

IDA Safety
Data Transmission Protocol

Table of Contents

Table of Contents	2
1. IDA-Safety	3
2. Introduction.....	3
3. Ethernet Data Transmission and Transport Structures	3
3.1. Ethernet Today.....	3
3.2. Ethernet Data Format	4
3.3. RTPS as Middleware	5
4. Possible Communication Errors.....	6
5. Structure and Area of Application of the Safety Ethernet	8
5.1. Communication Within the Safety Area	8
5.2. Direct Sensor/Actuator Communication	9
6. Technology and Measures for Error Detection and Error Removal	10
6.1. The IDA Safety Data Format.....	10
6.2. Format.....	11
6.3. Elements	13
6.3.1. ID Information.....	13
6.3.2. Time Information and Sequence (Order)	13
6.3.3. Variable Name.....	14
6.3.4. Data Field.....	15
6.3.5. Data Protection.....	15
6.4. Safety Data Traffic.....	15
6.4.1. Detection of Damaged or Incorrect Data Records	15
6.4.2. Checking the Data Transmission Quality	16
7. Conformity With SIL Requirements and Technical Conditions	17
7.1. Calculation of Residual Error Probability.....	17
7.2. Possible Procedure for Generating a Data Format	19
7.3. Bibliography.....	20

1. IDA-Safety

This document gives an overview about the principle of operation of IDA-Safety. The focus is set to the safety-related data transfer, the format and the services. For more information concerning IDA, please find: www.ida-group.org.

2. Introduction

Within the IDA organization, the IDA Safety working group has undertaken the task of developing a data transfer profile, which can be used to transmit safety data. The necessary data format must meet the requirements of EN 954-1 (Category 4). An appropriate level of redundancy must also be provided in the protocol and a procedure should describe how errors can be detected and removed according to SIL 3 (see EN 61508).

3. Ethernet Data Transmission and Transport Structures

3.1. Ethernet Today

Due to its diverse areas of application for office communications, computer networking, PC networking, and Internet technology, Ethernet has grown far more quickly than any other network. Although numerous local area networks are now available, such as Profibus, INTERBUS, ASI, DeviceNet, CAN, Safety-Bus P, Modbus, etc., Ethernet leads the way due to its considerable advantages:

- Internationally accepted
- Integration into numerous operating systems (UNIX, Windows, etc.)
- Fast data access
- High level of availability
- Wide-ranging compatibility
- Simple interface technology
- Supported by the Internet

It can therefore be expected that Ethernet will gradually replace and standardize the local area networks that are already available.

In recent years, countless local area networks have been modified so that they are suitable for the transmission of safety data (e.g., Profibus F, INTERBUS Safety, CAN Open Safety, etc.). Local area networks have also been developed, which are designed specifically for communication in the safety area (e.g., Safety-Bus P).

At present, Ethernet cannot use a standard layer for safe data transfer. There are only proprietary solutions, which support company-specific communication procedures (e.g., HIMA or TTTech-Ethernet).

The IDA working group believes that the integration of a safety profile for Ethernet should also provide solutions for the distribution of intelligence, because optimal and fast communication will not be achieved in the future without this development.

3.2. Ethernet Data Format

Ethernet uses a precisely defined data format, which is shown in Figure 3.2.1.

Pre 7	Strt 1	DA 6	SA 6	Len 2	Data 46-1500	Pad 0-46	CRC 4
----------	-----------	---------	---------	----------	-----------------	-------------	----------

Figure 3.2.1: Ethernet data format

Each Ethernet transmission begins with a preamble. A special start sequence follows this preamble. Ethernet then transmits the receiver address, sender address, and total data length. The actual data field must contain at least 46 bytes. If fewer bytes are to be transmitted, then empty bytes should be inserted (padding). For data protection, a CRC 32 test polynomial is provided at the end.

Table 3.2.2 provides an overview of the data string:

Name	Meaning	Length
Pre	Preamble	7 bytes 56 bits
Strt	Starting byte: This byte contains the following information: 10101011	1 byte 8 bits
DA	Destination address	6 bytes 48 bits
SA	Source address	6 bytes 48 bits
Len	Length	2 bytes 16 bits
Data	Data: Data to be transferred	Minimum 46 bytes Maximum 1500 bytes i.e., 368 bits, minimum
Pad	Padding: Padding bytes are added if the number of transferred bytes is less than 46	0-46 bytes
CRC	Cyclic Redundancy Check CRC-32 for error detection	4 bytes 32 bits

Table 3.2.2 The standard Ethernet data format

3.3.RTPS as Middleware

The IDA group decided to integrate special middleware for the "distributed intelligence" area of application. This additional layer is used for the quick and optimal transmission of data within a system with distributed resources. The middleware is known as RTPS (Real Time Publish/Subscribe) and was developed by RTI.

The following diagram illustrates the internal Ethernet structure:

