THE CIP NETWORKS LIBRARY

Volume 5

CIP Safety

Edition 2.17

April 2018

The CIP Networks Library
Volume 5: CIP Safety Specification

Publication Number: PUB00085

ODVA, Inc.
4220 Varsity Drive, Suite A, Ann Arbor, MI 48108-5006 USA
TEL 1-734-975-8840
FAX 1-734-922-0027
EMAIL odva@odva.org
WEB www.odva.org

Certificate

No.: 968/EL 373.04/18

Product tested	CIP Networks Library Volume 5, CIP Safety Edition 2.16	**Certificate holder**	ODVA, Inc. 2370 E. Stadium Blvd. #1000 Ann Arbor, MI 48104 USA
Type designation	CIP Safety on DeviceNet, CIP Safety on EtherNet/IP, CIP Safety on SERCOS		
Codes and standards	IEC 61784-3:2010 ISO 13849-1:2015		IEC 61508 Parts 1-7:2010
Intended application	The CIP Networks Library, Volume 5: CIP Safety Edition 2.16, November 2017 meets the requirements of the IEC 61784-3. It can be used as a safety communication layer (SCL) in applications up to SIL 3 according to IEC 61508 and EN ISO 13849-1 for Category 4 / PL e and enables vendors to build CIP Safety devices for DeviceNet, EtherNet/IP and SERCOS in compliance with these standards.		
Specific requirements	The design, development and suitability of devices for use in safety related applications has to be approved. The network conformance testing has to be performed for individual devices.		

Valid until 2023-02-15

The issue of this certificate is based upon an examination, whose results are documented in Report No. 968/EL 373.04/18 dated 2018-02-15.
This certificate is valid only for products which are identical with the product tested.

Köln, 2018-02-15

TÜV Rheinland Industrie Service GmbH
Bereich Automation
Funktionale Sicherheit
Am Grauen Stein, 51105 Köln
Certification Body Safety & Security for Automation & Grid

Dipl.-Ing. Stephan Häb

www.fs-products.com
www.tuv.com

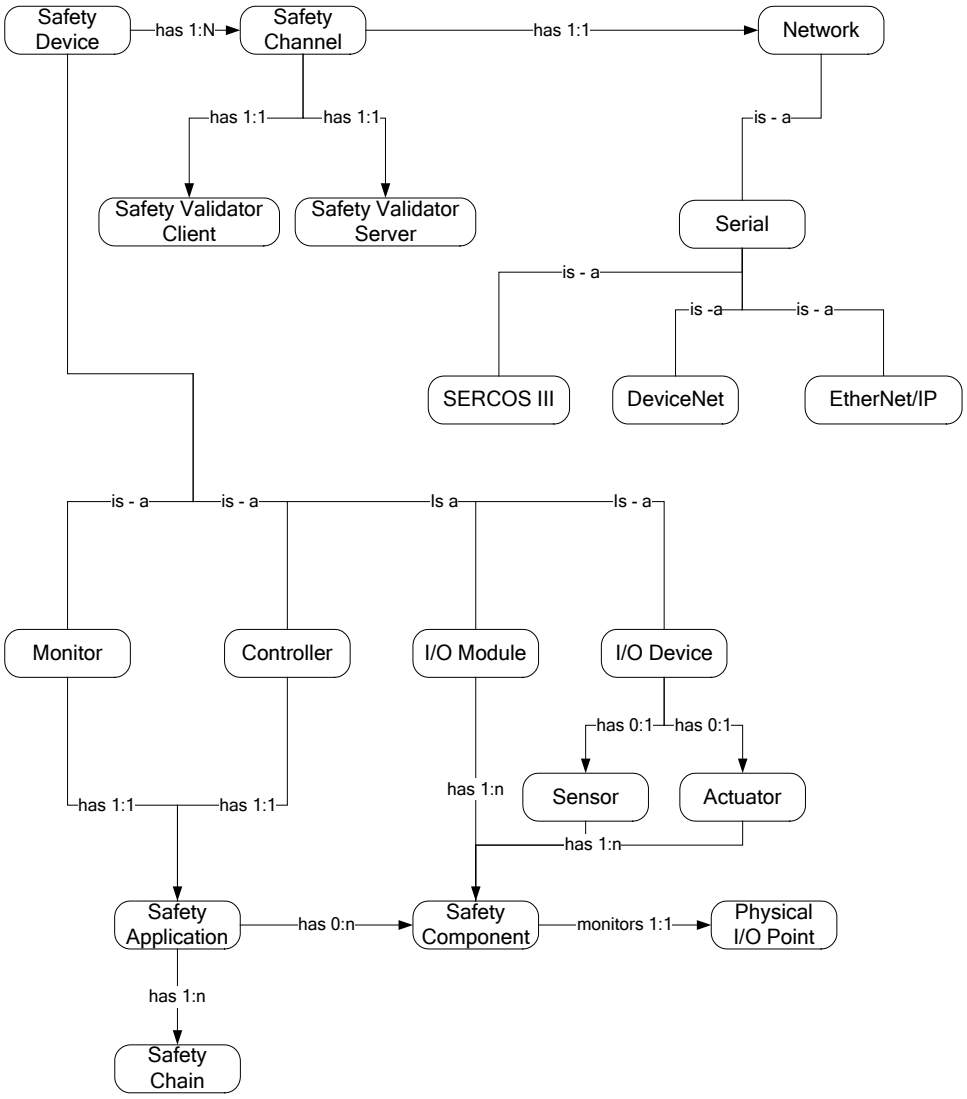
TÜVRheinland®
Precisely Right.

TÜV Rheinland Industrie Service GmbH, Am Grauen Stein, 51105 Köln / Germany
Tel.: +49 221 806-1790, Fax: +49 221 806-1539, E-Mail: industrie-service@de.tuv.com

2-1 Safety Protocol Overview

The safety network is designed as a protocol independent safety layer that resides above the existing network or backplane protocol. The entity-relationship shown in Figure 2-1.1 shows the relationship between possible devices, networks and components.

Figure 2-1.1 Entity Relationship Diagram of Target Safety System



In Figure 2-1.1, safety devices are controllers, monitors, I/O modules or I/O devices. I/O devices may be one of two types, sensor and actuator. Controllers and monitors originate communications with devices. I/O devices, I/O modules, monitors and controllers are all types of devices. Safety devices can have one or more safety communication channels, each of which can be connected to a single network. Each safety channel has a Safety Validator Client and a Safety Validator Server.

The entity relationship diagram in Figure 2-1.1 shows an entire safety application residing in a single controller or monitor. A safety application can contain multiple safety components that, in turn, can contain other safety components. A safety application will contain one or more safety chains. Much like the safety application, the safety chain can contain multiple safety components that, in turn, can contain other safety components. The resulting structure for a safety application is, therefore, a tree of safety components, with each having access the I/O points within the safety system.

2-1.1 Design Approach

The CIP Safety approach uses the safety processes and coding recommendations of the German Safety Bus committee, as documented in the German Safety Bus Committee Specification, Appendix A. that relies on providing measures for possible transmission errors as initially defined in ISO-62280-1. This specification requires a single measure to detect each error condition. The CIP Safety approach exceeds these basic requirements and provides alternate detection measures where possible. Table 2-1.1, is based on the Error and Measures table documented by the German Safety Bus Committee. Table 2-1.1 shows just those measures that the safety protocol uses to detect errors. Also provided in the table are references within this specification where the detection measures are fully described; thus, this table can be used as the central starting point in analyzing the safety protocol.

2-1.2 Communication Errors and Measures to Detect Errors

Table 2-1.1 Measures Against Errors in Messages

Communication Errors	Measures to detect communications errors				
	Time Expectation via time stamp	**Identification for sender and receiver**	**CRC**	**Redundancy with Cross Checking**	**Different data integrity assurance systems for safety and standard messages**
Message Repetition 2-1.2.1	X		X[2]		
Message Loss 2-1.2.2	X		X[2]		
Message Insertion 2-1.2.3	X	X	X[2]		
Incorrect Sequence 2-1.2.4	X		X[2]		
Message Corruption 2-1.2.5			X	X	
Message Delay 2-1.2.6	X				
Coupling of safety and safety information 2-1.2.7		X			
Coupling of safety and standard information 2-1.2.8	X	X	X	X	X
Increased age of data in bridge [1] 2-1.2.9	X				

1. This requirement is not part of the German Safety Bus Committee Specification
2. The Safety CRC provides additional protection for communication errors in fragmented messages.

