined as the black channel.

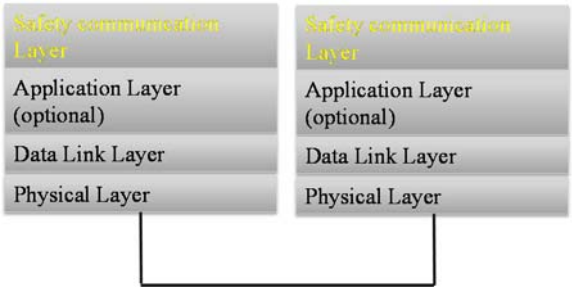


**Fig. 3.** Black channel communication.

The safety layer has to detect errors in data using additional detection methods which are discussed in the next section.

Also, the standard communication can be used for a safety communication and a safety protocol is embedded in the standard protocol [12]. This is often used in automation industries, where safe and non-safety data can be sent in one data frame. In an information part it is indicated if the data belongs to the safety layer or to the standard layer. Fig. 4 sketches the idea of using a combination of safe and non-safe communication using the same transmission lines. Fig. 5 [15] presents a more detailed information, how the communication can be used with both safe and non-safe communication and how the safe and non-safe application is strictly separated from each other, so that they do not interfere with each other.

The term grey channel appears in relation to railway systems and DIN EN 50159 [16, 17], but has

a similar meaning as the black channel in the standard IEC 61784-3 [15].

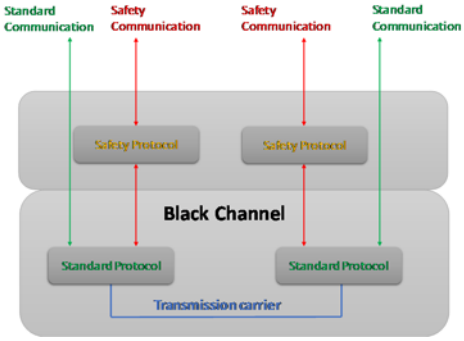


**Fig. 4.** Combining safety and standard protocols.

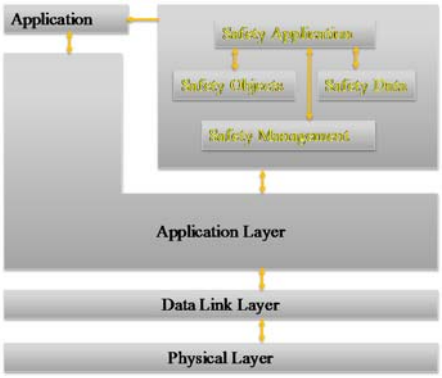


**Fig. 5.** Combining safety and non-safe application [15].

In the work of Åkerberg [18] a grey channel appears as a non-safe communication channel without a black channel assessment. Here, safe related data are transmitted over a grey channel for only information purposes.

In Åkerberg et al. [19] an interesting definition of a grey channel is presented, where a black channel has not to fulfil any requirements. A grey channel is therefore a communication that as to fulfil requirements or has to possess some constraints, which are minimum bandwidth or maximum amount of latency. The authors argue that nowadays a grey channel would be a better approach for safety communication.

In the present article a grey channel should be observable is such a way that errors and data loss can be detected and/or measured. A combination with measuring the transition speed, or measuring the deviation of the expected arrival time of a message to determine if a creeping latency occurs, can be beneficial and useful.

## 3. Error Detection Methods

To detect errors in messages several communication standards [15, 16] and reports on protocols for example for ProfiSafe [20-23] and OpenSafety [24-26] provide a lot of information about error detection mechanism. All messages are falsified due to some errors, faults or unintentional interference. The following possible errors have to be detected:

- *Unintended repetition* (1): the received message is outdated [15];
- *Corruption* (2): the message is falsified [15];
- *Incorrect sequence* (3): the defined sequence of the message is wrong [15];
- *Loss* (4): the sent message is not received or acknowledged [15];
- *Unacceptable delay* (5): the message is received after the allowed time [15];
- *Insertion* (6): a message is received belonging to an unexpected or unknown source [15];
- *Masquerade* (7): a non-safe relevant message is interpreted as a safety message [15].

The methods to identify errors in messages are the following:

- *Sequence number* (A): it is an integrated number that changes from message to message in a predefined way [15];
- *Time stamp* (B): the time or time and data is added to the message. It can be a relative or absolute time. It needs synchronization and it has to be monitored [15];
- *Time expectation* (C): the time between two received messages (emergency stop) or between sent and received message [15];
- *Connection authentication* (D): it is an exclusive source and/or destination identifier that describes the logical address of the safety relevant transmitter [15];
- *Feedback message (Acknowledgement)* (E): the receiver confirms the correct arrival of the message [15];
- *Data integrity assurance* (F): additional data is included in the message to detect corrupted data [15];
- *Redundancy with cross checking* (G): The message is sent twice, within one message or two separate ones. Identical or different integrity methods can be used [15].

**Table 2.** Relation Errors and detection methods.

| Method/Error | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| **1** | X | X | | | | | X |
| **2** | | | | | | X | X |
| **3** | X | X | | | | | X |
| **4** | X | X | | | X | | X |
| **5** | | X | X | | | | |
| **6** | X | X | | X | X | | X |
| **7** | | | | X | X | | |

Table 2 shows the different error types, identified by the numbers and the detection methods identified by the capital letters, as stated in the description above. As it can be seen, there is not one method that fulfil all criteria to detect the errors, so it will be a combination of them.

## 4. PFH Calculation

This section calculates the relevant equations, a detailed derivation can be found in [14, 27, 28].