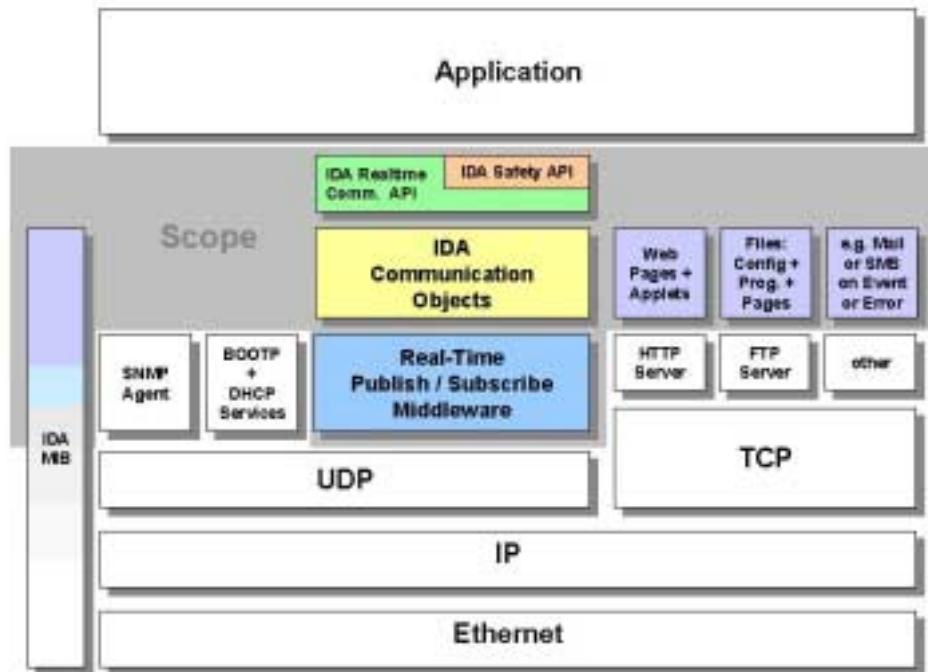


Figure 3.3.1 Illustration of the safety layer structure

As can be seen in Figure 3.3.1, the middleware (RTPS) forms the intermediate layer between the UDP (User Datagram Protocol) and the Realtime+Safety layer. All safety messages always communicate via UDP. This service is not acknowledged (unlike TCP, which acknowledges every data transfer). This means that responses are quicker (in terms of distributed intelligence).

Each layer within a layer model requires an additional overhead of data. Consequently, the RTPS (Real Time Publish Subscribe) requires additional bytes within the normal data information.

Ethernet provides a data field with a maximum of 1500 bytes on ISO/OSI layer 2. Above Ethernet is a layered structure, in which each layer (including the safety layer) performs special tasks.

The actual Ethernet data field with safety data comprises the following parts:

20 bytes: IP (IP: Internet Protocol)
8 bytes: UDP (UDP: User Datagram Protocol)
20 bytes: RTPS (RTPS: Real Time Publish Subscribe)

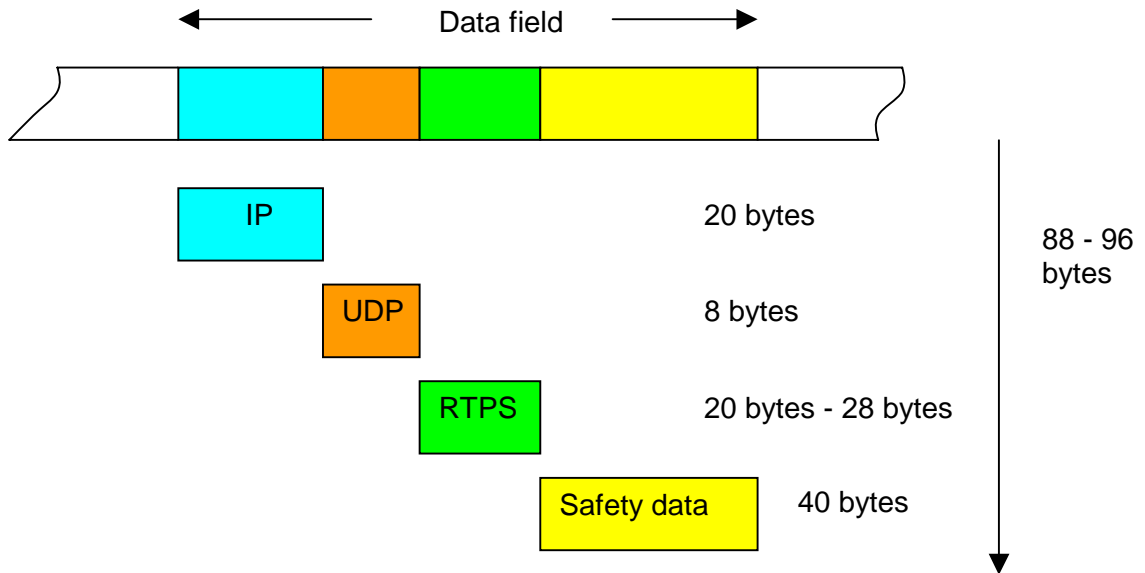


Figure 3.3.2 The safety data is embedded in the entire data format

4. Possible Communication Errors

Ethernet already provides very secure data transfer. The high level of security is due, in particular, to the fact that each data string is protected by a CRC-32 polynomial at the end.

The polynomial checks for the following errors:

- Error sequences, which are less than 32 bits
- All odd-numbered error cases
- All multiple errors, if at least 4 errors are present

A Hamming distance of 6 is achieved with this CRC unit.

Unfortunately, this internal test method is no guarantee for safe communication. For example, the CRC unit may fail and positively acknowledge all data. This failure of the CRC unit at the receiver would mean that all data is always correct.

This single error is not acceptable if the system is to conform with Category 4 (according to SIL 3).

In addition, the transmission of correct data alone does not guarantee a perfectly operating safety function. For example, the transmitter may have failed, and may send the same data without interruption. It is often impossible to distinguish between this data and updated data, because changes in status are rare in automation systems. Therefore, each transmitter must send a sign of life, which provides clear information about the proper operation of the system.

In principle, data strings may be repeated, lost or have an incorrect sequence. It must be possible to reliably detect all of these errors. These errors may be caused by EMI, refraction, system errors or runtime errors.

The draft standard (FAET paper) contains information on the detection of these error cases:

Measures Errors	Cons. No.	Time Mark	Echo	Identifier	Data Prot.	Red. Cross.
	↓	↓	↓	↓	↓	↓
Repetition	●	●				●
Loss	●		●			●
Insertion	●		●	●		●
Incorrect sequence	●	●				●
Delay		●				
Incorrect data			●		●	

Table 4.1 Measures according to FAET draft and their effectiveness

As can be seen in the table, two measures are particularly useful during data transmission:

- Introduction of error detection for the detection of single and multiple bit errors
- Addition of a consecutive number, which acts as a sign of life or provides information about the correct sequence of the data string

In addition to these measures, each device must contain a watchdog, which sets the actuator to a safe state if valid data is not received.

