THE CIP NETWORKS LIBRARY

Volume 2

EtherNet/IP Adaptation of CIP

Edition 1.4

November 2007

ODVA & ControlNet International, Ltd.

The CIP Networks Library
Volume 2: EtherNet/IP Adaptation of CIP

Publication Number: PUB00002

Open DeviceNet Vendor Association, 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

The right to make, use or sell product or system implementations described herein is granted only under separate license pursuant to a Terms of Usage Agreement or other agreement. Terms of Usage Agreements for individual CIP Networks are available, at standard charges, over the Internet at the following web sites:

***www.odva.org* -** Terms of Usage Agreements for CompoNet, DeviceNet and EtherNet/IP and CIP Safety, along with general information on CIP Networks and the association of ODVA

***www.controlnet.org* -** Terms of Usage Agreement for ControlNet along with general information on ControlNet and ControlNet International.

Warranty Disclaimer Statement

Because CIP Networks my be applied in many diverse situations and in conjunction with products and systems from multiple vendors, the user and those responsible for specifying CIP Networks must determine for themselves its suitability for the intended use. The Specifications are provided to you on an AS IS basis without warranty. **NO WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE ARE BEING PROVIDED BY THE PUBLISHER, ODVA AND/OR CONTROLNET INTERNATIONAL.** In no event shall the Publisher, ODVA and/or ControlNet International, their officers, directors, members, agents, licensors or affiliates be liable to you or any customer for lost profits, development expenses or any other direct, indirect incidental, special or consequential damages.

The CIP Networks Library: Volume 2

EtherNet/IP Adaptation of CIP

Table of Contents

Revisions

The CIP Networks Library Volume 2: EtherNet/IP Adaptation of CIP, Edition 1.4 contains the following changes from Edition 1.3. Please see the change bars on the pages noted here for specific modifications. Note: Some of the pages within the ranges noted may not contain any changes.