2-1.2.1 Message Repetition

It is possible for a message to be repeated on a network. This in itself does not represent an error, because the safety protocol allows overwriting of data. However, a repeated message will not have an updated time stamp, because only the base producer can update the time stamp. Repeated messages that were received in place of new data may result in the connection's termination (2-1.3.2) if they did not meet the consumer's time expectation.

2-1.2.2 Message Loss

Time Expectation detects loss of a message. FRS3 The safety protocol requires messages to occur within defined time expectations. Messages received later than these time expectations shall be treated as errors, resulting in the connection's termination.

2-1.2.3 Message Insertion

Message insertion is detected by two measures:

Time Expectation: An inserted message will result in a connection termination because the message received will have an unexpected value in its time-stamp or time-stamp/roll-over count.

An inserted message is detected by identification of sender and receiver. A unique identifier is encoded with the CRC-Sx of the time stamp section, time coordination section, and both of the data sections (2-1.7.1). If the CRC-Sx is incorrect, either the data is corrupted or a message has been sent to the wrong device. See FRS5 (Section 2-1.2.5) to see how this error is handled..

2-1.2.4 Incorrect Sequence

The time expectation detects an incorrect sequence. The incorrect sequence of a message will result in a connection termination because the message received has an unexpected value in its time-stamp and/or roll-over count.

2-1.2.5 Message Corruption

Two additional measures of safety networks exceed the native measures of standard networks to detect message corruption:

Cyclic Redundancy Check: A safety cyclic redundancy code, CRC-Sx (2-1.7.2), is encoded in each safety message.

FRS5 A message corruption shall be detected when the data and CRC-Sx are calculated and compared. This error shall cause the connection to be
Terminated
IF (Base format) OR (Extended Format AND Consumer_Fault_Count) >= Max_Fault_Number
OR Dropped
IF Extended Format AND Consumer_Fault_Count < Max_Fault_Number.

Redundancy with cross checking: All safety data is sent twice (one copy is the ones complement) in the same message packet. The received safety data is crosschecked when it arrives at the consumer. FRS6 A corrupted message that was not detected by the link or CRC-Sx check (i.e., error that exceeded the Hamming distance of CRC) shall be detected when the actual and complemented copies of the data are compared in the consumer. See Section 2-1.8 for details.

2-1.2.6 Message Delay

Time Expectation detects message delay. If an expected message is not received at the consumer during the required time interval, the connection’s periodic timer will expire and the connection shall be terminated (2-1.3.2) See Section 2-7 for a detailed description of safety network times.

2-1.2.7 Coupling of Safety and Safety Information

The coupling of safety messages is detected by the inclusion of a unique identifier, called the PID in the time stamp section, data sections and time correction message Safety CRC calculations. It is called the CID in the time coordination message Safety CRC calculations (2-1.7.1). The PID and CID are established at connection time and is not transmitted with the data. FRS7 The PID or CID is incorporated within the Safety CRC calculation in both the producer and the consumer, thus, if a message is mistakenly received, it shall be detected when the CRC is checked.

2-1.2.8 Coupling of Safety and Standard Information

All five of the safety detection measures are designed to detect the coupling of safety and standard information because a standard message will not use the safety format. FRS8 Safety consumer shall detect standard messages as an incorrectly formed message.

1. Redundancy with cross check: Any standard message that was inadvertently received by a safety consumer will not meet the requirement for redundant data in the same link packet and therefore an error will occur when cross checking is attempted, See Section 2-1.8.
2. Different data integrity assurance systems for safety and standard devices: Safety messages have a unique message encoding (2-1.7), which includes a time stamp and a unique Safety CRC-Sx. A standard device connection will not be able to encode information in this format or generate the correct cyclic redundancy code of a safety message.
3. A standard message will not contain the Safety CRC (as generated by the Safety CRC algorithm) and will not construct the message using the correct safety format, thus a standard message will be detected as an error condition.
4. A standard message will not contain a correct timestamp; this shall cause an error to be detected.
5. A standard message will not contain the correct packet format; this shall cause an error to be detected.

2-1.2.9 Increased Age of Data in Bridges

A time stamp using time expectation is used to detect possible increased age of data in bridges. If an expected message is not received at the consumer by the required time interval, the connection will be terminated (2-1.3.2). See Section (2-1.8.1) for a detail explanation of the time stamp protocol.

2-1.2.10 Addressing errors

This section describes the measures used to detect addressing errors.

2-1.2.10.1 Producer generates an incorrect address, changes header

1. A message is somehow misdirected to an invalid address. This condition detected by: Message Loss (2-1.2.2) in the original consumer.
2. A Message is somehow misdirected to another safety device, This condition is detected by: Message Insertion (2-1.2.3), Repetition (2-1.2.1), Incorrect Time Stamp (2-1.2.4), Message Delay (2-1.2.6), and, Coupling of Safety and Safety Information (2-1.2.7).

2-1.2.10.2 Consumer consumes a wrong address

1. A safety device consumes a message intended for another non-safety related device. This condition is detected by Coupling of Safety and Standard Information (2-1.2.8).

2. A safety device consumes a message intended for another safety device. This condition is detected by: Message Insertion (2-1.2.3), Message Repetition (2-1.2.1), Identification for receiver, and Incorrect Time Stamp (2-1.2.4), and, Coupling of Safety and Safety Information (2-1.2.7).

2-1.2.11 Measures to Protect Standard Devices from Safety Devices

Safety devices are built on the standard network protocol and do not impact standard devices other than using available bandwidth. Improper allocation of bandwidth may cause nuisance trips to the safety system, causing it to go to a safety state. See section (2-1.8.1).

2-1.3 Communication Protocol Behavior

2-1.3.1 Sequence of Safety Checks

FRS9 The following Sequence of checks shall be used to check messages:

- Ping count check
- Evaluate Time stamp section CRC
- Evaluated Time Stamps
- Evaluate Actual Data CRC
- Evaluate Complement Data CRC
- Perform Cross-Check

2-1.3.2 Connection Termination

FRS10 When a producer detects an error that requires connection termination it shall terminate the connection, and notify the application of the action.

FRS11 All consumers shall monitor the periodic transmission of data and go to a safety state if the periodic transmissions cease. If the consumer detects an error, it must go to a safety state and terminate the consuming connection associated with that error.

FRS12 When the Safety Layer detects an error requiring connection termination the termination shall be implemented using the following sequence.

- The safety layer detects an error requiring termination
- The safety layer notifies the application program by setting the connection status to indicate a safety communications fault
- The safety application shall transition data and I/O (e.g. set outputs to the safety state) associated with the connection to a safety state
- The safety layer shall notify the underlying communications system of an error and request the termination of the connection by setting its status to indicate a safety communications error
- The safety layer shall not transition from its safety state until the fault is cleared and a connection restart sequence is initiated.

2-1.3.3 Cross Checking Error

FRS13 When crosschecked safety data are found to be different, the data is treated as faulty. The base format consumer shall terminate the connection (See Section 2-1.3.2) and the Extended Format consumer will increment the Consumer_Fault_Count and drop the packet if the count is less than the Max_Fault_Number, or terminate the connection if is equal to or greater than the Max_Fault_Number.

2-2.1 High Level View of a Safety Device

Figure 2-2.1 shows a number of possible safety architectures which could use CIP Safety. It shows that any SIL level from 1-3 is possible for using CIP Safety. However, it also shows that in all cases, the CIP Safety runtime stack and CIP Safety configuration are developed with an SC3 development process. The assessment of the systematic capability, SIL level requirements, and appropriate non-interference measures is carried out by the safety certification body.