### 4.1. General Erasure Channel

The channel model of the digital transmission can be described by the *Generalised Erasure Channel* (GEC) [14, 27, 31]. This model considers of two elements for inputs {0,1} and three elements for the output {0, 1, e}, where the input is the sending device and the output is the received value by the receiver. The element *e* indicates that an input whether 0 or 1 is vanished or erased [27, 14].

$$P(e|0) = P(e|1) = \zeta, \quad (1)$$

$$P(0|1) = P(1|0) = \eta, \quad (2)$$

$$P(0|0) = P(1|1) = \theta \quad (3)$$

From this definition the Bit Error Rate (BER) ε and the Bit Loss Rate (BLR) φ can be explained. BER is the number of falsified bits per second and the BLR is the erased bits per second.

### 4.2. Relation of Probability of Undetected Error and PFH

The probability of undetected errors $P_{ue}$ for transmissions protected by a linear Code $C$ can be stated as follows [14, 27]:

$$P_{ue}(\zeta, \eta, \theta, C) = \sum_{l=1}^{n} A_l \cdot \eta^l \cdot \theta^{n-l}, \quad (4)$$

where $A_l$ is a number of code words of weight *l*. $\zeta$, $\eta$ and $\theta$ are already defined and finally, *n* refers to the block-length. Then, the transitions probabilities $\zeta, \eta$ and $\theta$ can be given as detailed in [14, 27, 31]:

$$\zeta = \varphi, \quad (5)$$

$$\eta = \varepsilon \cdot (1 - \varphi), \quad (6)$$

$$\theta = (1 - \varepsilon) \cdot (1 - \varphi) \quad (7)$$

Using these equations with (4) results in:

$$P_{ue}(\varepsilon, \varphi, C) = (1 - \varphi)^n \cdot P_{ue}(\varepsilon, C) \quad (8)$$

After some calculations as presented in [6, 14, 27, 29], this results finally in:

$$PFH = 36 \cdot 10^4 \cdot v \cdot (1 - \varphi)^n \cdot P_{ue}(\varepsilon, C) \quad (9)$$

For the mathematical derivation of the SIL estimations it is referred to [6, 14, 27, 29]; only the relevant equation will be shortly introduced. The estimation of BER can be given by:

$$\gamma_b = \frac{E_b}{N_0}, \quad (10)$$

where $\gamma_b$ is the signal-to-noise ratio per bit $E_b$ is the mean energy per bit and $N_0$ is the noise spectral density. The estimation of BLR starts with [27]:

$$\frac{C_{W,\gamma_b}}{W} = \log_2\left(1 + \frac{C_{W,\gamma_b}}{W}\,\gamma_b\right), \quad (11)$$

where $C_{w,\gamma_b}$ defines the channel capacity and $W$ desribes the bandwidth. The calculations finally result in [6,27,28]:

$$2^{a^x} = 2 \cdot a^{x \cdot \gamma_b}, \quad (12)$$

with the substitutions of:

$$x = \frac{(1-\varphi)}{W}, \quad (13)$$

and

$$a = 2 \cdot \varepsilon^{\varepsilon} \cdot (1 - \varepsilon)^{(1-\varepsilon)} \quad (14)$$

The probability of undetected error $P_{ue}$ can be estimated as an upper bound with a worst-case approach. The inequality of upper bound for $P_{ue}(\varepsilon, C)$ was derived by [28] for all $0 \leq \varepsilon \leq$ ½:

$$P_{ue}(\varepsilon, C) \leq \frac{72}{121} \cdot \frac{\sqrt{2\pi n}}{2^r \cdot d!} \cdot n^d \varepsilon^d + 2^n \left(\sqrt{\varepsilon}\right)^j, \quad (15)$$

with $j = n$ if $n \geq 3$ and even, and $j = n - 1$ if $n \geq 4$ and odd, where $d = d_{min}$ is the minimum distance of the code *C*, $\varepsilon$ is the value of BER and $n$ is the length of the message while $r$ defines the length of the checksum. The final argument for $P_{ue}$ gets:

$$P_{ue}^* = \frac{72}{121} \cdot \frac{\sqrt{2\pi n}}{2^r \cdot d!} \cdot n^d \varepsilon^d + \left(2\sqrt{\varepsilon}\right)^n \quad (16)$$

A detailed derivation can be found in [6, 14, 27, 29]. When combining (16) and (9), the this results in:

$$PFH = f(v, n, r, d) = 36 \cdot 10^4 \cdot v \cdot \cdot (1 - \varphi)^n \cdot \left(\frac{72}{121} \cdot \frac{\sqrt{2\pi n}}{2^r \cdot d!} \cdot n^d \varepsilon^d + \left(2\sqrt{\varepsilon}\right)^n\right) \quad (17)$$

The function consists of 6 variables: $v$, $k$, $r$, $d$, $\varphi$ and $\varepsilon$. BLR $\varphi$ and the BER ε are parameters that are measured and cannot be directly influenced. The speed of transition $v$ can be changed depending on the system under control. The remaining parameters, are those that can be really influenced. The Hamming distance $d$, can be changed when changing the CRC from one transition to the next. This has to be indicated in one of the flags that the CRC is changed. The parameter $r$ changes when length of the CRC deviates

due to a change of the Hamming distance $d$ when BER or BLR increases for example. The payload $k$ is the last parameter that will be adjusted depending on BLR $\varphi$ and BER ε. With an increase in noise the messages will be shorter, when the noise decreases the messages can be longer [6,29,30]. The message length can be split into the data length k and CRC length r:

$$n(message_{length}) = k(data_{length}) + \\ +r(CRC_{length}) \quad (18)$$

In the initial setup the Bit Loss Rate φ is set to zero. The BLR is interpreted as a communication problem with the device and not a problem of the communication channel. As the new method does not use for the message length n as a multiple of 8bit, but the smallest possible number, it cannot distinguish between a loss and a bit error. The entire message is wrong, and this is counted towards the numbers of errors.

$$PFH = 36 \cdot 10^4 \cdot v \cdot \\ \cdot \left(\frac{72}{121}\frac{\sqrt{2\pi n}}{2^r d} \cdot n^d \varepsilon^d + 2^n \cdot \left(\sqrt{\varepsilon}\right)^n\right) \quad (19)$$

## 5. Different New Protocols

This section describes two protocol structures, which are called FIDES1 (Finding Errors) and FIDES2.