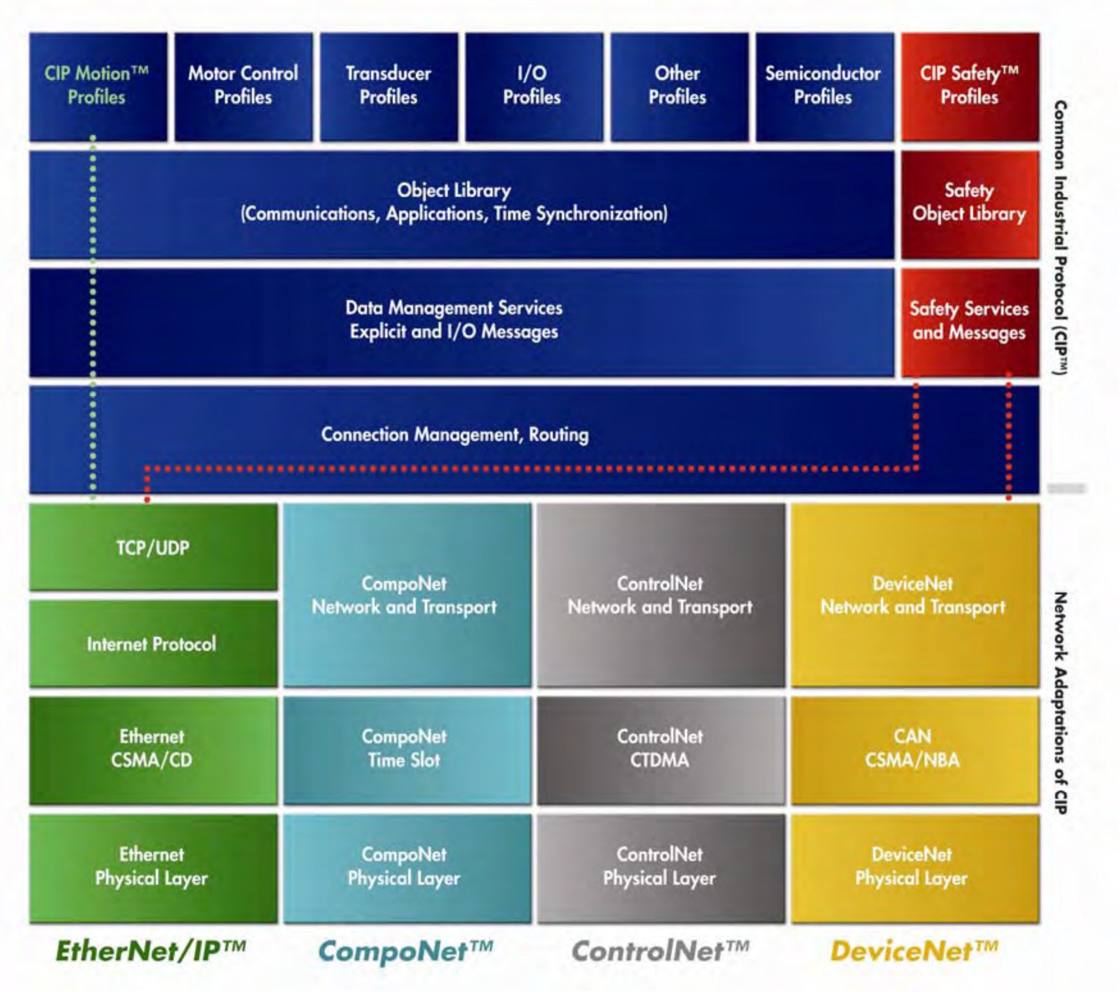
Chapter	Pages	Reason for Change
		Add support for multi-port devices
5-3.1	5-4	• Remove last sentence, which restricts references to instance 1
5-3.2.1	5-4	• Add Class attributes 2 and 3 as conditional attributes
5-3.2.2.4	5-9	• Add sentence to second paragraph
5-4.1	5-15	• Replace last sentence of first paragraph, removing the restriction to instance 1 references
5-4.2	5-15	• Add section to add missing object revision history.
5-4.3.1	5-15	• Add Class attributes 2 and 3 as conditional attributes, change revision of the Ethernet Link object to three
5-4.3.2	5-17, 18	• Add Instance attributes 7-10 as optional or conditional attributes
5-4.3.2.3	5-19	• Add sentences to the end of the paragraph
5-4.3.2.4	5-19	• Add sentences to the end of the paragraph
5-4.3.2.5	5-19	• Add sentences to the end of the paragraph
5-4.3.2.7 thru 5-4.3.2.10	5-20, 21	• Add new sections to describe new instance attributes
6-3	6-4 thru 6-7	• Add new section to show different types of multi-port devices supported by this enhancement
		EtherNet/IP Physical Layer Enhancements
1-5	1-8	• Add new terms to definitions
1-6	1-9	• Add new abbreviations
8	All	• Replaced the majority of the material in this chapter
		Cleanup typos, etc
2-5.3	2-20	• Corrected section reference for UnRegisterSession (no change bars)
2-6.1	2-22	• Improved clarity of wording for translation in Table 2-6.2
3-3.7	3-8	• Corrected number of bytes in last example in Table 3-3.1
5-4.2	5-15	• Corrected misspelling in Table 5-4.1

Preface

Organization of the CIP Networks Specifications

Today, four networks - DeviceNet™, ControlNet™, EtherNet/IP™ and CompoNet™ use the Common Industrial Protocol (CIP) for the upper layers of their network protocol. For this reason, the associations that manage these networks - ODVA and ControlNet International - have mutually agreed to manage and distribute the specifications for CIP Networks in a common structure to help ensure consistency and accuracy in the management of these specifications.

The following diagram illustrates the organization of the library of CIP Network specifications. In addition to CIP Networks, CIP Safety™ consists of the extensions to CIP for functional safety.


CIP Motion™ Profiles
Motor Control Profiles
Transducer Profiles
I/O Profiles
Other Profiles
Semiconductor Profiles
CIP Safety™ Profiles
Object Library (Communications, Applications, Time Synchronization)
Safety Object Library
Data Management Services Explicit and I/O Messages
Safety Services and Messages
Connection Management, Routing
Common Industrial Protocol (CIP™)
TCP/UDP
Internet Protocol
CompoNet Network and Transport
ControlNet Network and Transport
DeviceNet Network and Transport
Ethernet CSMA/CD
CompoNet Time Slot
ControlNet CTDMA
CAN CSMA/NBA
Ethernet Physical Layer
CompoNet Physical Layer
ControlNet Physical Layer
DeviceNet Physical Layer
Network Adaptations of CIP
EtherNet/IP™
CompoNet™
ControlNet™
DeviceNet™

This common structure presents CIP in one volume with a separate volume for each network adaptation of CIP. The specifications for the CIP Networks are two-volume sets, paired as shown below.