Figure 2-2.1 Possible Safety Architectures for CIP Safety

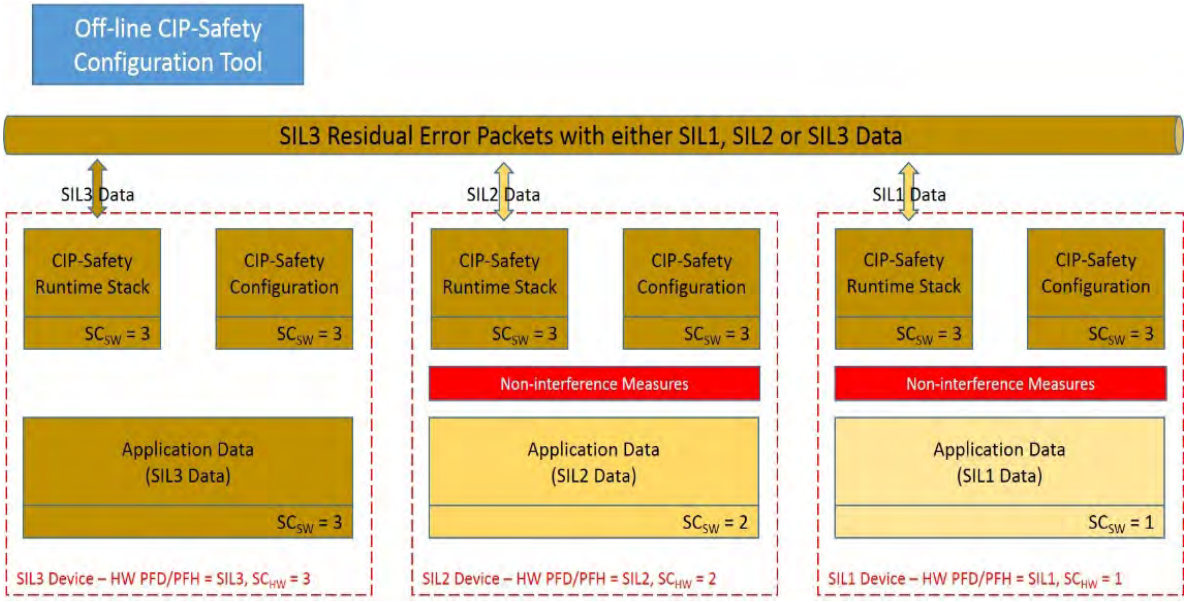
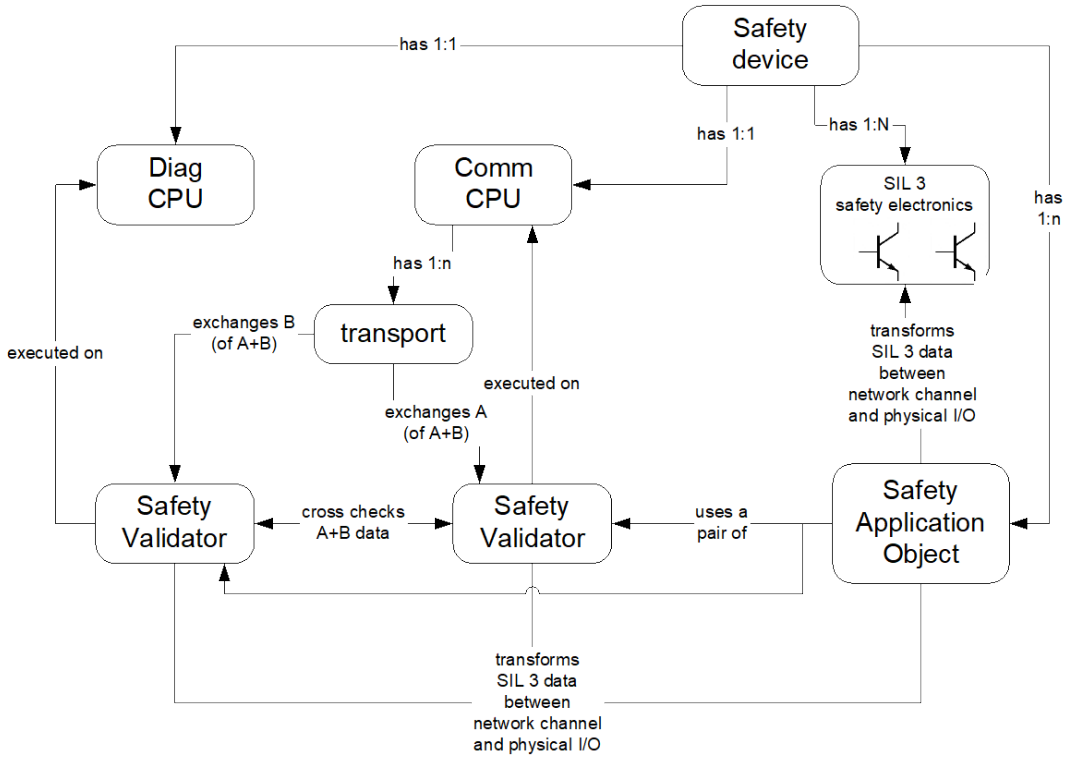


Figure 2-2.2 shows an example model of a SIL3 safety device containing one or more safety application objects. Each safety application objects functions with a pair of Safety Validator objects to exchange safety data between the CIP transport layer and the safety I/O electronic circuitry. These electronics are designed to provide a level of error detection and fault isolation suitable for SIL 3 applications. Each of the two Safety Validator objects is associated with a network connection on networks such as DeviceNet or Ethernet/IP. This example model assumes a 1oo2 Safety Topology in which two independent CPUs participate in the safety protocol. However with different hardware architectures, CIP Safety technology may be suitable for use in safety applications less than SIL3.

All implementations of CIP Safety technology shall use a safety certifying agency to ensure that the design and implementation of the CIP Safety protocol (safety-related communication software) provides a Systematic Capability of SC3 according to IEC 61508.

The CIP Safety protocol must be considered as part of a complete device, and the integration of the CIP Safety protocol into the device must be done to achieve/maintain SC3. For example, a Vendor cannot use a separately certified CIP Safety stack (one that does provide SIL3 and SC3) in their product without regard to the need to provide SC3 for the integration of the safety communication software in the integrated product.

Figure 2-2.2 Safety Device Reference Model Entity Relation Diagram



2-2.2 Safety Validator Object

The Chapter 5-5 of this volume defines a Safety Validator object from which both the SafetyValidatorServer and SafetyValidatorClient functions are implemented.

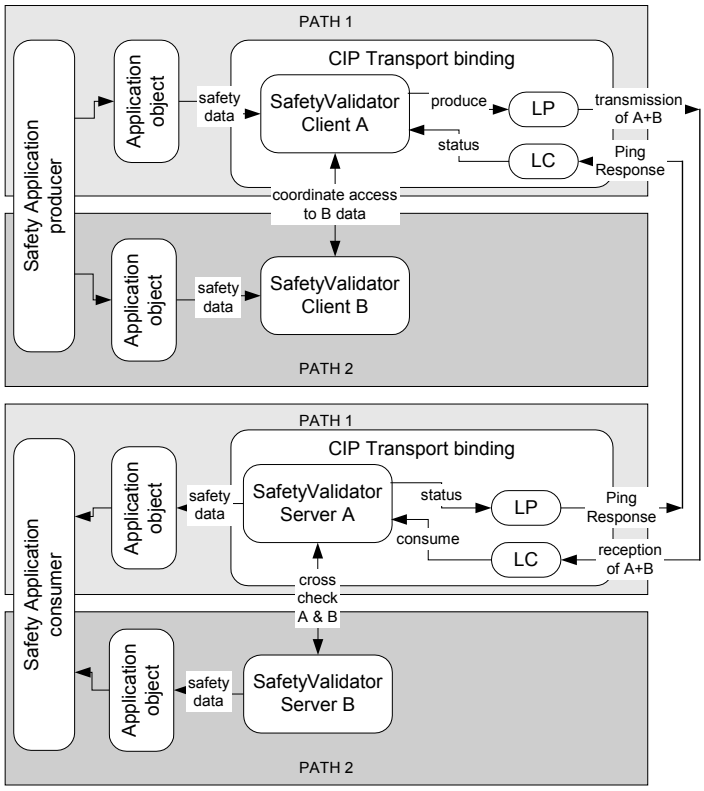
The next section describes the general principles and common attributes of these two safety layer functions.