### 5.1. General Structure

While in wired communication, large data packets can be transmitted [23], as for example in a simple summation frame of a field-bus system. All data for different field-bus nodes are chained up and send along the line. Fig. 6 shows a schematic, where separate slots are used for specific nodes or processes.


**Fig. 6.** Similar to Sensor-Actuator Field bus [23].

The figure presents the idea not a particular summation frame system. If an error occurs then the entire message has to be re-sent. Often each slot is separately secured with an individual CRC.

Fig. 7 shows another way where in an information or function field is stated which process or register should receive the new data. If the data gets corrupted, then for example in Modus a timeout is provoked as the CRC is an error detection but not an error correction method and it cannot be detected where or how many errors are within the data or if the CRC itself is falsified [12].

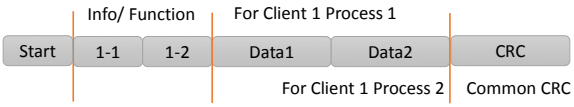


**Fig. 7.** Modbus.

In the new method, the data transition is split into small data packets., even if the data is for the same client but for a different process as shown in Fig. 8 [12].


**Fig. 8.** Wireless communication.

This means, if one data packet for a particular process is corrupted, this does not influence all the others, and only this individual data packet has to be resubmitted. The structure of the two protocol-structures is as follows, as shown in Fig. 9 [12]:

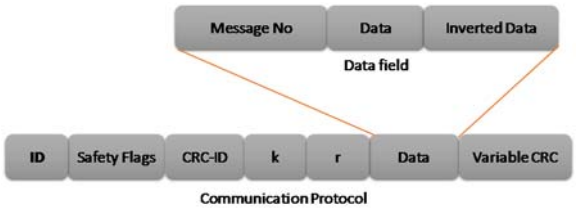


**Fig. 9.** Protocol-Structure.

The interpretation of the bits is presented in Table 2. Up to 16 different client or processes can be used. The safety flags differ from FIDES1 to FIDES2 and are described afterwards in the subsections. CRC-ID describes which CRC is selected. *K* differs depending of the protocol structure. With FIDES1 up to 1013bits are possible, therefore 10bit. With the CRCs of FIDES2 only up to 100bits can be transferred as data, therefore only 7 bits are necessary. Data varies depending of the information that is sent and the CRC length changes depending on the particular selected CRC [12].

**Table 3.** Relation Errors and detection methods.

| | Bitlength | Comments |
|---|---|---|
| ID | 4 | Client/Process identifier |
| Safety Flags | 4 | Message number |
| CRC-ID | 4 | Selected CRC |
| k | 10/7 | Data length |
| r | 4 | CRC length |
| Data | variable | Data + add. information |
| CRC | variable | CRC protection |

## 5.2. Fides1

FIDES1 was the first developed structure and the meaning of the safety flags are shown in the Table 4. Different safety features can be separately activated or deactivated. This frame uses only redundancy of the data section not an entire redundancy of the message [12].

**Table 4.** Safety Flag Bit Field for FIDES1.

| *Safety Flag* | ***Bit interpretation*** | |
|---|---|---|
| | Bit length | Comments |
| 1 | 1 | Time stamp |
| 2 | 1 | Consecutive number |
| 3 | 1 | CRC activated |
| 4 | 1 | Redundant, inverted data |

The selected CRCs are presented in Table 5. The CRCs are used from Koopman [32]. The CRCs varies from Hamming distance 3 to 6 and from data length from 57 up to 1013 bits [12].

**Table 5.** CRC Group 1.

| **CRCs Group 1 short messages** | | | | |
|---|---|---|---|---|
| ***No.*** | ***CRC*** | ***r*** | ***k*** | ***d*** |
| 0 | 0x33 | 6 | 57 | 3 |
| 1 | 0x65 | 7 | 120 | 3 |
| 2 | 0xe7 | 8 | 247 | 3 |
| 3 | 0x119 | 9 | 502 | 3 |
| 4 | 0x327 | 10 | 1013 | 3 |
| 5 | 0x5b | 7 | 56 | 4 |
| 6 | 0x83 | 8 | 119 | 4 |
| 7 | 0x17d | 9 | 246 | 4 |
| 8 | 0x247 | 10 | 501 | 4 |
| 9 | 0x583 | 11 | 1012 | 4 |
| 10 | 0xbae | 12 | 53 | 5 |
| 11 | 0x212d | 14 | 113 | 5 |
| 12 | 0xac9a | 16 | 241 | 5 |
| 13 | 0x372b | 14 | 57 | 6 |
| 14 | 0x573a | 15 | 114 | 6 |
| 15 | 0x9eb2 | 16 | 135 | 6 |

Fig. 10 shows the raw data-skeleton without any protection method, and is the minimal set to be transmitted [12].