The EtherNet/IP specification consists of:

Volume 1: Common Industrial Protocol (CIP™)

Volume 2: EtherNet/IP Adaptation of CIP

The DeviceNet specification consists of:

Volume 1: Common Industrial Protocol (CIP™)

Volume 3: DeviceNet Adaptation of CIP

The ControlNet specification consists of:

Volume 1: Common Industrial Protocol (CIP™)

Volume 4: ControlNet Adaptation of CIP

The CompoNet specification consists of:

Volume 1: Common Industrial Protocol (CIP™)

Volume 6: CompoNet Adaptation of CIP

The specification for CIP Safety™ is distributed in a single volume:

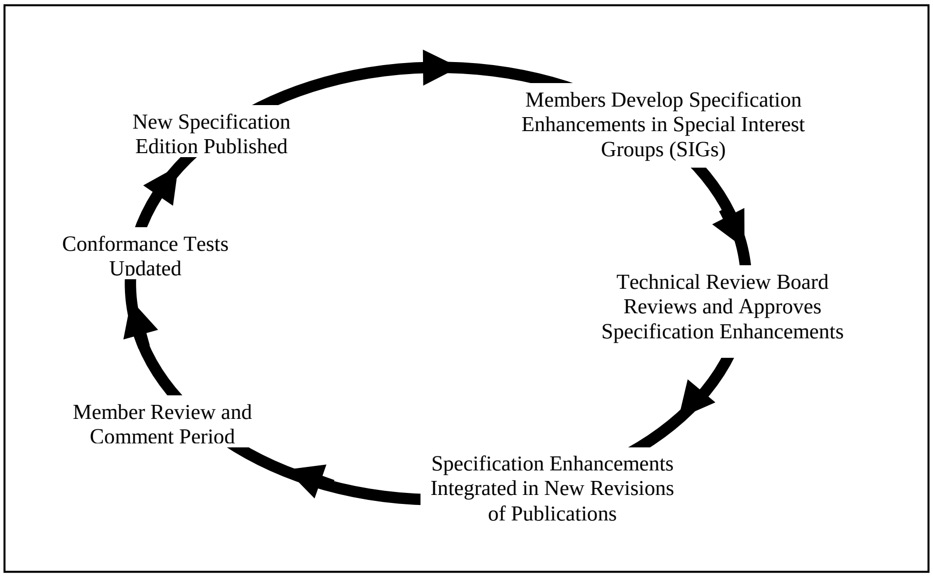
Volume 5: CIP Safety

The specification for integrating Modbus Devices is distributed in a single volume:

Volume 7: Integration of Modbus Devices into the CIP Architecture

Specification Enhancement Process

The specifications for CIP Networks are continually being enhanced to meet the increasing needs of users for features and functionality. ODVA and ControlNet International have also agreed to operate using a common Specification Enhancement Process in order to ensure open and stable specifications for all CIP Networks. This process is on going throughout the year for each CIP Network Specification as shown in the figure below. New editions of each CIP Network specification are published on a periodic basis.



Members Develop Specification Enhancements in Special Interest Groups (SIGs)
Technical Review Board Reviews and Approves Specification Enhancements
Specification Enhancements Integrated in New Revisions of Publications
Member Review and Comment Period
Conformance Tests Updated
New Specification Edition Published

Volume 2: EtherNet/IP Adaptation of CIP

Chapter 1: Introduction to EtherNet/IP

Contents

1-1 Introduction

EtherNet/IP (Ethernet/Industrial Protocol) is a communication system suitable for use in industrial environments. EtherNet/IP allows industrial devices to exchange time-critical application information. These devices include simple I/O devices such as sensors/actuators, as well as complex control devices such as robots, programmable logic controllers, welders, and process controllers.

EtherNet/IP uses CIP (Control and Information Protocol), the common network, transport and application layers also shared by ControlNet and DeviceNet. EtherNet/IP then makes use of standard Ethernet and TCP/IP technology to transport CIP communications packets. The result is a common, open application layer on top of open and highly popular Ethernet and TCP/IP protocols.