2-2.3 Relationship between SafetyValidatorServer and SafetyValidatorClient

Figure 2-2.2 shows a high level view of two devices interchanging data via a SafetyValidatorClient and a SafetyValidatorServer. A safety producing application uses a SafetyValidatorClient to send safety data to a safety consuming application that uses a SafetyValidatorServer.

Note that only one of the redundant paths interfaces to the CIP transport layer. The A+B (actual and complement) data is encapsulated in a single transport frame and only one processor is involved in the reception and transmission of the data frames

Figure 2-2.3 Two Devices Interchanging Data via a SafetyValidatorClient and a SafetyValidatorServer



2-2.4 Extended Format Time Stamp Rollover Handling

The Extended format requires coordinated synchronization between producers and consumers on connection establishment along with maintenance operations to detect and handle Time Stamp rollover events. The following sections graphically show these operations for each unique case. The example code shows these operations in their proper place in the code as well.

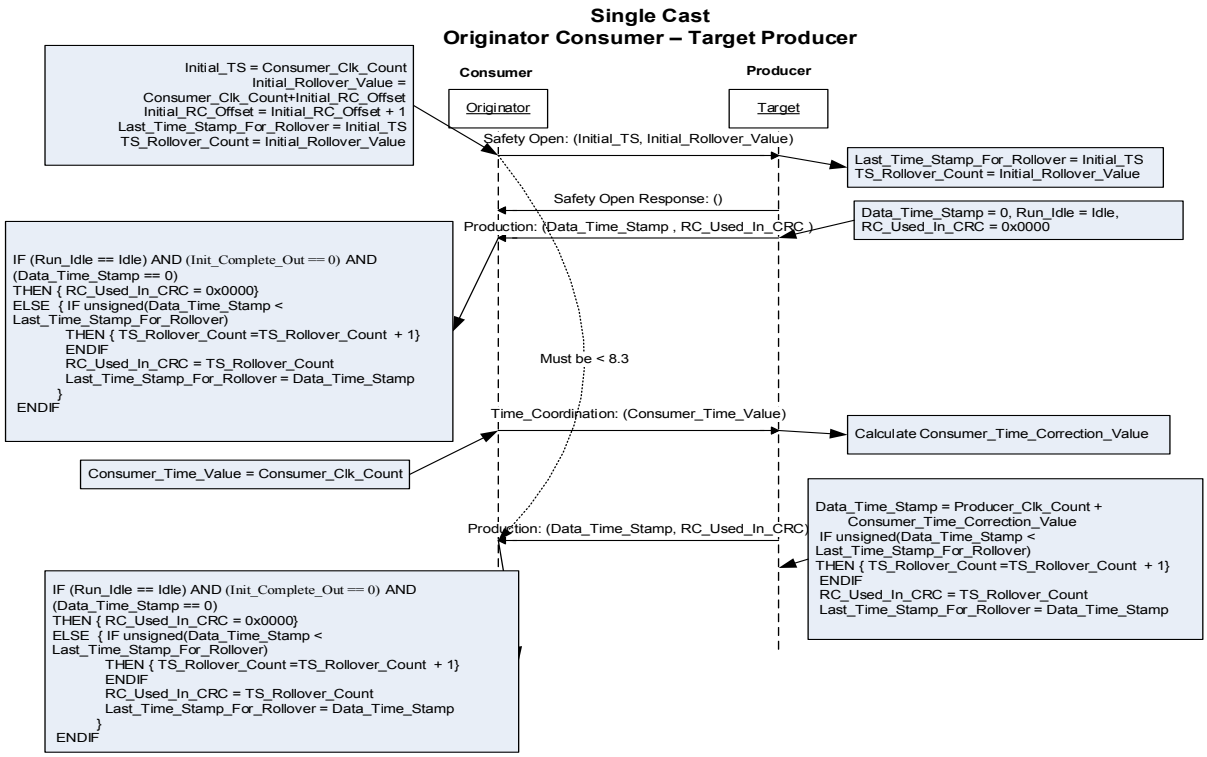
2-2.4.1 Single-Cast, Originator Consumer, Target Producer

When single-cast connections are originated by the consumer (i.e. single cast inputs), the originator has to provide initialization values in the SafetyOpen. The target uses these values to initialize production parameters.

FRS376 The rollover count used in the Extended Format Single-Cast CRC shall be zero until the initial Time Coordination exchange has been completed.

The complete process is shown below in Figure 2-2.3

Figure 2-2.4 Single-Cast, Originating Consumer Target Producer



The time between the generation of the Safety Open and the Consumers processing of the first packet produced after the producers reception of the Time_Coordination message must be < 8.3 seconds.

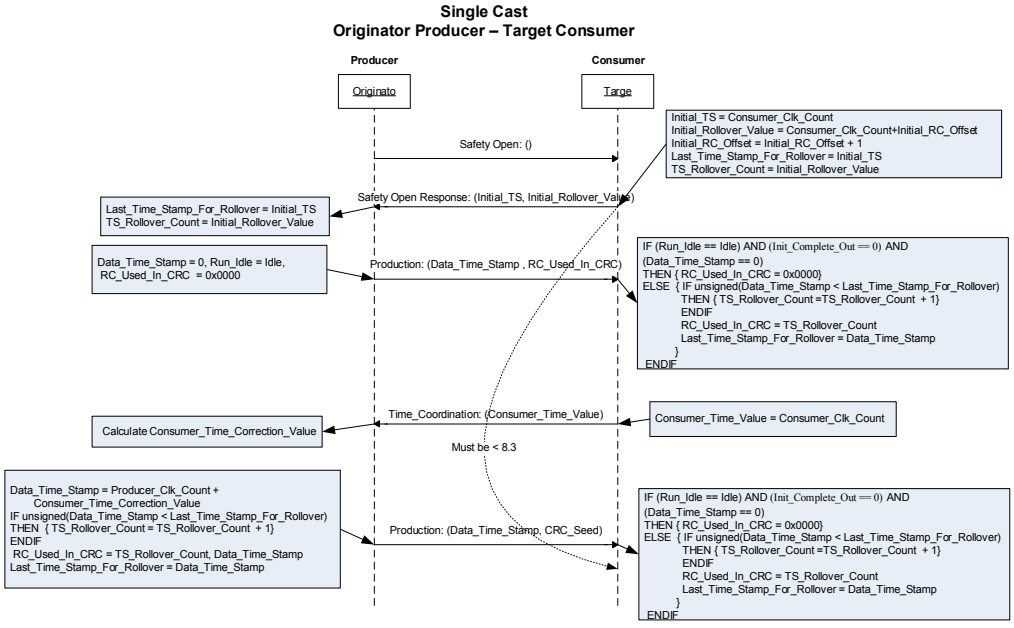
2-2.4.2 Single-Cast, Originator Producer, Target Consumer

FRS379 When Extended Format single-cast connections are originated by the producer (i.e. outputs), the target consumer shall provide initialization values in the SafetyOpen Response.

The originator producer uses these values to initialize production parameters.

The complete process is shown below in Figure 2-2.4.

Figure 2-2.5 - Single-Cast, Originator Producer, Target Consumer



The time between the generation of the Safety Open Response and the Consumers processing of the first packet produced after the producers reception of the Time_Coordination message must be < 8.3 seconds.