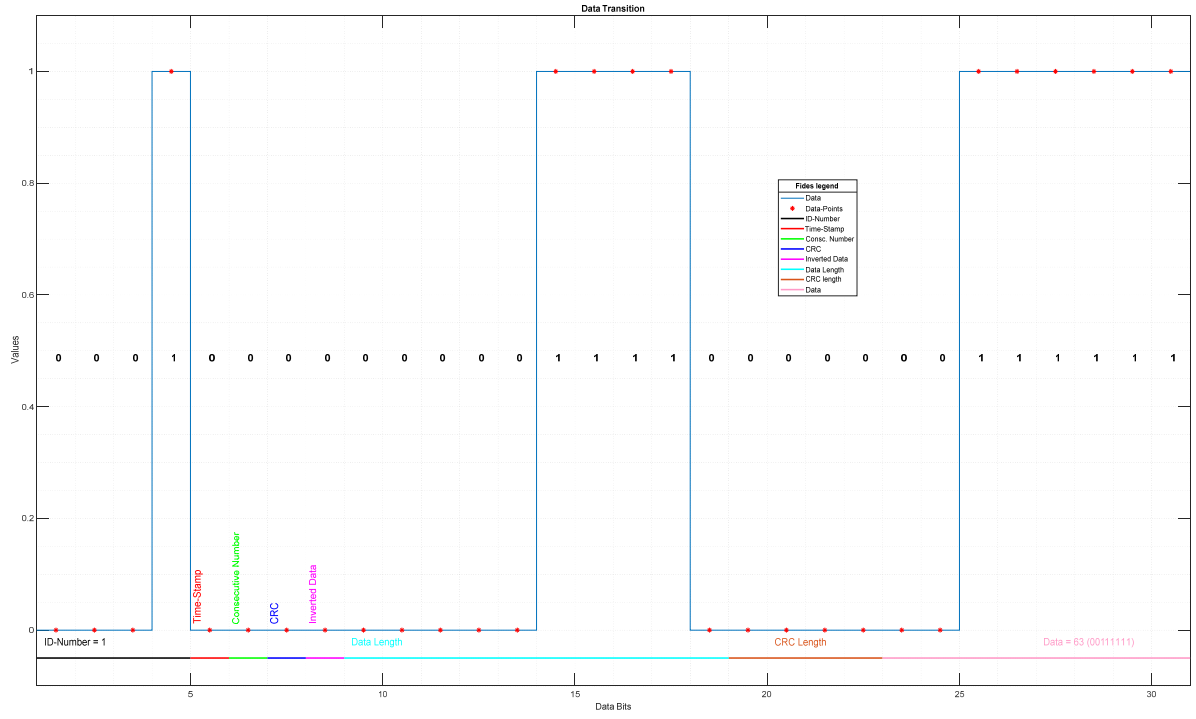


**Fig. 10.** Minimal structure.

Originally, the consecutive number was put after the safety-flags and before the parameter CRC-ID as shown in Fig. 11. This position was not optimal as the position of CRC-ID, k and r changed depending if the consecutive number feature was activated or not. Therefore, this was moved to the data section. This can be seen in the next Fig. 12, where three transmissions with consecutive number activated are presented. The time stamp is an often-used method to identify if a message is sent twice or is outdated. To synchronize several clients is possible, but an easier method is to request an answer from the client within a defined time. If this is done continuously, then it can be recognized that a client is out of order. In Fig. 13 the mechanism is demonstrated, therefore, the feature time stamp is neglected in the new framework [12].