This watchdog ensures that dangerous states after a network failure that last for more than the reaction time and long interruptions during data traffic (due to external activities or error bursts) are detected immediately.

The function of this type of watchdog is quite straightforward. When the correct data is received (data without bit errors and correct consecutive number), the watchdog is reset. If the watchdog is not reset within the specified reaction time, it sets the actuator to a safe state (as a rule, a digital output is loaded with the value 0).

Exponential values can be used to calculate error rates for digital data transmission. The following table provides an overview of the data:

Bit Error Probability	Transmission System
$> 10^{-3}$	Radio link
10^{-4}	Telephone cable
10^{-3} to 10^{-5}	Twisted pair cable
10^{-6} to 10^{-7}	Digital telephone
10^{-9}	Coaxial cable
10^{-12}	Fiber optics

Table 4.2 Comparison of typical bit error rates for various media
(See M. Schäfer, New concepts for safety bus systems, details in Bibliography)

Despite all of these measures, a certain residual error probability remains, i.e., a risk that incorrect information could be identified as correct. According to standard EN 61508 the following values are acceptable (depending on the Category required):

SIL	Probability of a Hazardous Error per Hour in Uninterrupted Operation Mode
3	$<10^{-8} \dots <10^{-7}$
2	$10^{-7} \dots <10^{-6}$
1	$10^{-6} \dots <10^{-5}$

Table 4.3 Residual error rate requirements according to SIL

For transmission within a safety network, experience shows that a mere 1% of the susceptibility of the entire system to errors depends on the error rate in the network itself.

Errors in the software, system errors, hardware failures, etc. have a far greater effect. Taking this into account, data transmission should exceed the SIL requirements by at least factor 100. To achieve Category 4 (SIL 3), the residual error probability must be more than 10^{-9} .

5. Structure and Area of Application of the Safety Ethernet

5.1. Communication Within the Safety Area

The statements made below for safe data transfer refer to a special safety area. This safety area is shielded from the rest of the system by gateways.

The following diagram illustrates this type of safety area.

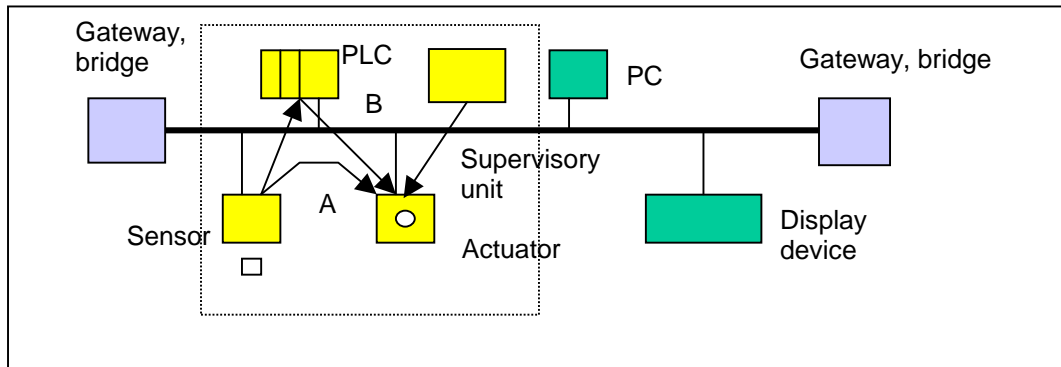


Figure 5.1.1 Typical structure and components of a safety area
The components that provide safety functions are enclosed in a box.

As can be seen in Figure 5.1.1, both safety and normal components can participate in communication in the safety area.

The number of units involved in the entire process, and their functions, is specified during system configuration. This prevents any ambiguous relationships during communication. However, a communication failure could still occur if, for example, a normal component were to behave in an unexpected manner. The communication services and watchdog functions must be able to handle these types of imponderables.

The provision of the safety function within the safety area depends on the specific application. The following options are possible:

- Use of a safety control system with cyclic data traffic (B)
- Direct communication between sensors and actuators (A)
- Use of a supervisory unit to check safety transmissions (B)

Safety data can of course also be transferred globally via one or more gateways. Additional services and data formats are available in these cases, which are not discussed in this document. It should be noted that the reaction times for the calculation of safe shutdown do not apply for these types of global data transfer.

5.2. Direct Sensor/Actuator Communication

According to the stated aim of the IDA working group, in the future only safety communication should take place between sensors and actuators. These data transfers can be achieved in a very short period of time, and simplify the structure of the entire system. This eliminates the need for PLCs, global microcomputers, and supervisory units. Of course, the functions within the actuator must be improved to achieve this direct communication (Figure 5.2.1).

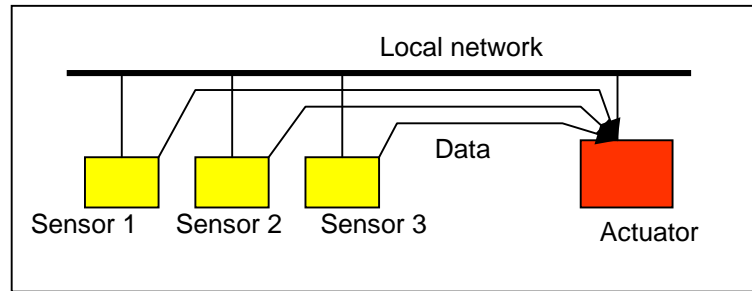


Figure 5.2.1 Direct communication between several sensors and one actuator

For example, an actuator receives its programmable function from 3 different sensors (sensors 1, 2, and 3). Each sensor makes its logical function available to Ethernet. In principle, they do not know the data destination. This means that another actuator may also want to receive the data from one of the sensors. The internal organization of RTPS means that the desired data can be transferred to any actuator within the required time period.