2-2.4.3 Multi-cast, Originator Consumer, Target Producer

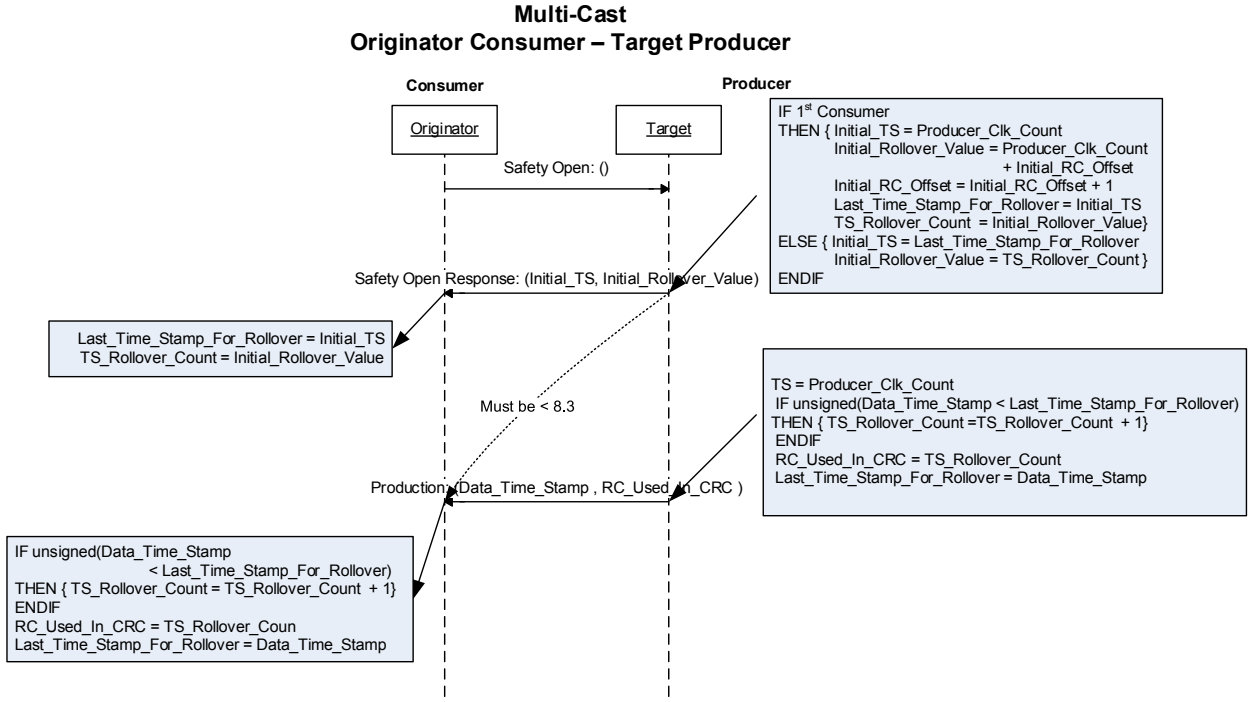
FRS377 When Extended Format multi-cast connections are originated by the consumer (i.e. multi-cast inputs); the target producer shall provide initialization values in the SafetyOpen Response.

The target consumer uses these values to initialize production parameters. The values the target producer sends in the SafetyOpen Response differs depending on whether this is the 1st consumer to request a connection or not.

FRS378 The active seeding of the CRC with the rollover count in Extended Format Multi-cast messages shall begin immediately on first production.

The complete process is shown below in Figure 2-2.5

Figure 2-2.6 - Multi-Cast, Originator Consumer, Target Producer



The time between the generation of the Safety Open Response and the Consumers processing of the first packet must be < 8.3 seconds.

2-2.5 SafetyValidatorClient Function definition

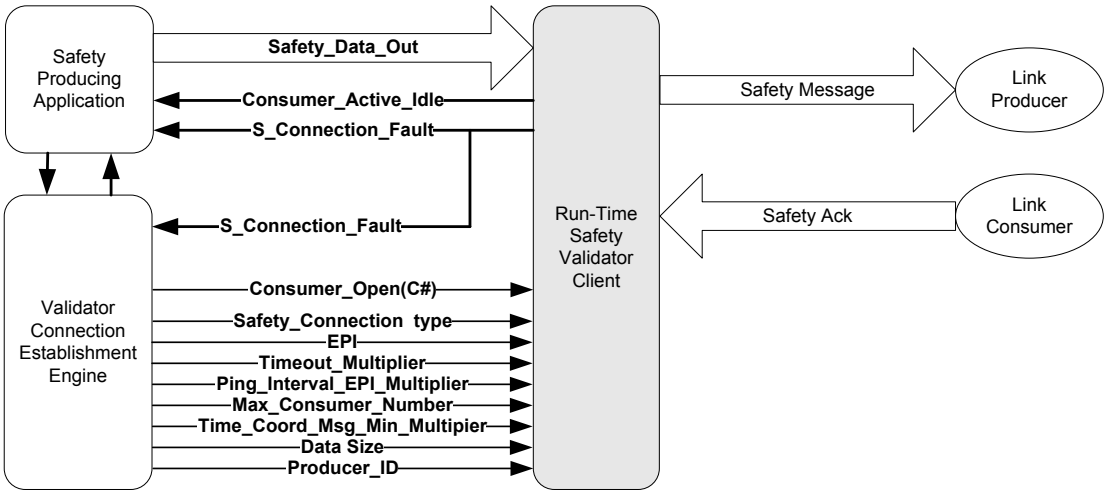
The SafetyValidatorClient is the embodiment of the safety layer that augments a standard CIP connection to transmit safety data. It coordinates the production of application data with a peer instance of the SafetyValidatorServer.

2-2.5.1 Safety Production

This section describes the production of data and the interfaces between the application, the SafetyValidatorClient and the underlying link layer. Note that the focus is on the SafetyValidatorClient run-time behavior and its interfaces. The application itself must be of high integrity to meet the target SIL level and this can be accomplished in many ways and is ultimately a vendor design decision; however, elsewhere in this document, a reference model is presented of an architecture that has been qualified for SIL3 requirements in specific instances. This reference model is for informative purposes only.

FRS123 Safety_Data_Out is produced at a periodic rate, referred to as the Expected Packet Interval (EPI). The SafetyValidatorClient shall sample, capture, and time stamp the data to be sent every EPI time period.

Figure 2-2.7 Safety Production Data Flow



2-2.5.1.1 Producing Application Interface

FRS124 The producing application provides Safety_Data_Out to the SafetyValidatorClient. The SafetyValidatorClient shall build the Mode_Byte, Actual_Data, and Complement_Data, and Time stamp section. (see section 2-1.7.1 for formats)

FRS125 The SafetyValidatorClient shall provide safety connection status back to the producing application for each consumer.

2-2.5.1.1.1 Safety Data Production Logic

This section describes the logic that shall be followed by all safety data producers. The actual implementation of the logic may vary, but equivalent results shall be obtained.
The logic in this document assumes an asynchronous producing application that makes the safety data available to the SafetyValidatorClient.

Definitions for the variables used in the safety data production logic can be found in section 2-4.5.

2-2.5.1.1.1.1 Example Safety Data Production Cold Start Logic

The production logic to initiate a cold start of a connection is:

```
////////////////////////////////////////////////////////////////
// Cold start after connection establishment processing
////////////////////////////////////////////////////////////////
// This Logic should be executed at the transition of the
//  producing connection from closed to open.
// For Single-Cast connections this logic may
//  be performed any time the connection is Opened or Re-Opened.
////////////////////////////////////////////////////////////////
```