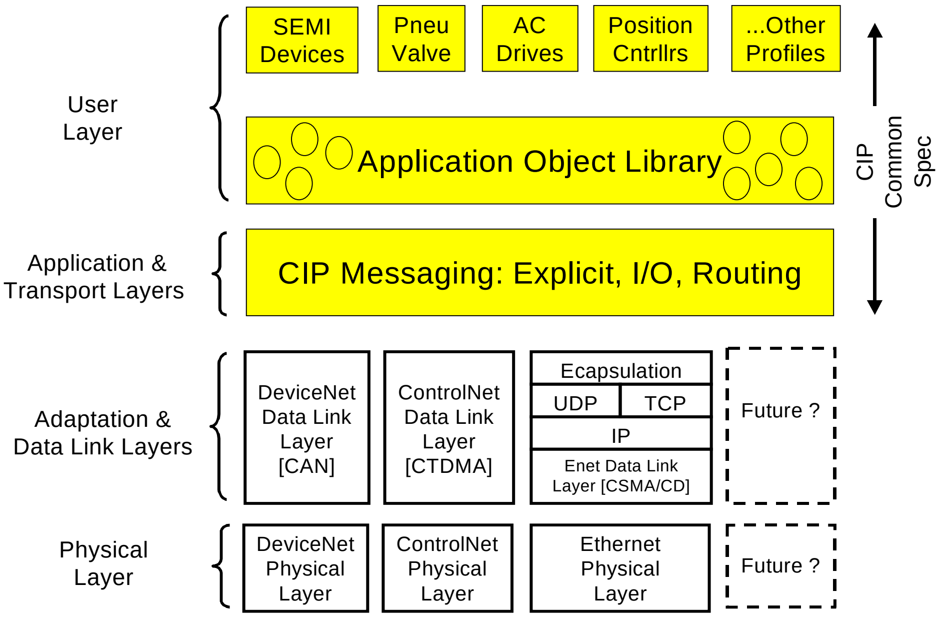
EtherNet/IP provides a producer/consumer model for the exchange of time-critical control data. The producer/consumer model allows the exchange of application information between a sending device (e.g., the producer) and many receiving devices (e.g., the consumers) without the need to send the data multiple times to multiple destinations. For EtherNet/IP, this is accomplished by making use of the CIP network and transport layers along with IP Multicast technology. Many EtherNet/IP devices can receive the same produced piece of application information from a single producing device.

EtherNet/IP makes use of standard IEEE 802.3 technology; there are no non-standard additions that attempt to improve determinism. Rather, EtherNet/IP recommends the use of commercial switch technology, with 100 Mbps bandwidth and full-duplex operation, to provide for more deterministic performance.

NOTE: EtherNet/IP does not require specific implementation or performance requirements due to the broad range of application requirements. However, work is underway to define a standard set of EtherNet/IP benchmarks and metrics by which the performance of devices will be measured. These measurements may become required entries within a product's Electronic Data Sheet. The goal of such benchmarks and metrics will be to help the user determine the suitability of a particular EtherNet/IP device for a specific application.

The figure below illustrates how EtherNet/IP, DeviceNet and ControlNet share the CIP Common layers.

Figure 1-1.1 CIP Common Overview


User Layer
SEMI Devices
Pneu Valve
AC Drives
Position Cntrllrs
...Other Profiles
Application Object Library
CIP Common Spec
Application & Transport Layers
CIP Messaging: Explicit, I/O, Routing
Adaptation & Data Link Layers
DeviceNet Data Link Layer [CAN]
ControlNet Data Link Layer [CTDMA]
Ecapsulation
UDP
TCP
IP
Enet Data Link Layer [CSMA/CD]
Future ?
Physical Layer
DeviceNet Physical Layer
ControlNet Physical Layer
Ethernet Physical Layer
Future ?

1-2 Scope

The EtherNet/IP specification is divided into the following chapters:

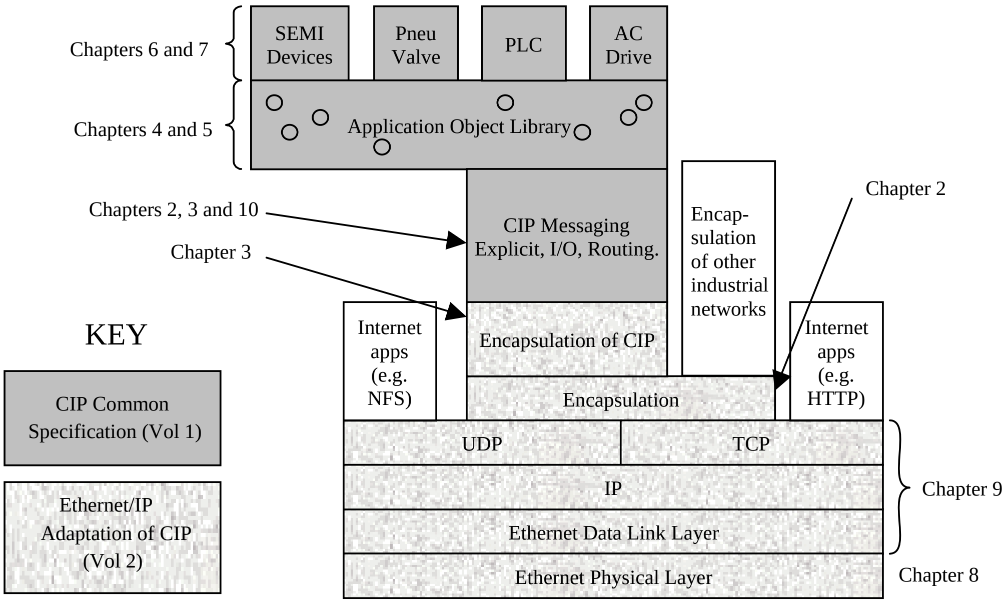
Chapter	Title	Description
1	Introduction	This chapter of the specification.
2	Encapsulation Protocol	Specifies the encapsulation protocol that is used to transport CIP packets over TCP/IP networks. The encapsulation protocol specified in this chapter may also be used to encapsulate non-CIP protocols.
3	Mapping of Explicit and I/O Messaging to TCP/IP	Contains EtherNet/IP-specific additions to the CIP Network and Transport layers. Specifies how the encapsulation protocol defined in Chapter 2 is used to transport CIP Network and Transport layer packets over TCP/IP networks.
4	Object Model	Contains EtherNet/IP-specific additions to the CIP object model.
5	Object Library	Supplements the CIP object library with objects specific to EtherNet/IP.
6	Device Profiles	Contains EtherNet/IP-specific additions to the CIP device profile library.
7	Electronic Data Sheets	Specifies additions to the CIP EDS definition required for EtherNet/IP.
8	Physical Layer	Specifies media and physical layer requirements for industrial use.
9	Indicators and Middle Layers	Specifies TCP/IP requirements of EtherNet/IP devices. This chapter also specifies the standard appearance and behaviour of EtherNet/IP diagnostic LEDs.
10	Bridging and Routing	Additions to the CIP routing definition.

This chapter is the Introduction to EtherNet/IP. The following drawing shows the relationship of these chapters to each other and to the CIP Common specification (published separately by ODVA and ControlNet International). Both this specification (volume2) and the CIP Common specification (volume1) are required to completely specify an EtherNet/IP product. The encapsulation protocol defined in Chapter 2 of this specification is also suitable to encapsulate other industrial protocols, as illustrated in the following drawing. However, the specific details of encapsulating other protocols are not included in this release of the specification.

As can be seen in Figure 1-2.1, the encapsulation protocol in chapter 2 uses a TCP/IP layer to insulate it from the network medium. As such, the encapsulation protocol may be used on any medium that supports TCP/IP. For example, the encapsulation protocol could run on an FDDI or PPP network. Chapter 9 (Indicators and Middle Layers) requires conformance with the RFC that documents how TCP/IP is implemented on a particular network. Furthermore, chapter 8 (Physical Layers) narrows the scope of certified EtherNet/IP implementations to run on either 10 or 100 Mb Ethernet. Specifically, chapter documents two permissible conformance levels of devices: one called “commercial” and the other “industrial”. Other conformance levels may be added through modification to this specification.

Figure 1-2.1 shows the relationship between the various parts of the EtherNet/IP specification. As shown in the figure, the darker sections (chapters 2-7 and 10) are predominately documented by the CIP Common specification (volume 1). The corresponding chapters of the EtherNet/IP Adaptation of CIP (volume 2) supplements or modifies these chapters of the CIP Common specification in some areas. The lightly shaded sections (chapters 2, 3, 8 and 9) are predominately documented by volume 2. These chapters contain information applicable specifically to EtherNet/IP devices, but not necessarily to those on other CIP networks (for example, DeviceNet or ControlNet).