The actuator must be able to internally perform the logical function and complete the necessary operation. Consequently, an IDA actuator can usually be programmed and parameterized. This means that an IDA actuator consists of both the actuator function itself (e.g., a drive) and a programmable logic part, which contains all necessary IDA services. For safety applications, the relevant safety hardware and software is also required. The safety part should be integrated according to Figure 3.3.1.

6. Technology and Measures for Error Detection and Error Removal

6.1. The IDA Safety Data Format

In order to meet the requirements of SIL 3 (see Section 4), a data format is specified below, which ensures a high level of error detection probability. The format provides the actual safety data with additional information, according to the profile in Table 4.1.

The entire data format is therefore part of the data information of the remaining data width (in addition to IP, UDP, and RTPS). (See also Figure 3.3.2, Safety Data).

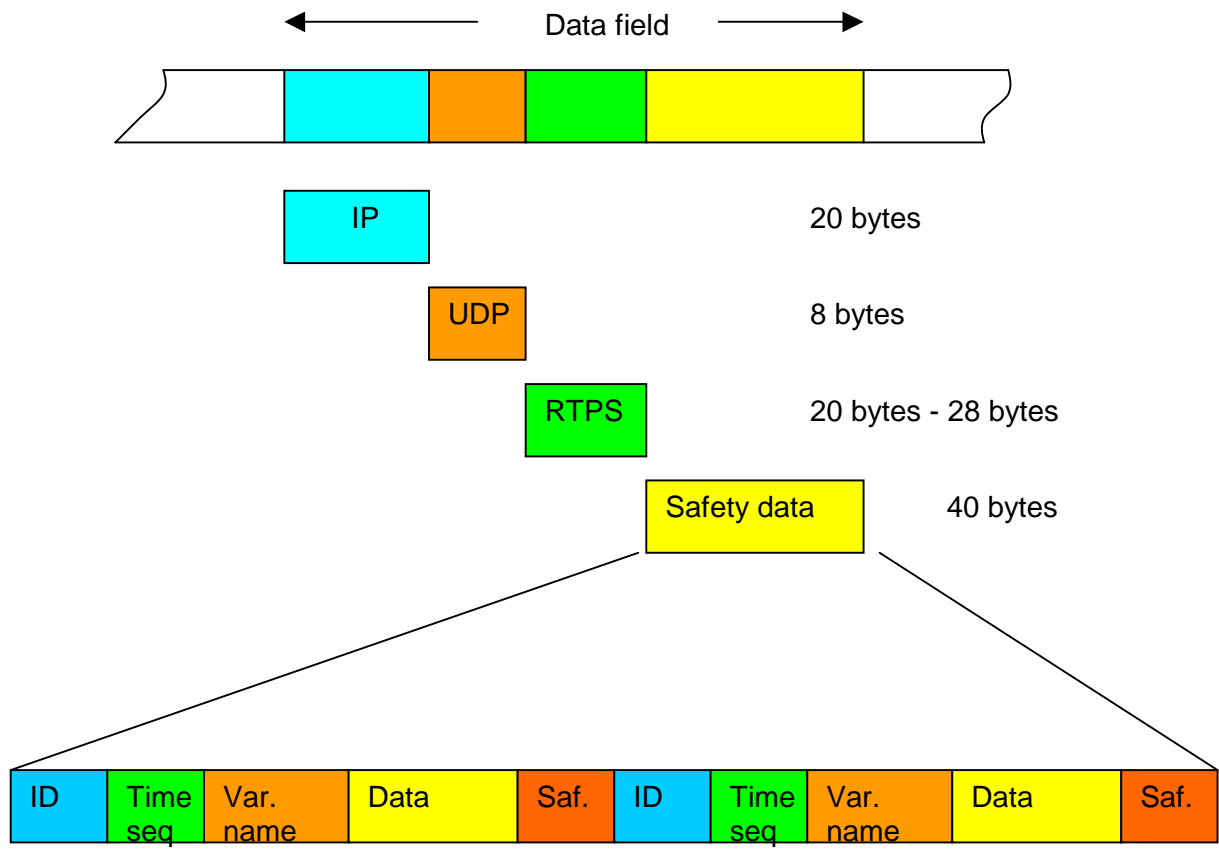


Figure 6.1.1 Internal structure of the data field for safety data

As shown in Figure 6.1.1, the IDA Safety data information consists of two data blocks, which have the same structure. The second data block (underlined) contains all the data in inverted form. In addition to the data itself, each individual block contains the variable name and the incremental time code with sequential number. The data record also contains an ID code and a safety sequence.

6.2. Format

The total length of the safety data is 40 bytes (or 320 bits).

The following table illustrates the data sequence (Table 6.2.1):

Byte No.	Contents	Meaning	Basic Data Type 1)	Value/Comment
1	Safety ID	MSB of ID information	IDA_BYTE	0: No chaining of safety data formats > 0: To be defined
2	Safety ID	LSB of ID information	IDA_BYTE	
3	Consecutive timer	MSB of timer information	IDA_BYTE	0 for first run 1 to 65535
4	Consecutive timer	LSB of timer information	IDA_BYTE	
5	Variable name	MSB of variable name		