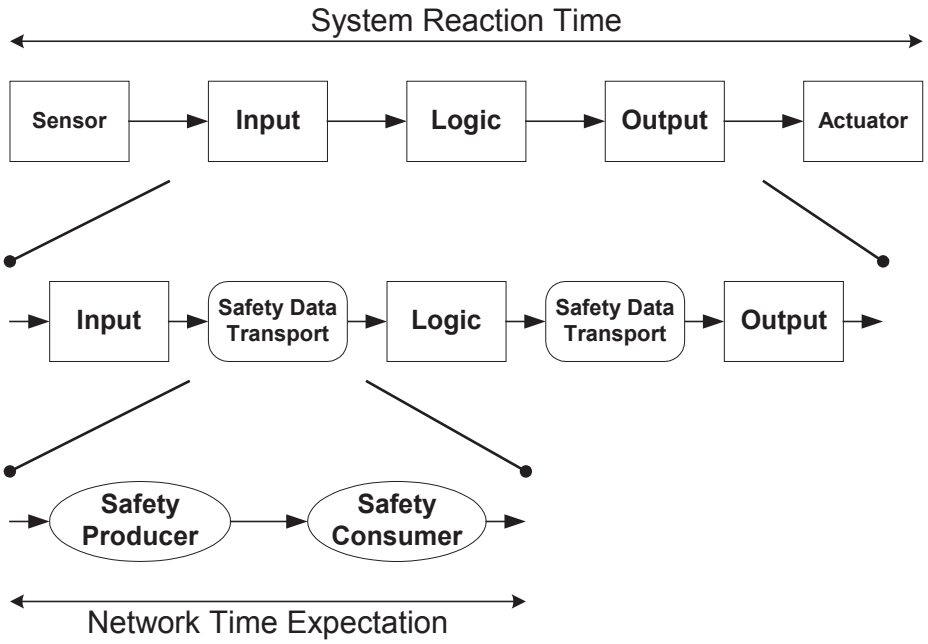
```
Ping_Interval_EPI_Count = 0,
RR_Con_Num_Index_Pntr = Max_Consumer_Number,

// Initialize the ping count in the safety message to 0
Mode_Byte.Ping_Count = 0,

// Time_Drift_Per_Ping_Interval, the minimum value is 1
Time_Drift_Per_Ping_Interval =
        Roundup(EPI * Ping_Interval_EPI_Multiplier / 320000),

FOR (Consumer_Num = 1 to Max_Consumer_Number),
{
    // Producer Dynamic Variables
    Consumer_Active_Idle[Consumer_Num-1] = Idle,
    S_Connection_Fault[Consumer_Num-1] = OK,
    Producer_Rcved_Time_Value[Consumer_Num-1] = 0x0000,
    Consumer_Time_Correction_Value[Consumer_Num-1] = 0x0000,
    Ping_Int_Since_Last_Time_Coord_Msg_Count[Consumer_Num-1] = 0x0000,
    Consumer_Time_Value[Consumer_Num-1] = 0x0000,
    Producer_Fault_Counter[Consumer_Num-1] = 0,  //For ExtendedFormat only

    // Producer Derived Variables

    // Time_Drift_Constant, the minimum value is 1. (Time_Drift_Constant is not
    // saved but is used in the calculation of the
    // Connection_Correction_Constant below.)
    Time_Drift_Constant =
                    Roundup((Timeout_Multiplier.PI [Consumer_Num-1] +1) * EPI *
                        Ping_Interval_EPI_Multiplier / 320000),

    // Connection_Correction_Constant
    Connection_Correction_Constant[Consumer_Num-1] =
Time_Drift_Constant + 1 - Time_Coord_Msg_Min_Multiplier [Consumer_Num-1],

    // Time_Coord_Response_EPI_Limit
    Time_Coord_Response_EPI_Limit[Consumer_Num-1] = Roundup((5000000 +
                    (Time_Coord_Msg_Min_Multiplier[Consumer_Num-1]*128) +
                    (EPI * ( Consumer_Num – 1))) / EPI),

    // Time_Coord_Response_EPI_Limit has a maximum value of 1000
    IF (Time_Coord_Response_EPI_Limit[Consumer_Num-1] > 1000),
    THEN
    {
            Time_Coord_Response_EPI_Limit[Consumer_Num-1] = 1000,
    }
    ENDIF
}
ENDFOR
IF (ExtendedFormat),
THEN
{
      RC_Used_in_CRC  = 0x0000
      IF (Multi-Cast),
      THEN
      {
```

2-7 System Reaction Time

2-7.1 Introduction

The system reaction time is the worst-case time from a safety related event as input to the system or as a fault within the system, until the time that the system is in the safety state.

Figure 2-7.1 System Reaction Time



To determine the system reaction time of any control chain the user shall add up the components of the safety chain.

For example, the system reaction time of the example above would be:

System reaction time = Sensor reaction time
+ Input reaction time
+ Network reaction time
+ Controller reaction time
+ Network reaction time
+ Output reaction time
+ Actuator reaction time

2-7.2 Network Time Expectation

The Network Time Expectation is a portion of the System Reaction Time. The Network Time Expectation is the worst case time, from the time the data is captured by the safety data producer, until the consuming application recognizes a safety state. This also includes errors during production and consumption.

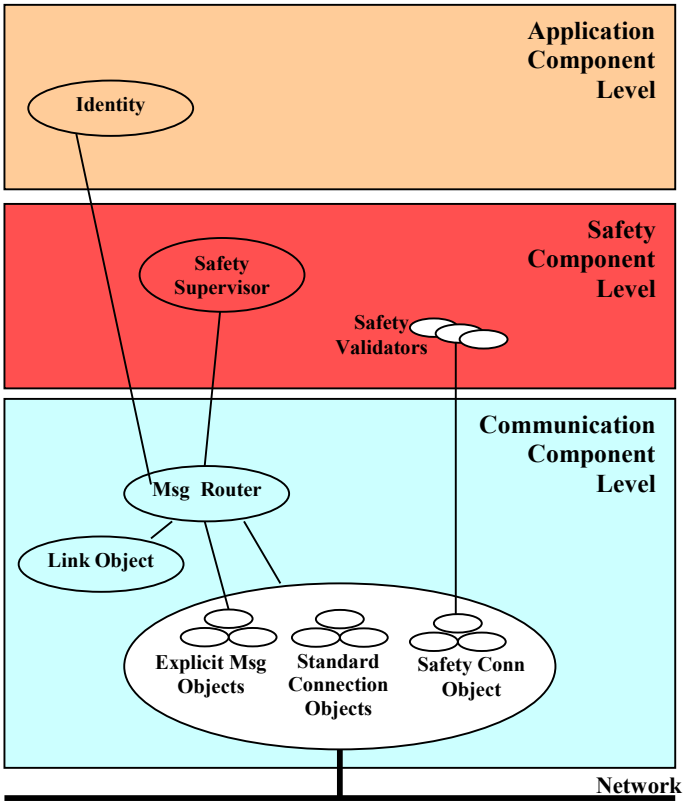
The Network_Time_Expectation_Multiplier used in the previous sections of this document, represents the worst case measured time from the time the data is captured by the safety data producer until the time that the consuming application recognizes a safety state indication.

6-3 Baseline Safety Device

All Safety Devices shall contain, at a minimum, a basic level of safety functionality. Safety devices are built off of this baseline. As shown in Figure 6-3.1, the baseline safety functionality is defined as one which contains a "baseline" Safety Supervisor (refer to Section 5-4) for high-integrity device control and one or more Safety Validator instances for high-integrity safety I/O connections.

It is optional for safety profiles to support the SNCT implementation of the Safety Supervisor, but is highly recommended since the profile then includes a common, TUV-certified configuration solution. All safety profile definitions shall specify which level (i.e. Baseline or SNCT) of Safety Supervisor is required.

Figure 6-3.1 Baseline Safety Device



6-4 Rules for Safety Profiles

There are three cases for Safety Profiles: those here in chapter 6 whose names begin with the word Safety, vendor specific Safety Profiles whose names begin with the word Safety, and non-Safety device profiles, such as PLC (type 0x0E).

Safety objects can be added to an existing standard profile and use that existing profile device type. Safety objects shall only be deployed in devices which meet all the requirements of CIP Safety in this volume, including sections 6-1, 6-2, 6-3 and 6-5. Devices which use a non-safety device type that have safety I/O assembly object instances shall use vendor specific Instance IDs for those assembly instances.

All vendor specific safety device type profiles and safety device type profiles created in this chapter shall have the word “Safety” as the first word in the name.

6-5 Safety Manual Requirements

This section will contain some of the specific requirements that must be contained in the user safety manual for devices which use the CIP Safety Network Protocol. These requirements are derived from the safety case where the user is required to perform some action to insure a safe process.

Table 6-5.1 Safety Manual Requirements

User Manual Requirements
SRS50 The safety manual shall contain a user instruction requiring the user to completely test a device’s operation before setting the Lock Attribute
SRS51 The safety manual shall contain a user instruction requiring the user to upload and compare the configuration from each affected safety devices to that which was sent by the SNCT before setting the Lock Attribute in those devices.
SRS52 The safety manual shall contain a user instruction requiring the user to clear any pre-existing configuration from any safety device before installing it onto a safety network.
SRS53 The safety manual shall contain a user instruction requiring the user to commission all safety devices with MacId (and Baud Rate if necessary) prior to installing it onto a safety network
SRS54 The Safety Manual shall include instructions for safety function implementers to carefully consider implications of mixing different SIL level devices on the network.
SRS193 The Safety manual shall contain a user warning advising that originators that have an “automatic” SNN setting feature should only use that feature when the safety system is not being relied upon.