Figure 1-2.1 Document Organization Overview



1-3 References

1-3.1 Normative References

ISO 7498-1:1984, Information processing systems — Open systems interconnection — Basic reference model

ISO 7498/AD1: 1987, Information processing systems — Open systems interconnection — Connectionless data transmission

ISO 7498-3:1987, Information processing systems — Open systems interconnection — Naming and addressing

ISO/IEC 8886:1992, Information technology — Open systems interconnection — Telecommunications and information exchange between systems — Data link service definition

ISO/IEC 10039:1990, Information technology — Telecommunication and information exchange between systems — Medium access control service definition

ISO/TR 8509:1987, Information processing systems — Open systems interconnection — Service conventions

ISO/IEC 10731:1992, Information technology — Open systems interconnection — Conventions for the definition of OSI services

ISO 8802-2:1989, Information processing systems — Local area networks — Part 2: Logical link control

ISO/IEC 8802-3:1993, Information technology — Local and metropolitan area networks — Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications

ISO/IEC 8802-4:1990, Information processing systems — Local area networks — Part 4: Token - passing bus access method and physical layer specifications

ANSI X3.159-1989, American National Standard for Information Systems — Programming Language C

1-4 Additional Reference Material

"Strategies for Real-time Systems Specification" by D. J. Hatley and I. A. Pirbhai

CEN/CENELEC Internal Regulations Part 3: Rules for the drafting and presentation of European Standards (PNE-Rules) - 1991-09

RFC 768: August 1980, User Datagram Protocol

RFC 791: September 1981, Internet Protocol

RFC 792: September 1981, Internet Control Message Protocol

RFC 793: September 1981, Transmission Control Protocol

RFC 826: November 1982, An Ethernet Address Resolution Protocol

RFC 894: April 1984, A Standard for the Transmission of IP Datagrams over Ethernet Networks

RFC 1035:1987, Domain names - implementation and specification

RFC 1103: June 1989, A Proposed Standard for the Transmission of IP Datagrams over FDDI Networks

RFC 1112: August 1989, Host Extensions for IP Multicasting

RFC 1117:1989, Internet numbers

RFC 1122: October 1989, Requirements for Internet Hosts -- Communication Layers

RFC 1123: October 1989, Requirements for Internet Hosts -- Application and Support

RFC 1127: October 1989, A Perspective on the Host Requirements RFCs

RFC 1171: July 1990, The Point-to-Point Protocol for the Transmission of Multi-Protocol Datagrams Over Point-to-Point Links

RFC 1201: February 1991, Transmitting IP Traffic over ARCNET Networks

RFC 1392: January 1993, Internet Users' Glossary

RFC2236: November 1997, Internet Group Management Protocol, Version 2

1-5 Definitions

For the purposes of this standard, the following definitions apply. Also see CIP Common Specification, Chapter 1 for additional definitions.

Term	Definition
automation outlet	The interface where the generic telecommunications cabling ends and the automation specific cabling begins, including the interfaces where automation specific cabling terminates within the automation island.
broadcast	A special type of multicast packet that all nodes on the network are always willing to receive. [Source: RFC1392]
broadcast storm	An incorrect packet broadcast onto a network that causes multiple hosts to respond all at once, typically with equally incorrect packets which causes the storm to grow exponentially in severity. [Source: RFC1392]
datagram	A self-contained, independent entity of data carrying sufficient information to be routed from the source to the destination computer without reliance on earlier exchanges between this source and destination computer and the transporting network. [Source: RFC1392]
bulkhead	A wall or barrier which maintains the ingress and climatic environmental classification applicable on either side
bulkhead connection	An assembly of two back-to-back connections separated by a bulkhead
bulkhead cable gland	A device at an enclosure bulkhead that provides cable passage for power or signals
channel	A channel is defined as the end-to-end transmission path between two points at which application-specific equipment is connected. Alternatively a channel is a path of data transfer between two end devices
encapsulation	The technique used by layered protocols in which a layer adds header information to the protocol data unit (PDU) from the layer above. As an example, in Internet terminology, a packet would contain a header from the physical layer, followed by a header from the network layer (IP), followed by a header from the transport layer (TCP), followed by the application protocol data. [Source: RFC1208]
Ethernet	A 10-Mb/s standard for LANs, initially developed by Xerox, and later refined by Digital, Intel and Xerox (DIX). All hosts are connected to a coaxial cable where they contend for network access using a Carrier Sense Multiple Access with Collision Detection (CSMA/CD) paradigm. See also: 802.x, Local Area Network, token ring. [Source: RFC1392]
EtherNet/IP	Products compliant with this specification as well as the CIP Common specification are known as EtherNet/IP products. EtherNet/IP stands for Ethernet Industrial Protocol. [Source: RFC1392]
frame	Single data transfer on a link.
link	The physical connection between two active mating components
MAC ID	the 48-bit physical address of an Ethernet node
network status indicators	Indicators on a node indicating the status of the Physical and Data Link Layers.
network address or node address	A node's 32-bit TCP/IP address on the link. In most CIP networks, this network address is the MAC ID; however, this is not the case on Ethernet. The DLL of Ethernet has a 48-bit MAC ID that is not used directly by the CIP communication stack.
physical medium dependant	An active interface defined by the appropriate standards to serve a specific medium such as copper 2/4 pair or fiber
physical topology	The physical layout of devices on a network, or the way that the devices on a network are arranged and how they communicate with each other, is called the physical topology

