CIP Safety: Safety Networking for the Future

David A. Vasko, Suresh R. Nair, Open DeviceNet Vendors Association

Safety networks have emerged, which allow control system developers to replace hardwired safety chains with communication networks. Typically using industry standard networks for the base services, these safety networks add additional services to transport data with high integrity. Unfortunately the user must change their approach when going from one network or media to another. This paper presents a scalable, network independent approach to safety network design, where the safety services are described in a well defined layer, allowing the network to be changed without impacting the user's approach to safety. This approach also enables the routing of safety data, allowing the user to create end to end safety chains across multiple links without being forced to difficult to manage gateways.

Introduction

The same motivations for greater distances, increased flexibility, reduced cost and maintainability which originally moved communication networks into the industrial environment are also driving the development of industrial safety networks, along with the realization of the limitations of traditional hardwired safety solutions.

Hardwired safety systems employ safety relays which are interconnected to provide a safety function. Hardwired systems are difficult to develop and maintain for all but the most trivial applications. Furthermore, these systems place significant restrictions in the distance between devices. As safety system developers progressed beyond basic E-stop functions, they found themselves forced to fall back to hardwired logic techniques, which have been out of widespread use since the early 1970s. Even when they were successful in developing a significant size safety system, they were often costly and difficult to maintain.

Because of these issues, as well as distance and cost considerations, it is desirable to allow safety services to be implemented on standard communication networks. The key to these developments was not to create a network which couldn't fail, but to create a system where failures in the network, would cause safety devices to go to a known state. If the user knew which state the system would go to, they could make their application safe. But this meant that significantly more checking and redundant coding information would be required. Fortunately communication networks evolved and more capable inexpensive microprocessors became available to implement these additional functions.

To clarify the additional safety requirements, an existing railway standard[1] was used and later extended by the Germany Safety Bus committee[2]. This provided design guidelines to safety network developers to allow their networks and safety devices to be certified to IEC61508[3].

Unfortunately since the first safety networks were intricately tied to a particular media type or media access scheme, users were forced to change their approach to safety when they changed media or network. This also meant that users that needed a safety chain to span more than one network link, would need to employ complex safety gateways, which now became part of the safety function.

Fortunately, the Common Industrial Protocol (CIP™)[4], which allows network independent routing of standard data, could be extended to allow high integrity[1]

[1] Integrity is defined as the ability to perform a function on demand. IEC61508 requires that the probability of failure on demand be less

safety services. This paper presents a solution for a scalable, routable, network independent safety layer, thus removing the requirement for dedicated safety gateways. Since all safety devices execute the same protocol, independent of which media they resided on, the user approach is consistent and independent of media or network used.

CIP Safety: A Common Industrial Protocol for Safety

The Common Industrial Protocol (CIP) is designed to allow different networks to be used with a common protocol. Since it is designed to be media and data link independent it allows for expansion to future networks. CIP Safety is the TÜV approved[2] extension to standard CIP. It extends the model by adding CIP Safety application layer functionality, as shown in Figure 1.

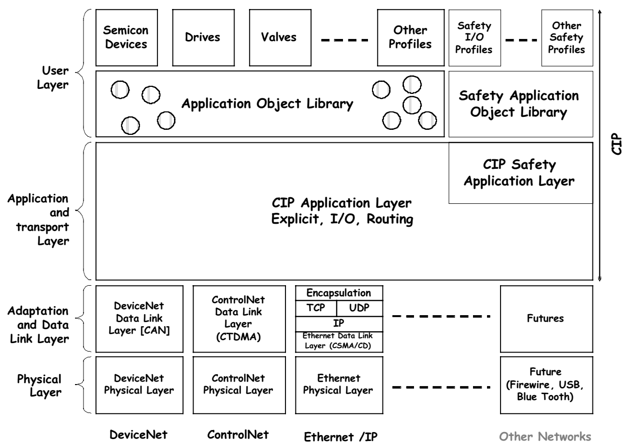

Figure 1: CIP communications layers

Because the safety application layer extensions do not rely on the integrity of the underlying standard CIP and data link layers, single channel (non-redundant) hardware can be used for the data link communication interface. This same partitioning of functionality allows standard routers to be used to route safety data, as shown in Figure 2. The routing of safety messages is possible, because the end device is responsible for ensuring the integrity of the data. If an error occurs in the transmission of data or in the intermediate router, the end device will detect the failure and take an appropriate action.