6 Variable name	LSB of variable name	IDA_ARRAY[4] of BYTE	To be defined
7 Variable name			
8 Variable name			
9 Data	MSB of data	IDA_ARRAY[8] of BYTE	Depends on application
10 Data			
11 Data			
12 Data			
13 Data			
14 Data			
15 Data			
16 Data			
17 Sum	Sum of all bytes 1-16 (mod)	IDA_BYTE	
18 Parity 1	Parity of bytes 1-8 8: MSB, 1: LSB	IDA_BYTE	Even parity 2)
19 Parity 2	Parity of bytes 9-16 16: MSB, 9: LSB	IDA_BYTE	Even parity 2)
20	Spare	IDA_BYTE	To be defined
21 Safety ID	MSB of ID information	IDA_BYTE	0: No chaining of safety data formats > 0: To be defined Each byte inverted
22 Safety ID	LSB of ID information	IDA_BYTE	
23 Consecutive timer	MSB of timer information	IDA_BYTE	0 for first run 1 to 65535 Each byte inverted
24 Consecutive timer	LSB of timer information	IDA_BYTE	
25 Variable name	MSB of variable name	IDA_ARRAY[4] of BYTE	To be defined
26 Variable name			
27 Variable name			
28 Variable name			
29 Data	MSB of data byte 1	IDA_ARRAY[8] of BYTE	Depends on application
30 Data			
31 Data			
32 Data			
33 Data			
34 Data			
35 Data			
36 Data			
37 Sum	Sum of all bytes 21-36 (mod)	IDA_BYTE	
38 Parity 1	Parity of bytes 21-28 28: MSB, 21: LSB	IDA_BYTE	Even parity 2)
39 Parity 2	Parity of bytes 9-16 36: MSB, 29: LSB	IDA_BYTE	Even parity 2)
40	Spare	IDA_BYTE	To be defined

1) Data format identical to IEC 1131-3 format

2) Supplement to even number of 1's

Table 6.2.1 Overview of data bytes in the safety data field

6.3. Elements

6.3.1. ID Information

The entire data record begins with the ID information. At present, this consists of 2 bytes, which contain 0. In future applications, it will be possible to specify other contents, which can be used to identify the data record.

6.3.2. Time Information and Sequence (Order)

Safe data transfer requires both correct data and data transfer services, which transfer this data with a guaranteed reaction time.

This reaction time should be as short as possible so that cyclic and sudden intervention by users in a danger zone cannot lead to injuries. If the maximum reaction time is exceeded, there is a risk that a cutting or punching device may be too slow to detect that a person has placed his/her hand under a photoelectric barrier to remove the machined material. The machining process would not be aborted until it was too late and an injury would be inevitable.

With any network topology, and when additional devices are used (switches, gateways, etc.), the data runtime can fluctuate significantly. In particular, it can no longer be calculated deterministically because it also depends on the network load. However, it must always be ensured that an important safety data item will be transferred and processed in the guaranteed reaction time.

In normal local networks, this requirement is met by assigning a consecutive number to each message, which is monitored on the receiver side. If the numerical sequence is not observed or no number is received within a specific time interval, the watchdog in the actuator initiates a safety function. For Ethernet, the consecutive number is only used to check the correct sequence. The reaction time cannot be monitored because the runtime within the network can rise imperceptibly but steadily (e.g., due to the constant addition of data within the Ethernet stack). The consecutive numbering is then in the correct sequence and also within the required reaction time. However, there is no information about the actual data runtime, which means that a safety reaction may be too late.

The consecutive timer information indicates whether the order is correct and the data is up-to-date.

The following conventions apply:

D15 D14	:	00	Timer in microseconds
D15 D14	:	01	Timer in 10 microseconds
D15 D14	:	10	Timer in 100 microseconds
D15 D14	:	11	(free for future applications)

The consecutive timer information can be used to cover a time period from a few microseconds to over 1.6 seconds. A receiver can use this time information to determine whether the order is correct and whether the data is up-to-date.

The procedure used does not require synchronization of the clocks within the individual devices, but operates using the transmission of a relative time, which is determined in a data

transmission interval. As can be seen in Figure 6.3.2, the transmitter provides the receiver with data without interruption (top half of the diagram). As the runtime of the data is not known and can also fluctuate significantly, the receiver does not know whether the data is already out-of-date. To determine whether the data is up-to-date, the receiver sends a time request to the transmitter in a regular sequence (bottom half of the diagram). The transmitter acknowledges this request with its internal time information (confirmation). This enables the receiver to compare the transmitter's internal clock with its own time. Even with this data exchange, the runtime of the information is not known. However, the following statement can be made: Both clocks can be adjusted relative to one another, whereby the time error is as close as possible to the response time. If this response time is small enough, the adjustment was perfect. Any other information from the transmitter also contains the relative time and the receiver can now determine whether it has received a data record on time within the reaction interval.

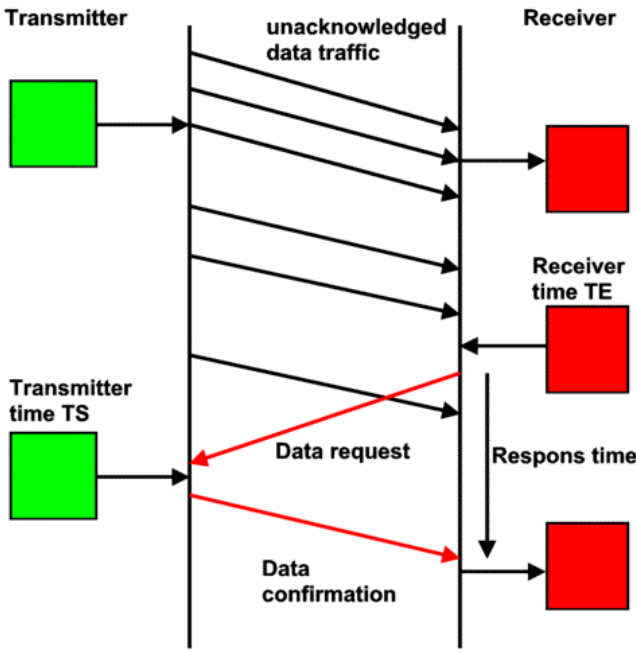


Figure 6.3.2 Relative synchronization of clocks within devices

6.3.3. Variable Name

The variable name indicates the type of data contained in the data field. The formatting and data area of the variable name are based on standard DIN IEC 1131-3.

Section 2.2 (of IEC 1131-3) is particularly relevant. It should be noted at this point that safety data is usually information, which can be represented as numbers or digits. String variables are not permitted or should only be used subject to certain restrictions.