Term	Definition
port	Within the EtherNet/IP specific context, a TCP or UDP port is a transport layer demultiplexing value. Each application has a unique port number associated with it. [Source: RFC1392]. See CIP Common Specification for an additional definition of this term.
Power over Ethernet	The delivery of device power along with Ethernet signals, as defined by 803.3an in cooperation with TIA-TR42 standards committee
redundant media	A system using more than one medium to help prevent communication failures.
segment	Trunk–cable sections connected via taps with terminators at each end; a segment has no active components and does not include repeaters.
transceiver	The physical component within a node that provides transmission and reception of signals onto and off of the medium.

1-6 Abbreviations

For the purposes of this standard, the following abbreviations apply. Also see the CIP Common Specification Chapter 1 for additional abbreviations.

Abbreviation	Meaning
AO	Automation Outlet
COTS	Commercially off the shelf. Refers to commercial grade components
FTP	File transfer protocol. An internet application that uses TCP reliable packet transfer to move file between different nodes. (not to be confused with STP/FTP)
LED	Light emitting diode
PMD	Physical Media Dependant
PoE	Power over Ethernet
rcv	Receive
RFC	Request For Comments (RFC) – The document series, begun in 1969, which describes the Internet suite of protocols and related experiments. Not all (in fact, very few) RFCs describe the Internet standards, but all Internet standards are written up as RFCs. The RFC series of documents is unusual in that the proposed protocols are forwarded by the Internet research and development community, acting on their own behalf, as opposed to the formally reviewed and standardized protocols that are promoted by organizations such as CCITT and ANSI. [Source: RFC 1392]
rx	Receive
STP/FTP	Shielded twisted pair/foil twisted pair
TCP	Transmission Control Protocol (TCP) - An Internet Standard transport layer protocol defined in STD 7, RFC 793. It is connection-oriented and stream-oriented, as opposed to UDP. See also: connection-oriented, stream-oriented, User Datagram Protocol. [Source: RFC1392]
Tx	transmit
UDP	User Datagram Protocol (UDP) - An Internet Standard transport layer protocol defined in STD 6, RFC 768. It is a connectionless protocol which adds a level of reliability and multiplexing to IP. See also: connectionless, Transmission Control Protocol. [Source: RFC1392]
UTP	Unshielded twisted pair
Xmit	transmit

This page is intentionally left blank

Volume 2: EtherNet/IP Adaptation of CIP

Chapter 2: Encapsulation Protocol

Contents

2-1 Introduction

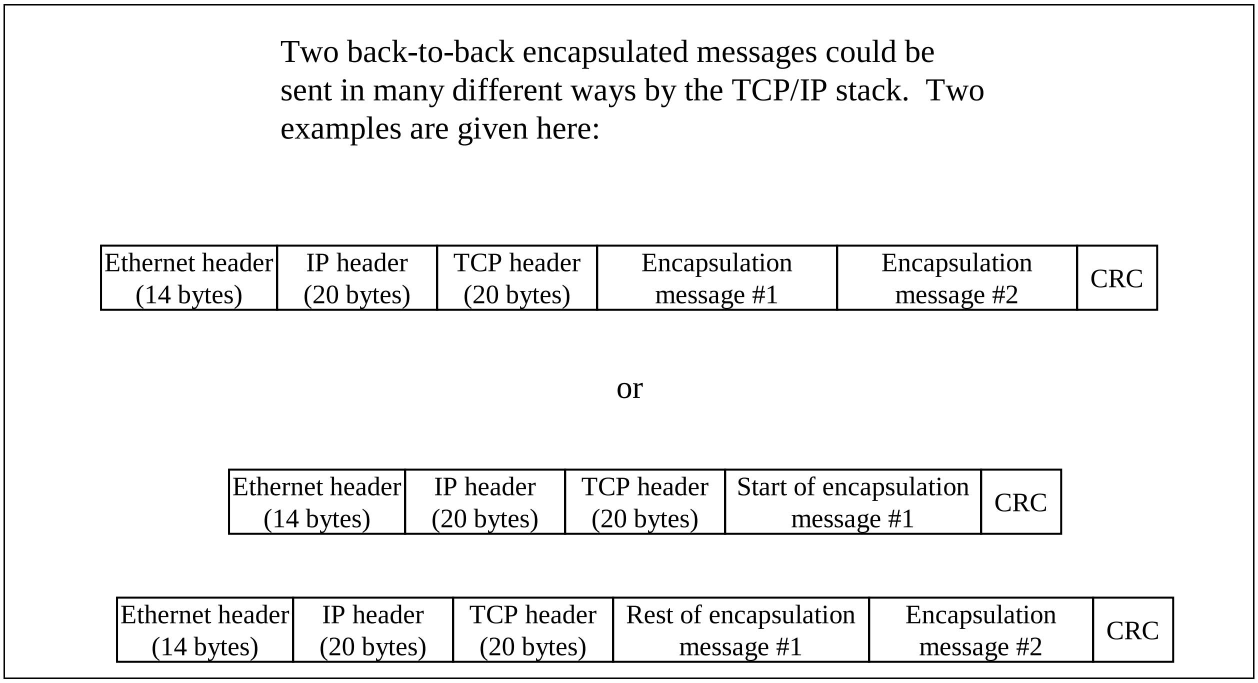
This chapter (chapter 2) of the specification documents the method used to encapsulate industrial protocols on a TCP/IP network. This mechanism can be applied to the CIP industrial protocol or to other networks. Chapter 3 of this specification details the application of this encapsulation protocol to CIP.