6-6 Safety Discrete I/O Device

Device Type: 23 Hex

A Safety Discrete I/O Device type interfaces to multiple Safety I/O device types that do not have network capabilities. Examples include Safety sensors and actuators

6-6.1 Object Model

The Object Model in Figure 6-6.1 represents a Safety Discrete I/O Device. The Table below includes:

- the object classes present in this device
- whether or not the class is required
- the number of instances present in each class

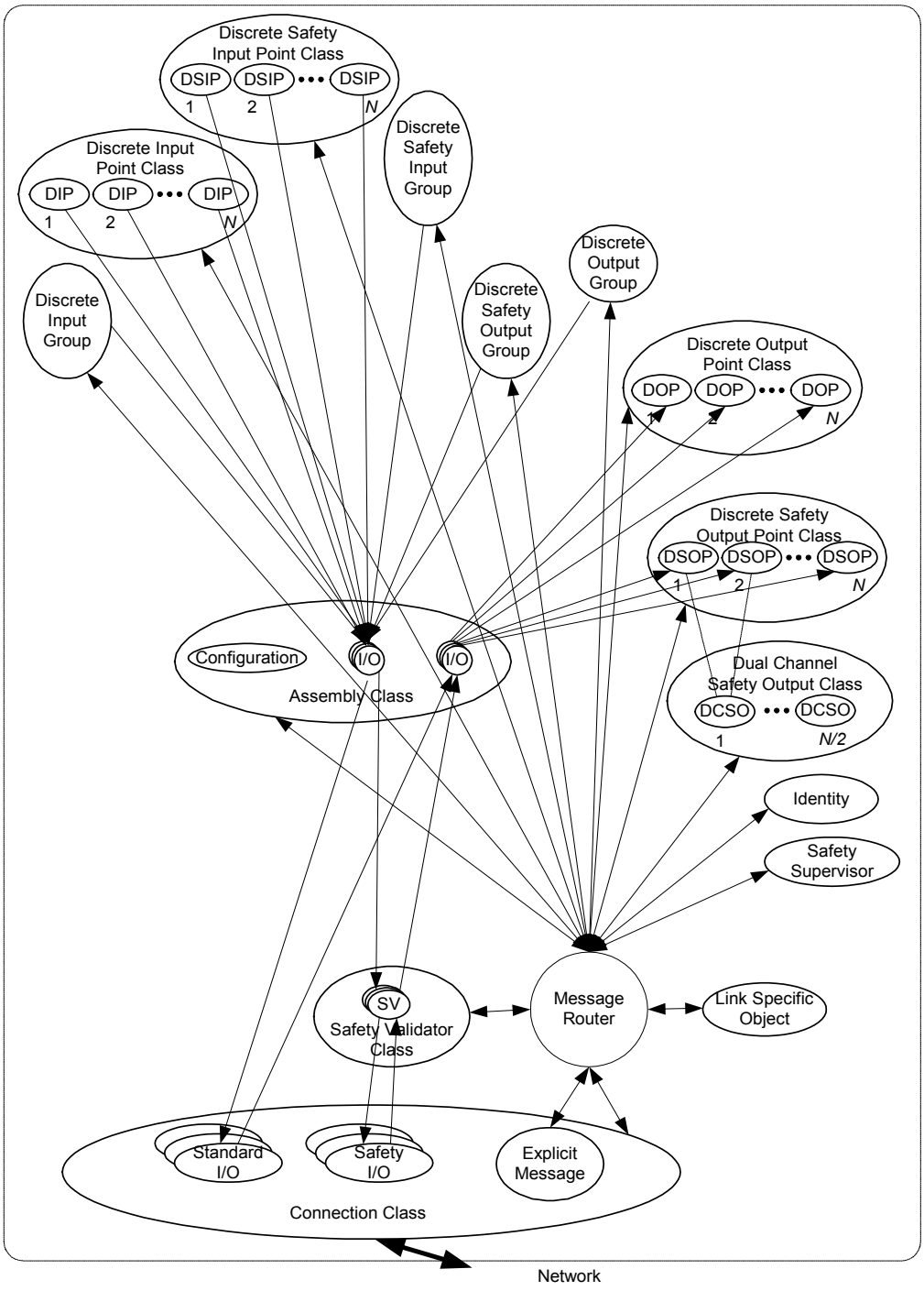
Table 6-6.1 Objects Present in a Safety Discrete I/O Device

Object Class	Option/Required	# of Instances
Identity	Required	1
Message Router	Required	1
Link Specific Object (s)	Required	1
DeviceNet	Required for DeviceNet Safety	1
Connection Control	Required	#
Connection	Required for DeviceNet Safety	1
Safety Validator	Required	#
Safety Supervisor	Required	1
Assembly	Required	*
Discrete Input Point (DIP)	**	*
Safety Discrete Input Point (SDIP)	****	*
Discrete Output Point (DOP)	***	*
Safety Discrete Output Point (SDOP)	*****	*
Safety Dual Channel Output (SDCO)	##	*
Discrete Input Group (DIG)	Optional	1
Safety Discrete Input Group (SDIG)	Optional	1
Discrete Output Group (DOG)	Optional	1
Safety Discrete Output Group (SDOG)	Optional	1

Table Footnotes:

* = # of instances depends on the level of I/O support provided by the product.
** = Optional for Standard Input Functions (provides backward compatibility)
*** = Optional for Standard Output Functions
**** = Required for Safety Input Functions
***** = Required for Safety Output Functions
\# = Depends on the level of communications support provided by the product.
= Required for Safety Output Dual Channel Functions

Figure 6-6.1 Object Model for Safety Discrete I/O Device



6-6.2 How Objects Affect Behavior

The objects for this device affect the device's behavior as shown in the table below.

Table 6-6.2 Object Effect on Behavior

Object	Effect on Behavior
Identity	No effect
Message Router	No effect
Link Specific Object	Configures communication link attributes
Connection Control Object	Contains the logical ports into and out of the device
Assembly	Defined I/O data format and configuration data format
Safety Supervisor	Implements the Safety Network Configuration Tool Interface
Safety Validator	Handles safety protocol
Discrete Input Point (DIP)	Defines behavior of the discrete standard input points for this device
Safety Discrete Input Point (SDIP)	Defines behavior of the discrete safety input points for this device
Discrete Output Point (DOP)	Defines behavior of the discrete standard output points for this device
Safety Discrete Output Point (SDOP)	Defines behavior of the discrete safety output points for this device
Safety Dual Channel Output	Defines behavior of the dual channel discrete safety outputs for this device
Discrete Input Group	Stores the combined status of the Discrete Input Points
Safety Discrete Input Group	Stores the combined status of the Discrete Safety Input Points
Discrete Output Group	Stores the combined status of the Discrete Output Points
Safety Discrete Output Group	Stores the combined status of the Discrete Safety Output Points

6-6.2.1 Safety Supervisor Requirements

The safety supervisor definition contains a number of "profile dependent" functions. This section defines what the required behavior shall be for the Safety Discrete I/O device profile.

Table 6-6.3 Safety Supervisor Implementation Level

Implementation Level	Profile Requirement
Baseline functionality	Required
SNCT functionality	Optional

Table 6-6.4 Safety Supervisor Profile-dependent State Event Behavior

Event	"IDLE" State Behavior	"Executing" State Behavior	Comments
Safety Connection Failed/Closed	Remain in IDLE	If any standard or safety I/O connection still open, remain in EXECUTING, Else, Transition to IDLE	In this profile, device is in IDLE unless at least one standard or Safety I/O connection is established
Standard or Safety I/O Connection established	Transition to EXECUTING	Remain in EXECUTING state	
Type 1 SafetyOpen	Configure device, transition to EXECUTING	Drop standard and safety I/O connections, configure device, return to EXECUTING	
Safety I/O Connection Deleted	Not supported	Not supported	Connection deletion not supported
Mode Change	Error response: "Service Not Supported"	Error response: "Service Not Supported"	No mode change defined for this profile

6-6.3 Defining Object Interfaces

The objects in this device have the interfaces listed in the following table.