6.3.4. Data Field

The data field consists of a total of 8 bytes, which represent the actual data content of the information. The data field is therefore the variable, which is (or was) to be transferred. The variable name field indicates the meaning of this variable.

6.3.5. Data Protection

Data protection is implemented and checked using the cross checking procedure. Cross checking refers to the following data fields.

- ID field (identifier)
- Consecutive timer field
- Variable name
- Data field

The entire summation should be carried out as follows:

	ID1	ID2	CT 1	CT2	VN1	VN2	VN3	VN4	D1	D2	D3	D4	D5	D6	D7	D8	Sum	P1	P2
Bit7		●							D17	D27	D37	D47	D57	D67	D77	D87	ID17-D87	P ID1	PD1
Bit6		↓							D16							D86	ID16-D86	P ID2	PD2
Bit5									D15							D85	ID15-D85	P CT1	PD3
Bit4									D14							D84	ID14-D84	P CT2	PD4
Bit3	●								D13						→	D83	ID13-D83	P VA1	PD5
Bit2									D12							D82	ID12-D82	P VA2	PD6
Bit1									D11							D81	ID11-D81	P VA3	PD7
Bit0		↓							D10	D20	D30	D40	D50	D60	D70	D80	ID10-D80	P VA4	PD8
	P ID1	P ID2	P CT1	P CT2	P VA1	P VN2	P VN3	P VN4	P D1	P D2	P D3	P D4	P D5	P D6	P D7	P D8			
	P1 (Parity 1)								P2 (Parity 2)										

Figure 6.3.5.1 Cross checking procedure

As can be seen in Figure 6.3.5.1, each data record receives 1 byte for the summation and 2 bytes for the parity. The data in the field highlighted in yellow is transmitted.

This cross checking procedure leads to a Hamming distance of 4 for the illustrated data record. Taking into account the fact that the entire data record is sent again in inverted form, this gives an error detection probability corresponding to a Hamming distance of 8.

6.4. Safety Data Traffic

6.4.1. Detection of Damaged or Incorrect Data Records

Each receiver (usually an actuator) has an internal storage area, in which it records the contents and states of all the transmitters (usually sensors). In particular, the data field indicates whether the transmission was completed without problems and within the desired time.

To ensure the correct structural management of this storage area, the transmitter and receiver must agree on the following before the start of data transfer:

- The internal watchdog time for the maximum reaction time specifies the maximum period that the receiver (actuator) can wait for a data item before it must enter a safe state or trigger an appropriate safety function for the specific transmitter (sensor).
- The data transfer repetition time is the time interval in which the transmitter receives at least one error-free message from the receiver.

If no error-free message is received within the data transfer repetition time (i.e., message is damaged or no message arrives), this does not usually trigger a programmed safety function. It is simply recorded that a transmission error has occurred. A safety function is only triggered when no correct message is received within the watchdog period.

As a rule, the watchdog time is 3-4 times the data transfer repetition time.

The following table (Table 6.4.1.1) provides an overview of the actions in the receiver for a specific transmitter:

Action	Reaction		
	Message counter (Good)	Message counter (Bad)	Reset watchdog
Message was received in the correct time interval upon the first attempt and contains a time stamp, which enables updating	+1	-	Reset
Message is damaged		+1	
Message is OK, but the time stamp corresponds to the last interval, so the data is not up-to-date	-	-	-
Message does not arrive within the data transfer repetition time		+1	

Table 6.4.1.1 Overview of actions/reactions in the receiver

If the message counter has been increased, the message is an error-free message, which also leads to a watchdog reset.

6.4.2. Checking the Data Transmission Quality

Numerous measures are available for monitoring the data transmission quality. For data transfer within the IDA, a special procedure has been developed, which does not require any other layers or services. The IDA Safety API can independently determine the quality of safe data transfer and initiate a safety action if a specified level of quality is not maintained. As a rule, the safety node remains connected to Ethernet, but the relevant output enters a failsafe state, which prevents any hazards arising from the unit.

For quality purposes, a host processor (software) is integrated in the safety layer, which uses a decision table to determine how many messages are registered as error-free (events: GOOD) and how many are registered as missing or damaged (events: BAD). Each event is recorded in a counter. Depending on the application, the individual counters (GOOD or BAD)

are of different lengths. When the GOOD counter overflows (or reaches a specific maximum value), the data transmission quality is calculated by dividing the contents of the BAD counter by the contents of the GOOD counter (Figure 6.4.1).

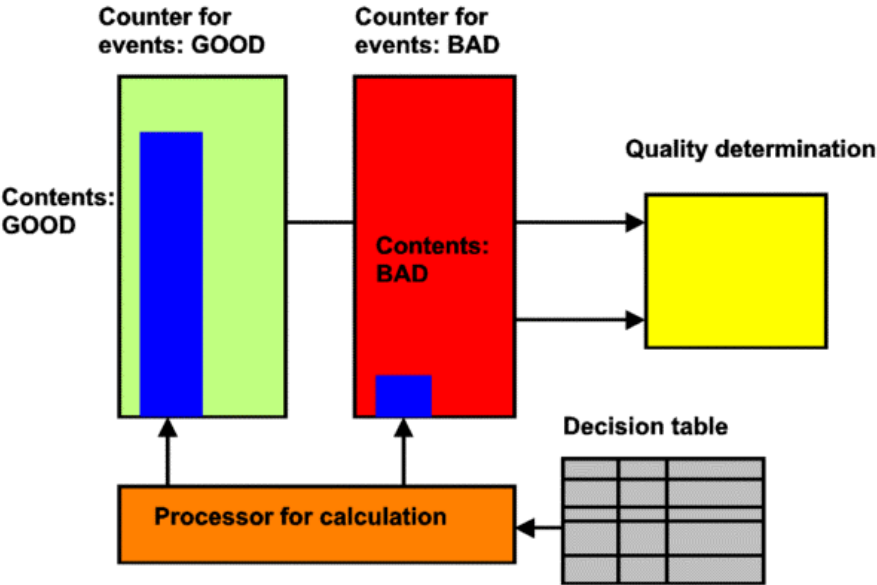


Figure 6.4.2.1 Determining the data transmission quality using a counting procedure

7. Conformity With SIL Requirements and Technical Conditions

7.1. Calculation of Residual Error Probability

The data format described in Section 6 ensures a Hamming distance of 8. This means that each error or multiple error up to 7 bits is safely detected. A higher number of errors will only remain undetected if specific bit positions change in a precisely defined way. Depending on the data content, the user can easily achieve additional redundancy by specifying the possible data structures. The Hamming distance can also be set to a much higher value.