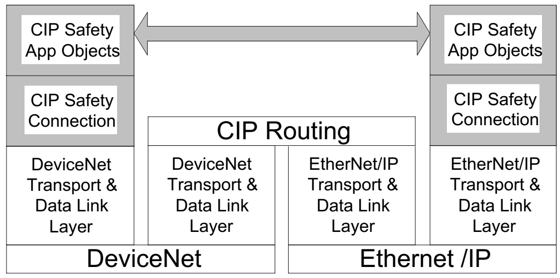

Figure 2: Routing of safety data

This routing capability allows the creation of DeviceNet™ Safety cells with quick reaction times to be interconnected with other cells via a backbone networks such as EtherNet/IP™ Safety for interlocking, as show in Figure 3. Only the safety data that is needed is routed to the required cell, which reduces the individual bandwidth requirements. The combination of fast responding local safety cells and the inter cell routing of safety data allows users to create significant safety applications with fast response times.

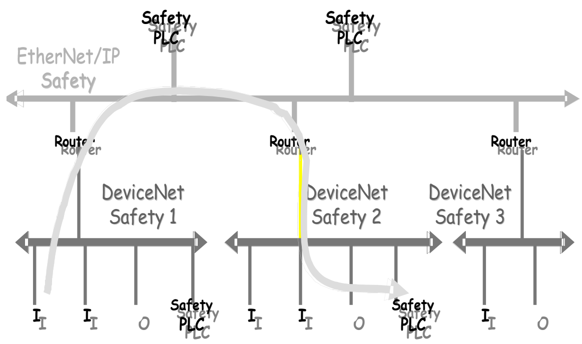

Figure 3: Network Routing

Implementing Safety

The CIP Safety application layer is specified using a Safety Validator object. This object is responsible for managing the CIP Safety connections and serves as the interface between the safety application objects and the link layer connections, as shown in Figure 4. The Safety Validator ensures the integrity of the safety data transfers.

---

than 10^{-3} for high integrity SIL3 safety applications.

[2] The CIP Safety concept has been approved by TÜV Rheinland for use in IEC61508 SIL3 and EN954-1 Cat. 4 applications

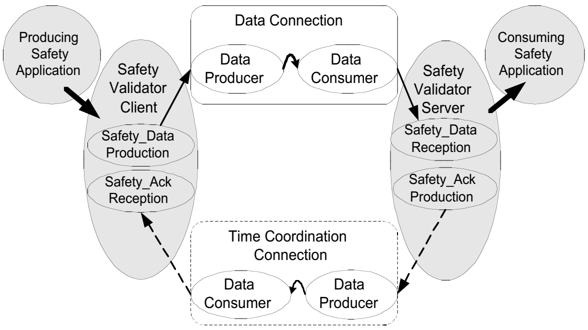

Figure 4: Relationship of Safety Validators

- The producing safety application uses an instance of a client validator to produce safety data and ensure time coordination.
- The client uses a link data producer to transmit the data and a link consumer to receive time coordination messages.
- The consuming safety application uses a server validator to receive and check data.
- The server uses a link consumer to receive data and a link producer to transmit time coordination messages.

The link producers and consumers have no knowledge of the safety packet and fulfill no safety function. The responsibility for high-integrity transfer and checking of safety data lies within the Safety Validators.

Safety Validators Ensure Integrity

CIP Safety does not prevent communication errors from occurring, but it ensures transmission integrity by detecting errors and allowing devices to take appropriate actions. The Safety Validator is responsible for detecting these communication errors. The nine communication errors which must be detected are shown in Table 1[3] along with the five measures CIP Safety uses to detect these errors.

[3] Initially based on Draft proposal test and certification guideline, safety bus systems, BG Fachausschuß Elektrotechnik 28-May-2000

	Measures to detect communication errors				
	Time Expectation via time stamp	ID for send and receive	Safety CRC	Redundancy with Cross Checking	Diverse measure
Message Repeat	X		X*		
Message Loss	X		X*		
Message Insertion	X	X	X*		
Incorrect Sequence	X		X*		
Message Corrupt			X	X	
Message Delay	X				
Coupling of safety and safety data		X			
Coupling of safety and standard data	X	X	X	X	X
Increased age of data in bridge	X				

* The Safety CRC provides additional protection for communication errors in fragmented messages.

Table 1: Error detection measures

Time Expectation via a Timestamp

All CIP Safety data is produced with a timestamp which allows safety consumers to determine the age of the produced data. This detection measure is superior to the more conventional reception timers. Reception timers can tell how much time has elapsed since a message was last received, but they do not convey any information about the actual age of the data. A timestamp allows transmission, media access/arbitration, queuing, retry and routing delays to be detected.