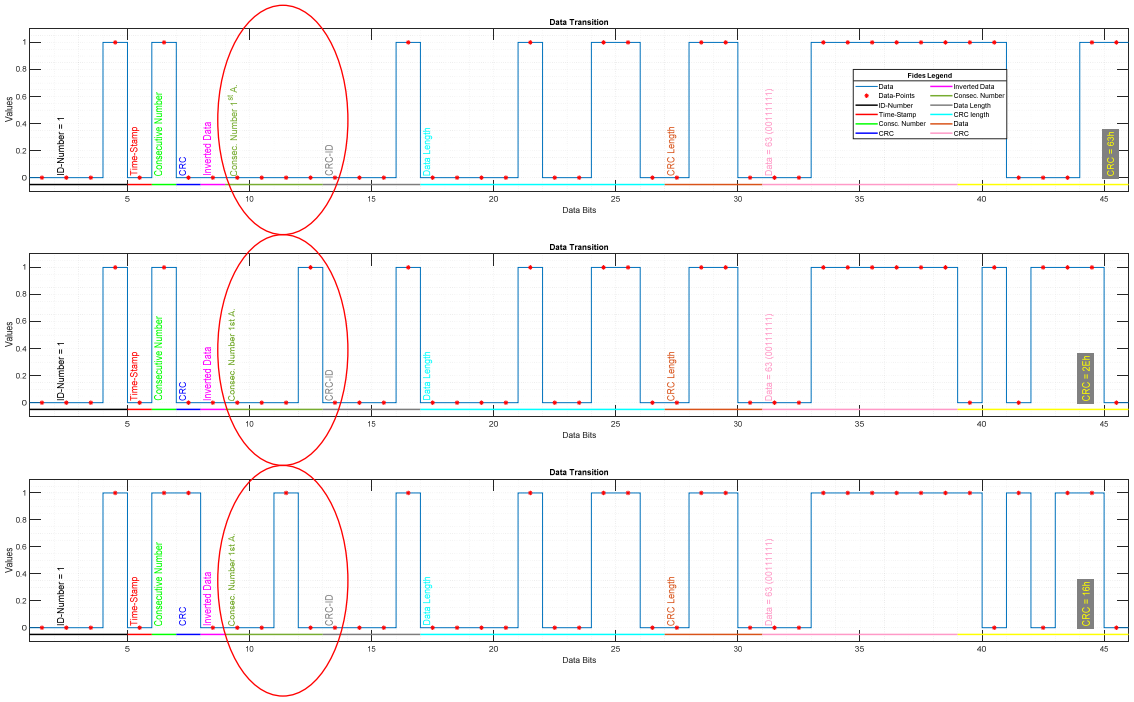


**Fig. 11.** Consecutive number.

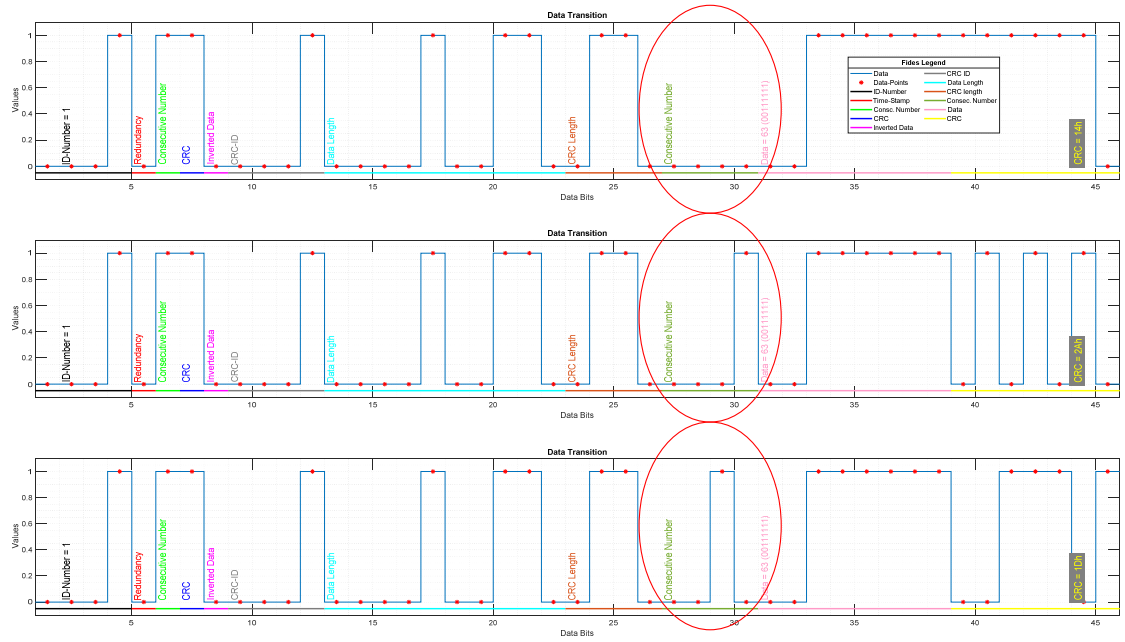


**Fig. 12.** Consecutive number in data section.

If the acknowledge is received on time, then everything is alright, otherwise if the message takes too long or does not reach its destination at all, then the receiver knows about this problem and can react on it, with a stop of the communication, or inform the next hierarchy about this problem [12].

Inverting the message or in the case of FIDES1 just inverting the data field is another method similar to a CRC to detect falsified messages. Equation (20) demonstrate this. The first line is the original message, which gets falsified in the second line when using an XOR function with the inverted data, then the number of falsified data bits can be identified. This is particular interesting when counting the number of errors in a message [12, 14].

$$\begin{array}{cccccccc} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \quad \textit{Original Message} \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \quad \textit{faulty Message} \\ & & & & \textit{XOR} & & & \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \quad \textit{Inverted Message} \\ \hline 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \quad \textit{Result} \end{array} \tag{20}$$

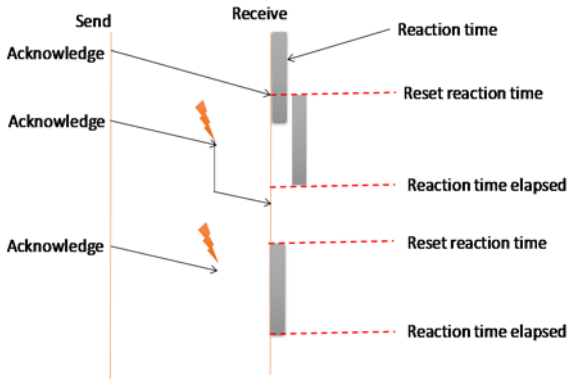


**Fig. 13.** Acknowledge time / reaction time.

## 5.3. Fides2

FIDES2 is the second approach and takes the found results from FIDES1 into considerations. Table 6 presents the new safety flags with the new features. Now the CRC cannot be switched off anymore, it can be selected between the two CRCs Tables 5 and 7 [12].

**Table 6.** Safety Flag Bit Field for FIDES2.

| ***Safety Flag*** | ***Bit interpretation*** | |
|---|---|---|
| | Bit length | Comments |
| 1 | 1 | Full redundancy |
| 2 | 1 | Message number |
| 3 | 1 | CRC group 1 or group 2 |
| 4 | 1 | Redundant, inverted data |

**Table 7.** CRC Group 2.

| ***Nr.*** | ***CRC*** | ***r*** | ***k*** | ***d*** |
|---|---|---|---|---|
| 0 | 0x12faa5 | 21 | 106 | 7 |
| 1 | 0x189efe | 22 | 105 | 7 |
| 2 | 0x5e2419 | 23 | 106 | 7 |
| 3 | 0x880ee6 | 24 | 231 | 7 |
| 4 | 0x289cfe | 22 | 105 | 8 |
| 5 | 0x469d7c | 23 | 105 | 8 |
| 6 | 0xcba785 | 24 | 105 | 8 |
| 7 | 0x4429686 | 27 | 48 | 9 |
| 8 | 0xeaa72ab | 28 | 99 | 9 |
| 9 | 0x1e150a87 | 29 | 100 | 9 |
| 10 | 0x242c0684 | 30 | 100 | 9 |
| 11 | 0x12b00d4 | 25 | 40 | 10 |
| 12 | 0x32def69 | 26 | 40 | 10 |
| 13 | 0x51aff9a | 27 | 41 | 10 |
| 14 | 0x1e150a87 | 29 | 100 | 10 |
| 15 | 0242c0684 | 30 | 100 | 10 |

This means that now 32 different CRCs are available with different lengths and different Hamming distances. The consecutive message number was kept, as it is a simple but important feature to detect errors such as:

- Repetition of a message;
- Loss of a message;
- Insertion of a message;
- Wrong sequence.

Fig. 14 shows the feature: consecutive number with transmitted data and the inverted transmitted data in one message. If the data is corrupted then a comparison of data and inverted data with the help of an XOR function can be performed, as explained earlier.