For the calculation of the residual error probability given below, a Hamming distance of 8 is used, with precisely 320 bits.

Note: The error probability indicates how likely it is that an error will occur. This value applies for a special configuration and a particular type of transmission. The residual error probability indicates the errors that can still be expected after application of the selected measures.

Figure 7.1.1 illustrates the relationship between the bit error rate and the target residual error probability or Lambda value.

The functions shown are for typical reaction times of 20 milliseconds where only one data item leads to a safety shutdown.

The diagram contains the graphs for typical bit error rates for data transmission on serial networks with a degree of significance between 0.01 (or 10^{-2}) and 0.0001 (or 10^{-4}).

The graph only shows the total Lambda value. Additional errors in the system have not been taken into account.

The mathematical formulae used to represent the functions are given in Section 9.

Note on the values used:

Lambda value:

The Lambda value indicates the probability that an error will not be detected. A time interval of 1 hour is used as the reference time. The number of relevant protocols or the relevant reaction time must be used in the calculation.

A Lambda value of 10^{-9} means that only one undetected error will occur within a time period of 1100 years. The Lambda value does not indicate whether this error will lead to a fatal system failure, or whether the system will repair the error immediately and thus return to a safe state.

- R(p): Residual error probability for the system with error detection.
- v: System reaction time (at least one message is expected in this time).
- k: Number of devices, which must exchange safety data to trigger a shutdown (the value 1 is used here, because usually one sensor requests a shutdown).

The Lambda value is calculated as follows (see also Section 8):

$$\Lambda = 3600 \cdot R(p) \cdot v \cdot k$$

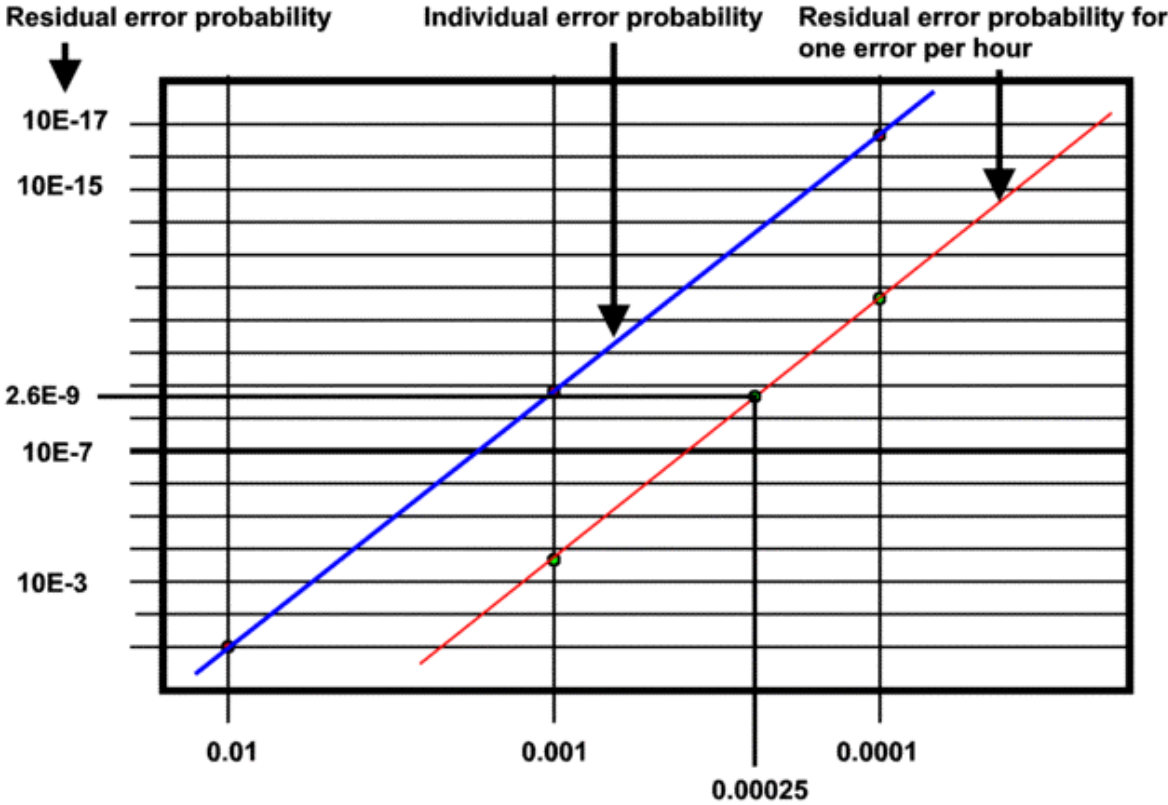


Figure 7.1.1 Lambda values for the transmitted data bits

Category 4 or SIL 3 (according to ISO 61508) is achieved if the system ensures a Lambda value that is less than 10^{-7} . As other components apart from the bus system can also fail or cause faults due to errors, the actual error rate for the transmission is often specified as 0.01 (see following section). Consequently, only a bit error rate of approximately 0.00025 is permitted, as can be seen in the diagram (Figure 7.1.1).

A quality monitor (see Section 6.4.2) can be used to precisely determine the bit error rate.