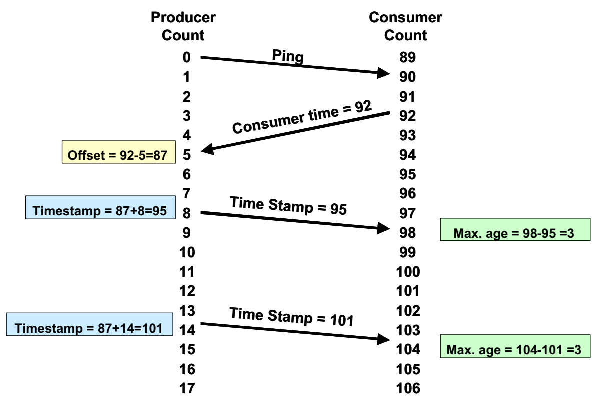

Figure 5: Timestamp

Time is coordinated between producers and consumers using ping requests and ping responses, as shown in Figure 5. After a connection is established, the producer will produce a ping request, which causes the consumer to respond with its consumer time. The producer will note the time difference between the ping production and the ping response and store this as an offset value. The producer will add this offset value to its producer time for all subsequent data transmissions. This value is transmitted as the timestamp. When the consumer receives a data message it subtracts its internal clock from the timestamp to determine the data age. If the data age is less than the maximum age allowed, the data is applied, otherwise the connection goes to the safety state. The device application is notified so that the connection safety state can be appropriately reflected.

The ping request and response sequence is repeated periodically to correct for any drift in producer or consumer crystal drift.

Production IDentifier (PID)

A Production IDentifier is encoded in each data production to ensure that each received message arrives at the correct consumer. The PID is derived from an electronic key, the device Serial Number and the CIP Connection Serial Number. Any device inadvertently receiving a message with the incorrect PID will go to a safety state. Any device that doesn't receive a message within the expected time interval with the correct PID will also go to a safety state. This measure ensures that messages are routed correctly in multilink applications.

Safety CRC (Cyclic Redundancy Code)

All safety transfers on CIP Safety use Safety CRCs to ensure the integrity of the transfer of information. The Safety CRCs serve as the primary measure to detect possible corruption of transmitted data. They provide detection up to a Hamming distance[4] of 4 for each data transfer section, though the overall Hamming distance coverage is greater for the complete transfer due to the redundancy of the protocol. The Safety CRCs are generated in the safety producers and checked in the safety consumers. Intermediate routing devices do not examine the Safety CRCs. Thus by employing end-to-end Safety CRCs, the individual data link CRCs are not part of the safety function. This eliminates certification requirements for intermediate devices and helps to ensure that the safety protocol is independent of the network technology. The Safety CRC also provides a strong protection mechanism which allows underlying data link errors such as bit stuffing[5] or fragmentation errors to be detected.

The individual link CRCs are not relied on for safety, but they are still enabled. This provides an additional level of protection and noise immunity, by allowing data retransmission for transient errors at the local link.

Redundancy and Crosscheck

Data and CRC redundancy with cross checking provides an additional measure of protection by detecting possible corruption of transmitted data. They effectively increase the Hamming distance of the protocol. These measures allow long safety data packets, up to 250 bytes, to be sent with high integrity. For short packets of 2 bytes or less, data redundancy is not required; however, redundant CRCs are cross checked to ensure integrity.

Diverse Measures for Safety and Standard

The CIP Safety protocol is present only in safety devices; this prevents standard devices from masquerading as a safety device.

[4] Hamming distance used in communication theory to measure the minimum number of bit errors required before a transmission error may not be detected.

Safety Connections

CIP Safety provides two types of safety connections:

- Single-Cast
- Multi-Cast

A Single-Cast, as shown in Figure 6, allows a Safety Validator Client to be connected to a Safety Validator Server using two link layer connections.