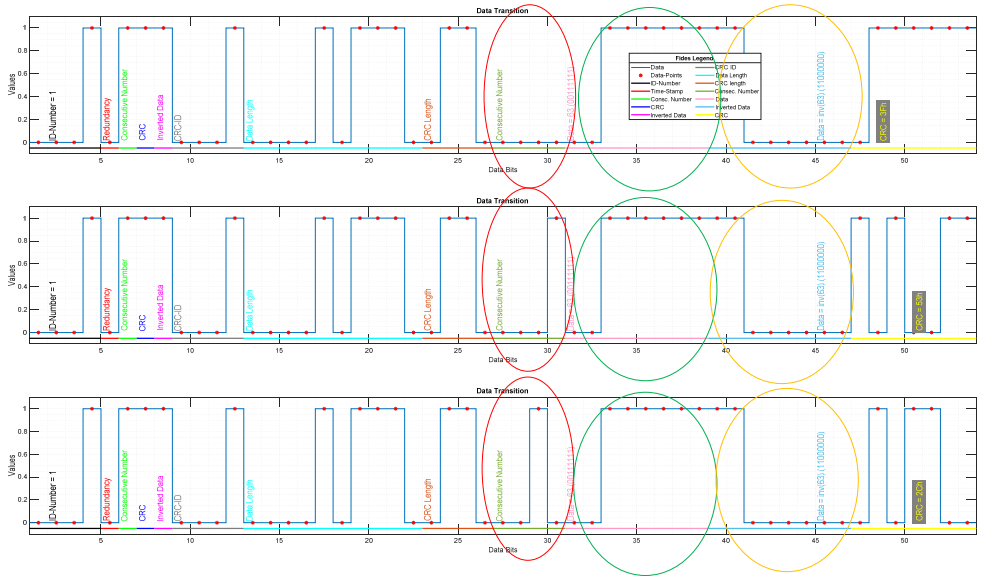


**Fig. 14.** Redundant data.

The new feature full redundancy is an entirely new message [25] where full redundancy bit is active and the message is complete inverted, besides the client / process ID the full redundancy bit and the CRC. If the ID would be inverted the client would not recognize this message. The inverted data frame/ full redundancy

is also not inverted so that the client can identify the this is a special message. The CRC is also not inverted, as the CRC should protect the sent message without understanding the meaning of safety flag bits. Fig. 15 should the original and the inverted message. It also shows the full redundancy and inverted data can be combined. The actual data is sent twice in one message and again twice in the second inverted message. From equation 20 it can be seen that it can be easily compared using an XOR function and counting the errors if present. Practically, this can also be used to try to correct the message, but this is not part of this investigation [12].

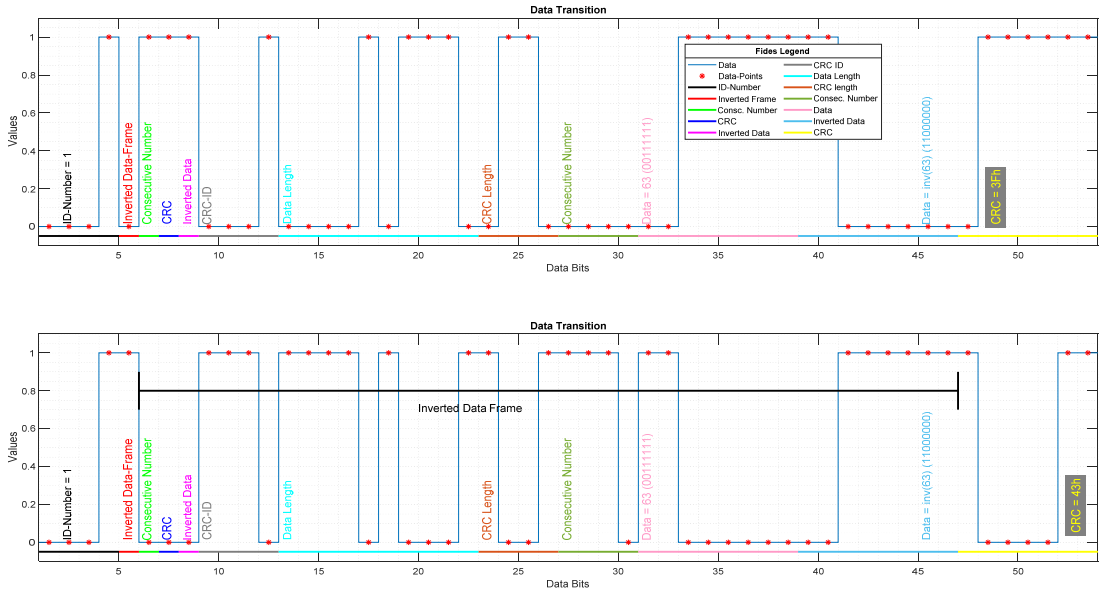


**Fig. 15.** Redundant message.

## 6. Data Error

The protocol structure submits the two parameters $k$ and $r$. $K$ can be calculated as follows:

$$k = Length(ID) + Length(CRC_{ID}) + \\ +Length(r) + Length(Data) \quad (21)$$

As the parameters are fixed, besides the length of the data, the equation becomes [12]:

$$k = 4Bit_{Length(ID)} + \quad (22)$$

$$+4Bit_{Length(CRC_{ID})} + 4Bit_{Length(r)} + \\ +Length(Data),$$

and k has to be smaller or equal as the maximum value of the specified CRC, also the value will be often a multiple of 8bits [12].

$$k \leq Length(\max(CRC_{ID})) \quad (23)$$

Fig. 16 illustrates that k and r defined the entire length of the message and most of the message's structure is fixed [12].