Table 6-6.5 Object Interfaces

Object	Interface
Identity	Message Router
Message Router	Explicit Messaging Connection Instance
DeviceNet	Message Router
Connection	Message Router
Assembly	I/O Connection (for standard connections) or Safety Validator (for Safety I/O connections) or Message Router
Safety Supervisor	Message Router
Safety Validator	Message Router or Safety I/O Connection
Discrete Input Point (DIP)	Message Router
Safety Discrete Input Point (SDIP)	Message Router or Assembly Object
Discrete Output Point (DOP)	Message Router or Assembly Object
Safety Discrete Output Point (SDOP)	Message Router or Assembly Object
Safety Dual Channel Output	Message Router or Assembly Object
Discrete Input Group	Message Router
Safety Discrete Input Group	Message Router
Discrete Output Group	Message Router
Safety Discrete Output Group	Message Router

6-6.4 Relationship between DIP and SDIP

A module using the Safety Discrete I/O profile may support either the SDIP, DIP, or both. An instance of the SDIP can be used to access inputs via safety evaluated data, or to access inputs as standard inputs without safety evaluation, or to access both. An Instance of the DIP can be used as standard discrete input data which is equivalent to the SDIP standard input data without safety evaluation.

6-6.5 Relationship between DOP and SDOP

A module using the Safety Discrete I/O profile may support either the SDOP, DOP, or both. The DOP may be used for standard Output functions or for Test Output functions.

6-6.6 I/O Assembly Instances

The I/O Assemblies for the Safety Discrete I/O Device may be classified into the following types.

Table 6-6.6 I/O Assembly Object Instances for the Safety Discrete I/O Device

Input or Output	Data Type
Input	Standard Data Only
Input	Safety Data Only
Input	Safety and Standard Data
Output	Standard Data Only
Output	Safety Data Only
Output	Safety and Standard Data

The assembly instance definition will differentiate safety data from standard data.

When an input assembly can be accessed by both a Safety I/O connection and/or a Standard I/O connection, the same instance number will be used by either I/O connection type.

When an output assembly can be accessed by a Safety I/O connection or a Standard I/O connection, the same instance number will be used by either of the I/O connection types. Only one I/O connection can be made to an output assembly at any given time.

Where ever possible the Standard Data only assemblies that are mapped to the DIP, and DOP objects will use the definition in the General Purpose Discrete I/O Profile (Device Type = 07_{hex}). In addition this profile will define Standard Data only assemblies that are mapped to the SDIP that provide comparable functionality, but don't require the support of or the additional configuration of the DIP. These assemblies will be defined in the 180-1FFhex instance range.

The use of reserved bits previously allowed within I/O assemblies is deprecated as of this publication and can no longer be used for any other purpose

As denoted in the table below instance number ranges 01-63hex and 100-$17F_{hex}$ of this profile will refer to the General Purpose Discrete I/O Profile. Those assembly definitions and mapping will not be repeated in this profile.

Volume 5: CIP Safety

Appendix F: Safety Test Plan

Contents

Required Initial Conditions:

1. Run SPTE as an originator of EF connections. Both DUT producing and consuming connections must be tested.
2. Test Procedure:
3. Establish a single-cast output connection (DUT is consumer) with the EF segment type and a SCID equal to the DUT SCID
4. Confirm a positive SafetyOpen response is received in the base response format
5. Test will inspect the SafetyOpen Response and confirm the parameters are correct
6. Confirm the O-to-T API, T-to-O API and Time Correction API all match what was sent in the SafetyOpen
7. Establish a Multi-cast input connection (DUT is producer) with the EF and a SCID equal to the DUT SCID
8. Confirm a positive SafetyOpen response is received with proper EF response format
9. Test will inspect the SafetyOpen Response and confirm the parameters are correct
10. Confirm the O-to-T API, T-to-O API and Time Correction API all match what was sent in the SafetyOpen
11. Test will inspect the SafetyOpen Response and confirm the parameters are correct
12. Close connection

F-3.4.1.3 TST3 – Connection Initialization

This test will support both the base format and Extended Format safety connection.

Requirement Number	Requirement
FRS280	After the connection is first established, the consuming application shall close base format connections if initialization is not completed within 10 seconds. The consuming application shall close Extended connections if initialization is not completed within 8.3 seconds.
FRS281	For Single-Cast, the flag Init_Complete_Out shall indicate that the first time coordination message has been received by the producer and the producer is producing data with the time stamp relative to the consumer's clock.
FRS282	For Multi-Cast, the flag Init_Complete_Out shall indicate that the first time coordination message has been received by the producer for this consumer and the producer has received the 1st time correction message based on the time coordination information.
SRS57	Safety Nodes which reside on DeviceNet shall implement the SafetyOpen and SafetyClose services of the Connection Object.

TST3 The connection establishment test shall confirm that the Consumer will close its connection if initialization isn't completed within the required time.

Required Initial Conditions:

1. This test will support both the base format and Extended Format safety connection.
2. Run SPTE as an originator.

Test Procedure:

1. If the DUT Supports Multi-cast consumer,
2. Configure the DUT so it originates a multi-cast connection request
3. Confirm the service code used 0x54
4. Reply appropriate for the connection to be established
5. Begin producing data and generate a ping count change.

F-5 Safety Test Matrix

This section contains a matrix that cross references the tests defined in the Safety Test Guide to the behaviors that may be implemented in a safety device. Use the behaviors at the top of the matrix to determine which tests must be applied to the safety device. If there is a discrepancy between the Safety Test Matrix and the specification in the main sections of this volume then the specification shall take precedence.

F-5.1 Safety Test Matrix

refer to footnotes at bottom of table								Single-cast Producing Target	Multicast Producing Target	Single-cast Consuming Target	Multicast Consuming Originator	Single-cast Consuming Originator	Single-cast Producing Originator	Bridge Interface
Test Names	Test Number	DUT Type	Base Format Test	Extended Format Test	Dnet Device	Enet Device	SERCOS III Device							
Standard DeviceNet Conformance	1				X									
Standard EtherNet/IP Conformance	107					X								
Standard SERCOS III Conformance	129						X							
Type 2 Connection Est. Pos. Test	2	Target	X		X	X	X	RO	RO	RO				
Extended Format Type 2 Connection Est. Pos. Test	113	Target		X	X	X	X	RO	RO	RO				
Connection Initialization Test	3	Target	X	X	X	X	X	ROPFS	ROPFS	ROPFS				
Connection Parameters CRC Negative Test	4	Target	X	X	X	X	X	ROPFS	ROPFS	ROPFS				
Type 2 SCID Check	5	Target	X	X	X	X	X	ROPFS	ROPFS	ROPFS				
Electronic Key Mismatch	6	Target	X	X	X	X	X	ROPFS	ROPFS	ROPFS				
Target Connection ID Allocation	7	Target	X	X	X			ROPFS	ROPFS	ROPFS				
Multi-cast Producer, Consumer Number Allocation	8	Target	X	X	X	X	X		ROPFS					
Producer Multi-Cast Time Correction	99	Target	X		X				X					

refer to footnotes at bottom of table								Single-cast Producing Target	Multicast Producing Target	Single-cast Consuming Target	Multicast Consuming Originator	Single-cast Consuming Originator	Single-cast Producing Originator	Bridge Interface
Test Names	Test Number	DUT Type	Base Format Test	Extended Format Test	Dnet Device	Enet Device	SERCOS III Device							
DeviceNet Extended Format Multi-cast producer formats, seeding & connections	120	Target		X	X				X					
Type 2 Single-Cast Connection Generation Test on DeviceNet	9	Originator	X	X	X							ROPFS	ROPFS	
Type 2 Single-Cast Connection Generation Test on EtherNet/IP	118	Originator	X	X		X	X					ROPFS	ROPFS	
Type 2 Multi-cast Connection Generation on DeviceNet	10	Originator	X		X						ROPFS			
Type 2 Multi-cast Connection Generation on EtherNet/IP	119	Originator	X	X		X	X				ROPFS			
Electronic Key Generation Test	100	Originator	X	X	X	X	X				ROPFS	ROPFS	ROPFS	
SafetyClose Processing by Targets	101	Target	X	X	X	X	X	ROPFS	ROPFS	ROPFS				
Connection Enable/SafetyClose	105	Originator	N/A	N/A	X	X	X				RO, C6	RO, C6	RO, C6	
Producer CRC & PID/CID Test	13	Producer	X	X	X	X	X	ROPFS	ROPFS				ROPFS	
Orig Producer Packet Generation - 1 to 2 Bytes Data	14	Producer	X		X	X	X	C12	C12				C12	