7.2. Possible Procedure for Generating a Data Format

Figure 7.3.1 provides a basic diagram, which can be used to implement the desired data format.

There are 2 microcontrollers (controllers 1 and 2) on the transmitter side, which determine the sensor value completely independently and monitor each other's functions. Both controllers then (independently) generate the data format with the header, consecutive number, and variable name.

Both controllers provide this data redundantly to the network. The second controller inverts all of its data before transmission. The data may be placed in a buffer before data transfer.

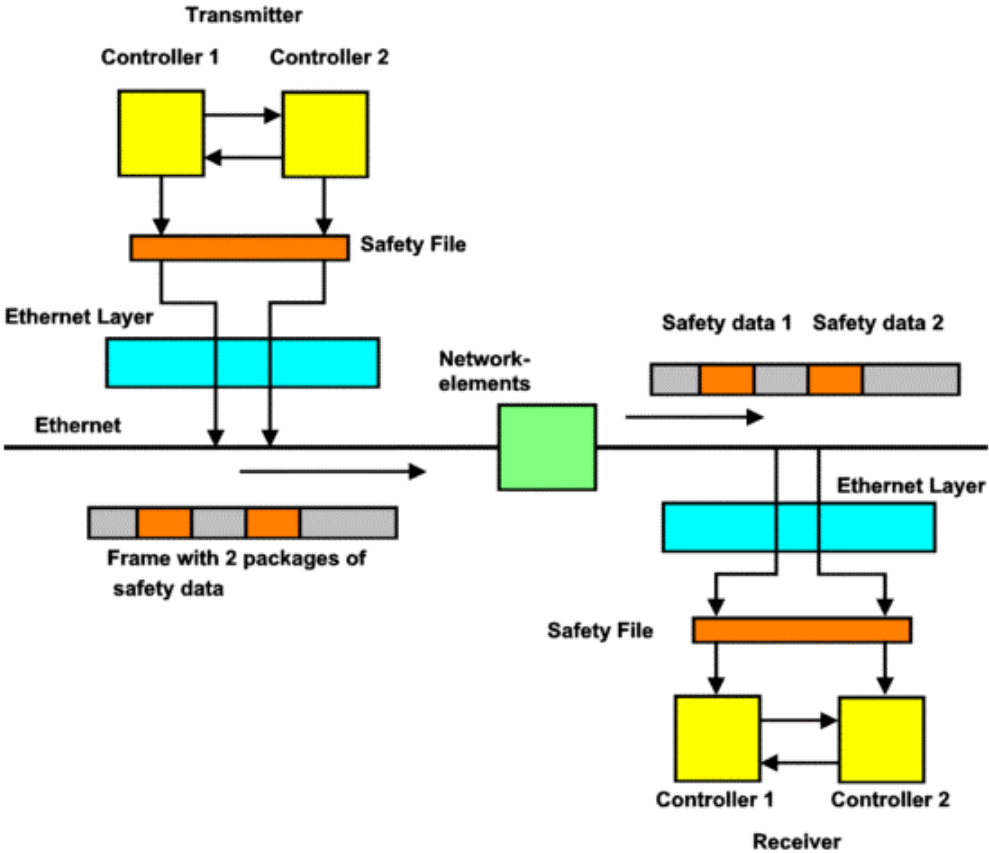


Figure 7.3.1 Example of safety data generation

However, another option is that one of the controllers generates all the data, and the second controller then checks the contents (in the buffer, for example).

Once all the data has been generated, Ethernet sends both data structures as one data item.

The receiver then performs the same process in reverse order. It transfers both blocks to the controllers, which check and compare the data contents.

7.3. Bibliography

M. Schaefer, D. Reinert: Bus-Software mit Feuermelder [Bus software with fire alarm], published by Hüthig, Volume 8/98

M. Schaefer: New concepts for safety bus systems, BIA St. Augustin

P. Schicker: Datenübertragung und Rechnernetze [Data transmission and computer networks], published by Teubner, ISBN 3-519-02463-2

M. Bossert, M. Breitbach: Digitale Netze [Digital networks], published by Teubner, ISBN 3-519-06191-0

J. Braband: Safety analysis for a closed transmission system

A. Baginski, M. Müller: Interbus, Grundlagen und Praxis [Interbus, Basics and Practice], published by Hüthig, ISBN 3-7785-2471-2

Gieck: Technische Formelsammlung [Technical formulary], published by Gieck, ISBN 3-920379-21-7

Schröder, Rommel: Elektrische Nachrichtentechnik [Electrical message technology], published by Hüthig, ISBN 3-8101-0045-5

W.-D. Haaß: Handbuch der Kommunikationsnetze [Communication network manual], published by Springer, ISBN 3-540-61837-6

Hofer: Datenfernverbindungen [Remote data connections], published by Springer, ISBN 3-540-08621-8

B. Schürmann: Rechnerverbindungsstrukturen [Computer connection structures], published by Vieweg, ISBN 3-528-05562-6

W. Heinlein: Grundlagen der faseroptischen Übertragungstechnik [Basics of optical fiber transmission technology], published by Teubner,

ISBN 3-519-06117-1

M. Bossert, M. Breitbach: Digitale Netze [Digital networks], published by Teubner, ISBN 3-519-06191-0

R.-H. Schulz: Codierungstheorie [Encoding theory], published by Vieweg, ISBN 3-528-06419-6

P. Gerdson: Digitale Übertragungstechnik [Digital transmission technology], published by Teubner, ISBN 3-519-00093-8

P. Wratil: Speicherprogrammierbare Steuerung in der Automatisierungstechnik

[Programmable logic controllers in automation technology], published by Vogel,

ISBN 3-8023-0235-4

Standards: DIN V VDE 0801, IEC 61508, EN 954-1, IEC 1131-3, etc.

Stan Schneider, Mark Hamilton: RTPS: The Real-Time Publish-Subscribe Network, 155A Moffett Park Drive, Sunnyvale, CA, (www.rti.com)