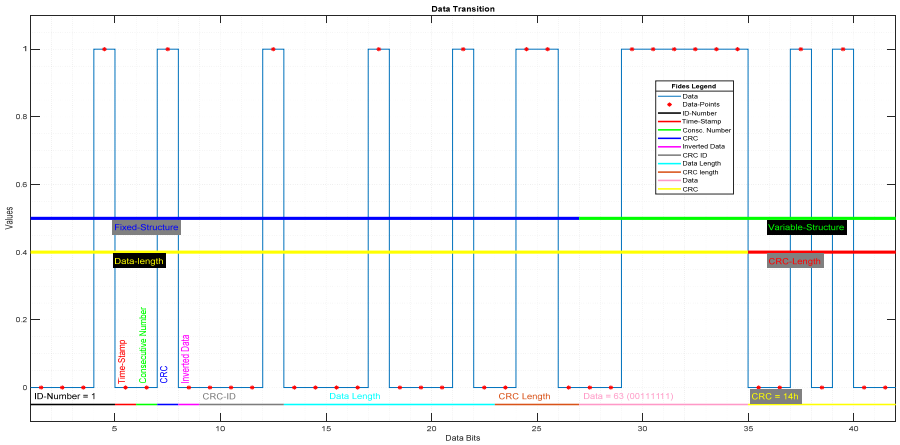


**Fig. 16.** Data loss.

As presented in the last section, if fully redundancy or data inverted redundancy is activated, then the errors can often be fully identified and counted also if data is lost it is counted as an error. Therefore, the loss or added data can be calculated as follows [12]:

$$BER_{Loss/Add} = \\ = \frac{\sum_0^n |Expected\ Bits - Received\ Bits|}{Expected\ Bits} \quad (24)$$

The data errors can be calculated as shown [12]:

$$BER_{\substack{Original \\ Inverted}} = \frac{\sum_0^n falisfied\ Bits}{Entire\ Data\ Bits} \quad (25)$$

Inserting equation (24) and (25) into equation (19), gets:

$$PFH = 36 \cdot 10^4 \cdot v \cdot \left( \frac{72}{121} \cdot \frac{\sqrt{2\pi n}}{2^r \cdot d!} \cdot n^d \left( BER_{Lost/Add} + BER_{Ori/Inv} \right)^d + 2^n \left( \sqrt{BER_{Loss/Add} + BER_{Ori/Inv}} \right)^n \right) \quad (26)$$

Finally, the equation becomes [12]:

$$PFH = 36 \cdot 10^4 \cdot v \cdot \left( \begin{array}{l} \frac{72}{121} \cdot \frac{\sqrt{2\pi n}}{2^r \cdot d!} \cdot n^d \cdot \left( \frac{\sum_0^n |Exp.Bits - Rec.Bits|}{n} + \frac{\sum_0^n falisfied\ Bits}{n} \right)^d + \\ +2^n \left( \sqrt{\frac{\sum_0^n |Exp.Bits - Rec.Bits|}{n} + \frac{\sum_0^n falisfied\ Bits}{n}} \right)^n \end{array} \right) \quad (27)$$

If the transition speed is known and the CRC is selected, then the PFH value can be calculated.

## 7. Decision-maker

For the first results, the following procedure was carried out. When no information is given about the PFH value and the transition speed is known, then the data is fit into a data frame with the smallest usable k and the lowest Hamming distance d. If this does not suit the PFH value, the next CRC with a higher Hamming distance is used. This continues until a CRC with the related Hamming distance fits or if the last one of the second CRC table (Table 7) is used and does not fit, then the communication has to be stopped. When two consecutive Hamming distances do not fit then the data redundancy is activated and if it is switched from first CRC table to the second CRC table then fully redundancy is activated. Fig. 17 illustrates the principle of the observation, where one system can keep its safety integrity and the second one (blue line) violates its integrity.

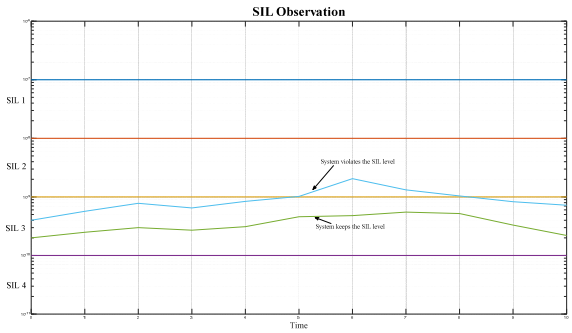


**Fig. 17.** Example of SIL observation.

## 8. Results

Initial tests are carried out. The new protocol was embedded in a wireless communication between two programmable logic controllers (PLC). The data was perturbed to investigate the change of the PFH-values and its SIL value. So far, the calculation and the switching worked well. When the redundant data is not sent, the CRC can only tell if the data is falsified or not. This means, it has to be estimated, how many errors are in the data. At the first glance, the maximum detectable number was used as a worst case. Further investigations have to be carried out.

## 9. Conclusions

New adaptive communication protocols were developed and both change their features depending on the current PFH value. Different CRCs are used for data protection depending on the data length, the Hamming distance and the length of the CRC itself. The switching points are at the first step acceptable but have to be modified.

### 9.1. Future Work

The switching conditions have to be more related to the actual PFH values, which is currently been investigated and going to be implemented. The observer has to be extended towards a predictor. For the estimation of the amount of errors in a message, when only a CRC is used, has to be developed. This represents an optimization problem, and an evolutionary algorithm seems to be appropriate. The minimal length of the message has to be adjusted if the

new protocols are embedded in some already existing protocols. An idea is to used stuffing bits at fixed or variable positions.

## Acknowledgements

This article is part of the project MEPHYsto_Zwei which is funded by the German federal state Hessen through the programme Distr@l.

We thank Distra@l (digitales.hessen) for the financial support, with the project-nr.: Distra@l 493/22 0018 2A

## References

[1]. B&R sichere Datenübertragung per Funk, https://www.br-automation.com/de-at/ueber-uns/presse/sichere-datenuebertragung-per-funk-24-09-2018

[2]. Schildknecht, https://www.schildknecht.ag/produkte/industrial-wireless/

[3]. D. F. N., Gordon, Y. Wu, X. An, Communication-aware motion control for mobile robot applications, in *Proceedings of the 4th IFSA Winter Conference on Automation, Robotics & Communications or Industry 4.0/5.0 (ARCI'2024)*, Innsbruck, Austria, 7-9 February 2024, pp. 327-331.