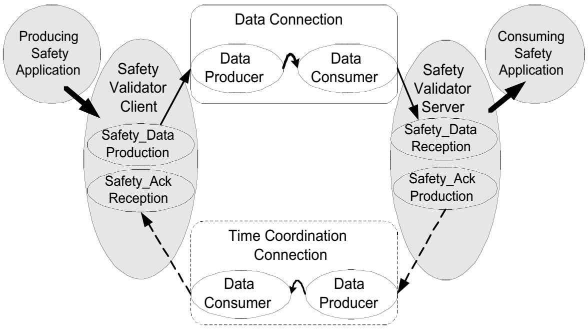

Figure 6: Single-Cast Connection

A Multi-Cast connection, as shown in Figure 7, allows up to 15 Safety Validator Servers to consume safety data from a Safety Validator Client. When the first Safety Validator Server establishes a connection with a Safety Validator Client, a pair of link layer connections are established, one for data and time correction and one for time coordination. Each new Safety Validator Server will use the existing data and time correction connection and establish a new time coordination connection with the Safety Validator Client.

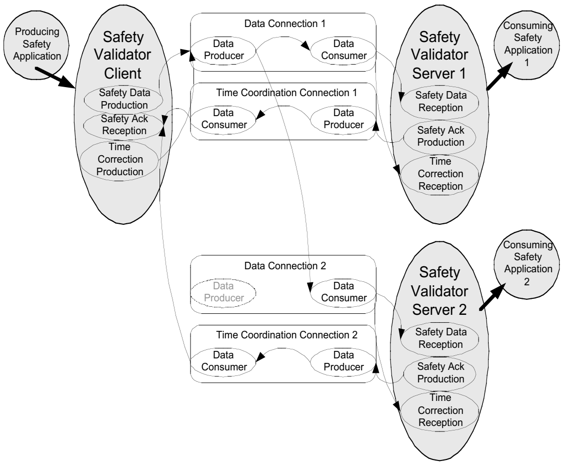

Figure 7: Multi-Cast Connection

To optimize the throughput on DeviceNet, three data link connections are used for each Multi-Cast connection, as shown in Figure 8. The data and time correction messages are sent on separate connections. This allows short messages to be transmitted on DeviceNet within a single CAN frame and reduces the overall bandwidth, since the time correction and time coordination messages are sent at a much slower periodic interval .

When Multi-Cast messages are routed off link, the router combines the data and time correction messages from DeviceNet and separates them when messages reach DeviceNet. Since the safety message contents are unchanged, the router provides no safety function.

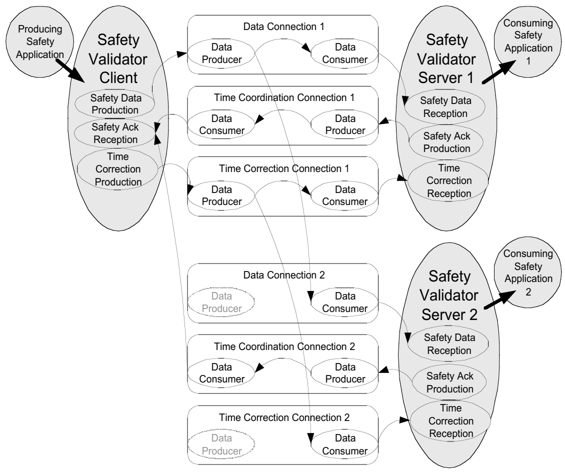

Figure 8: Multicast Connection on DeviceNet

Message Packet Sections

CIP Safety has four message sections:

1) Data section
2) Timestamp section
3) Time correction section
4) Time coordination section

CIP Safety supports two formats for the data section. The short format, shown in Figure 9, provides high integrity transmission for up to 2 bytes of safety data and serves as the primary format for most safety data messages. It includes a single instance of the safety data, an 8-bit Safety CRC and an 8-bit Safety CRC calculated on an inverted image of the data.

Short Data Section				
Actual Data	Mode Byte		Actual CRC	Comp. CRC
1 - 2 Bytes			CRC-S1	CRC-S2

Figure 9: Short data section format (1-2 bytes)

The long format, shown in Figure 10, provides high integrity transmission for up to 250 bytes of safety data. In the long format the original safety data and inverted safety data are sent along with a 16-bit Safety CRC and a 16-bit Safety CRC of the inverted safety data. This strong protection mechanism allows safety messages to be as long as 250 bytes.

Long Data Section					
Actual Data	Mode Byt		Actual CR	Complemented Data	Comp. CR
3 - 250 Bytes			CRC-S3	3 - 250 Bytes	CRC-S3

Figure 10: Long Data Section Format (3-250 bytes)

The Timestamp section of the protocol, as shown in Figure 11, is used to mark the production time of all safety productions.