[4]. J. Wu, R. Li, X. An, C. Peng, et al., Toward native artificial intelligence in 6G networks: System design, architectures, and paradigms, *arXiv Preprint,* 2021, arXiv:2103.02823.

[5]. W. Ikram, N. F. Thornhill, Wireless communication in process automation: A survey of opportunities, requirements, concerns and challenges, in *Proceedings of the UKACC International Conference on Control*, 2010, pp. 1-6.

[6]. L. Gaus, M. Schwarz, J. Börcsök, A theoretical approach to a safety-based predictive adaptation of wireless communication channel parameters in harsh environments, *Safety in Extreme Environments*, Vol. 2, 2020., pp. 93-101.

[7]. E. Mueller, X.-L. Chen, R. Riedel, Challenges and requirements for the application of Industry 4.0: a special insight with the usage of cyber-physical system, *Chinese Journal of Mechanical Engineering*, Vol. 30, 2017, pp. 1050-1057.

[8]. M. N. O. Sadiku, Y. S. Wang, S. M. Cui Musa, G. Roy, Cyber-physical systems: a literature review, *European Scientific Journal*, Vol. 13, Issue 36, December 2017.

[9]. E. Ugljesa, H.-D. Wacker, J. Börcsök, Modelling security aspects in safety environment, in *Proceedings of the 7th International Conference on Electrical and Electronics Engineering (ELECO'11)*, 2011, pp. II-46-II-50.

[10]. A. Kara, Practical implications of the use of radio frequency fingerprinting (RFF) in cybersecurity of smart grids, in *Proceedings of the 4th IFSA Winter Conference on Automation, Robotics & Communications or Industry 4.0/5.0 (ARCI '2024)*, Innsbruck, Austria, 7-9 February 2024, pp. 346-348.

[11]. S. R. Rondla, Investigation of using artificial intelligence to detect safety critical intrusion, MD Thesis, *University of Kassel*, Kassel, Germany, 2023.

[12]. M. H. Schwarz, J. Börcsök, A communication observer for monitoring the safety integrity level in real-time, in *Proceedings of the 4th Winter IFSA Conference on Automation, Robotics & Communications or Industry 4.0/5.0 (ARCI' 2024)*, Innsbruck, Austria, 7-9 February 2024, pp. 343-348.

[13]. IEC 61508, International Standard 61508 Functional Safety of Electrical/Electronic/ Programmable Electronic Safety Related Systems, *International Electrochemical Commission*, Geneva, Switzerland, 2000.

[14]. P. K. Pendli, Contribution of Modelling and Analysis of Wireless Communication for Safety related Systems with Bluetooth Technology, *Kassel University Press*, 2014.

[15]. IEC 61784-3, Ed. 2.0: Industrial communication networks, Part 3: Functional safety fieldbuses, *International Electrochemical Commission*, Geneva, Switzerland, 2010.

[16]. CENELEC: En 50159-2. railway applications - communication, signalling and processing systems part 2: Safety related communication in open transmission systems Beuth Verlag GmbH, 2001

[17]. J. Peleska, O. Schulz, Reliability analysis of safety-related communication architectures, in *Proceedings of the International Conference on Computer Safety, Reliability, and Security*, 2010, pp. 1-14.

[18]. J. Åkerberg, Mobile Interaction with safety critical systems: A feasibility study, MD Thesis, *Mälardalen University*, Västerås, Sweden, 2015.

[19]. J. Åkerberg, M. Gidlund, F. Reichenbach, M. Björkman, Measurements on an industrial wireless HART network supporting PROFIsafe: a case study, in *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation: (ETFA'11)*, 2011, pp. 1-8.

[20]. ProfiSafe, https://www.profibus.com/ technology/profisafe

[21]. M. Friske, H. Schlingloff, H., Barthel, Verification and tests of PROFIsafe-safety-profile, in *Proceedings of the 26th Meeting of GI-FG TAV*, Stuttgart, Germany, 2007.

[22]. R. Pigan, M. Metter, Automatisieren mit Profinet, *Publics Corporate Publishing*, 2008 (in Germany).

[23]. B. Reißenweber, Feldbussystem zur industriellen Kommunikation 2. Auflage, *Oldenburg Industrieverlag GmbH*, München, 2002 (in Germany).

[24]. OpenSafety, https://www.ethernet-powerlink.org/open-safety

[25]. F. Haslhofer, OpenSafety for wired and wireless connections over MQTT, MD Thesis, *Johannes Kepler University*, Linz, Austria, 2020

[26]. B+R Automation, OpenSafety, Bernecker & Rainer, https://www.br-automation.com/enus/technologies/opensafety/

[27]. H.-D. Wacker, J. Boercsoek, The probability of undetected error of some communication channels, in *Proceedings of the European Safety and Reliability Conference (ESREL'07)*, June 2007, pp. 385-391.

[28]. H. D. Wacker, J. Boercsoek, Binomial and monotonic behavior of the probability of undetected error and the 2-r-Bound, *WSEAS Transactions on Communication*, Vol. 7, 2008, pp. 188-197.

[29]. L. Gaus, M. H. Schwarz, J. Börcsök, A mathematical approach to a real-time optimization of safety parameters in wireless communication systems, in *Proceedings of the 4th Workshop & Symposium on*

*Safety and Integrity Management of Operations in Harsh Environments*, St. John's, NL, Canada, 2019.
[30]. M. H. Schwarz, L. Gaus, J. Börcsök, Investigation of a safety parameter observer for wireless communication, *International Journal of Circuits, Systems and Signal Processing*, Vol. 14, 2020, pp. 1005-1010.
[31]. T. Sköllermo, M. Skoglund, A sub band image coder for channels with both errors and erasures, in *Proceedings of the 37th Asilomar Conference on Signals, Systems & Computers*, 9-12 November 2003, pp. 1553-1557.
[32]. Ph. Koopman, Best CRC Polynomials, https://users.ece.cmu.edu/~koopman/crc/