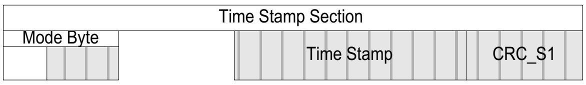

Figure 11 Timestamp Section

The time correction section, shown in Figure 12, is used only for Multi-Cast messages. It is used to adjust an individual consumer's time count for Multi-Cast connections. This section is not needed in Single-Cast messages because each producer is only associated with a single consumer.

Ack_ Byte	Consumer_Time _Value	Ack_ Byte_2	CRC-S3

Figure 12: Time Correction Section (Multi-Cast only)

The time coordination section, shown in Figure 13, contains the information sent from consumers to producers to correct the time value.

MCast_ Byte	Time_Correction Value	MCast_ Byte 2	CRC-16

Figure 13: Time coordination section

The Complete Message Telegrams

The individual message sections are appended together to form complete message telegrams. Figure 14 through Figure 16 show the message packets for short data messages (1-2 bytes).

The Single-Cast message packet, shown in Figure 14, appends the data section to the timestamp section to form the producer to consumer message packet. The consumer to producer message packet consists entirely of the time coordination message section.

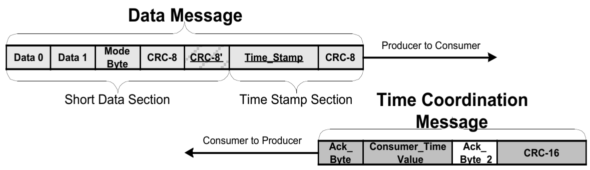

Figure 14: Single-Cast Message Packets

In the multicast message packet, shown in Figure 15, an additional Time Correction section is added to the producer to consumer message to provide time synchronization among the multiple consumers.

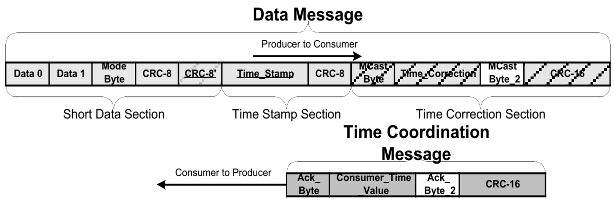

Figure 15: Multi-Cast Message Packets

When the Multi-Cast message packet is sent across DeviceNet, as shown in Figure 16, the Time Correction section is sent as a separate producer to consumer message. This optimization allows the high frequency data messages to be sent in a separate CAN frame, while the background time correction messages are produced at a slower frequency, conserving bandwidth and improving response time.

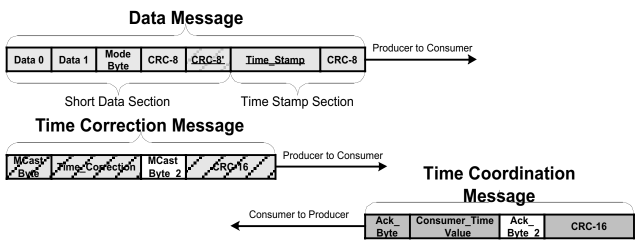

Figure 16: Multi-Cast Message Packets (DeviceNet)

The complete message telegram for long messages is formed by replacing the short data section in Figure 14 through Figure 16 with the long data section shown in Figure 10.

Configuration

Before safety devices can be used in a safety system, they must first be configured and connections must be established. The process of configuration requires configuration data from a configuration tool to be placed in a safety device. There are two possible sequences for configuration:

- Configuration Tool directly to device, or
- Via an intermediate device.

In the configuration tool to device case, as shown in Figure 17, the configuration tool writes directly to the device to be configured (1) (2).

In the case of intermediate device configuration, the tool first writes to an originator (1) and the originator writes to the target using an Originator to Target Download (3) or a *Safety_Open* service (4). The *Safety_Open* service (4) is unique in that it allows a safety connection to be established at the same time that a device is configured.

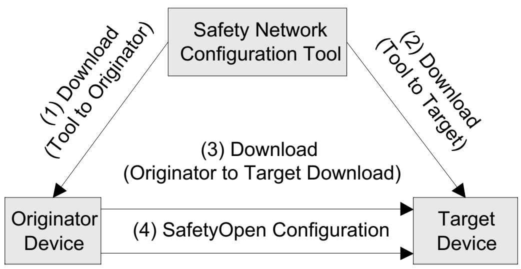

Figure 17: Configuration Transfers

Connection Establishment

The CIP protocol provides a connection establishment mechanism, using a *Forward_Open* service which allows producer to consumer connections to be established locally or across multiple links via intermediate routers. An extension of the *Forward_Open*, called the *Safety_Open* service has been created to allow the same multi-link connections for safety.

There are two types of *Safety_Open* requests:

- Type 1: With configuration
- Type 2: Without configuration

With the Type 1 *Safety_Open*, configuration and connections are established at the same time. This allows rapid configuration of devices with simple and relatively small configuration data.

With the Type 2 *SafetyOpen*, the safety device must first be configured and the *SafetyOpen* then establishes a safety connection. This separation of configuration and connection establishment allows the configuration of devices with large and complex configuration data.

In both cases, the SafetyOpen establishes all underlying link layer connections: across the local link as well as any intermediate links and routers.

Configuration Implementation

CIP Safety provides the following protection measures to ensure the integrity of configuration:

- Safety Network Number
- Password Protection
- Configuration Ownership
- Configuration Locking

Safety Network Number

The safety network number provides a unique network identifier for each network in the safety system. The safety network number combined with the local device address allows any device in the safety system to be uniquely addressed.

Password Protection

All safety devices support the use of an optional password. The password mechanism provides an additional protection measure, prohibiting the reconfiguration of a device without the correct password.

Configuration Ownership

The owner of a CIP Safety device can be specified and enforced. Each safety device can specify that its configuration is configured by a selected originator or that the configuration is only configured by a configuration tool.

Configuration Locking

Configuration locking provides the user with a mechanism to ensure that all devices have been verified and tested prior to being used in a safety application.

Safety Devices

The relationship of the objects within a safety device is shown in Figure 18. Note that CIP Safety extends the CIP common object model, with the addition of Safety I/O assemblies, Safety Validator, and Safety Supervisor objects.

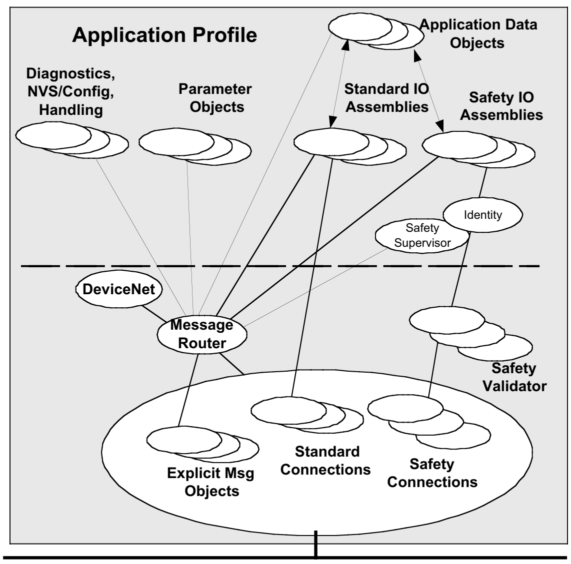

Figure 18: Safety Device Objects

Safety Supervisor

The Safety Supervisor object provides a common configuration interface for safety devices. The Safety Supervisor object centralizes and coordinates application object state behavior and related status information, exception status indications (alarms and warnings), and defines a behavior model which is assumed by objects identified as belonging to safety devices.

Summary

The concept presented in this paper demonstrates a scalable, routable network independent safety protocol based on extensions to the CIP architecture. This concept can be used in solutions ranging from device level networks such as DeviceNet™ to higher level networks such as EtherNet/IP™. By designing network independence into CIP Safety, multilink routing of safety connections can be supported. Functions such as multilink routing and Multi-Cast messaging provide a strong foundation that enable users to create the fast responding local cells and interconnect remote cells that are required for today's safety applications. The design also enables expansion to future network technologies as they become available.

David A. Vasko
Rockwell Automation
1 Allen-Bradley Drive Mayfield Hts. OH
44124 USA
Phone: 440 646 4695
Fax: 440 646 3076
davasko@ra.rockwell.com
www.rockwellautomation.com

Suresh R. Nair
Rockwell Automation
1 Allen-Bradley Drive Mayfield Hts. OH
44124 USA
Phone: 440 646 4471
Fax: 440 646 3258
srnair@ra.rockwell.com
www.rockwellautomation.com

References

[1] EN 50159-1:2001: Railway applications, communication, signaling and processing systems.

[2] Draft proposal test and certification guideline, safety bus systems, BG Fachausschuß Elektrotechnik 28-May-2000

[3] IEC 61508/1999: Functional safety of E/E/PES safety-related system

[4] CIP Common Specification, ODVA

[5] Eushiuan Tran, May 1999 Multi-Bit Error Vulnerabilities in the Controller Area Network Thesis, CMU