With respect to the OSI reference model, this encapsulation protocol inhabits Layer 2 Data Link functions.

2-2 Use of TCP and UDP

The encapsulation protocol defines a reserved TCP port number that shall be supported by all EtherNet/IP devices. All EtherNet/IP devices shall accept at least 2 TCP connections on TCP port number 0xAF12. Once the TCP connection to TCP port number 0xAF12 is established, all data sent through the TCP stream shall be in the format specified in section 2-3.

NOTE: TCP is a stream-based protocol. It is permitted to send almost any length IP packet it chooses. For example, if two back-to-back encapsulated messages were passed to a TCP/IP stack, the TCP/IP stack may choose to put both encapsulated messages in one Ethernet frame. Alternately, it may choose to place half of the first message in the first Ethernet frame and all the rest in the next Ethernet frame. This is shown in Figure 2-2.1 Usage of TCP to Encapsulate Two Messages.



Two back-to-back encapsulated messages could be sent in many different ways by the TCP/IP stack. Two examples are given here:

Ethernet header (14 bytes)	IP header (20 bytes)	TCP header (20 bytes)	Encapsulation message #1	Encapsulation message #2	CRC

or

Ethernet header (14 bytes)	IP header (20 bytes)	TCP header (20 bytes)	Start of encapsulation message #1	CRC

Ethernet header (14 bytes)	IP header (20 bytes)	TCP header (20 bytes)	Rest of encapsulation message #1	Encapsulation message #2	CRC

Figure 2-2.1 Usage of TCP to Encapsulate Two Messages

NOTE: It is not the intention of this specification to document the details of the TCP, UDP and IP transport mechanisms. Many excellent resources including the RFCs referenced throughout this specification should be used to obtain this information.

The encapsulation protocol also defines a reserved UDP port number that shall be supported by all EtherNet/IP devices. All devices shall accept UDP packets on UDP port number 0xAF12. Since UDP, unlike TCP, does not have an ability to reorder packets, whenever UDP is used to send an encapsulated message, the entire message shall be sent in a single UDP packet. Only one encapsulated message shall be present in a single UDP packet destined to UDP port 0xAF12.

Some encapsulated messages shall only be sent via TCP. Other may be sent via either UDP or TCP. See “Table 2-3.2 Encapsulation Commands” for details about which commands are restricted to TCP.

2-3 Encapsulation Messages

2-3.1 Encapsulation Packet Structure

All encapsulation messages, sent via TCP or sent to UDP port 0xAF12, shall be composed of a fixed-length header of 24 bytes followed by an optional data portion. The total encapsulation message length (including header) shall be limited to 65535 bytes. Its structure shall be as follows:

Table 2-3.1 Encapsulation Packet

Structure	Field Name	Data Type	Field Value
Encapsulation header	Command	UINT	Encapsulation command
	Length	UINT	Length, in bytes, of the data portion of the message, i.e., the number of bytes following the header
	Session handle	UDINT	Session identification (application dependent)
	Status	UDINT	Status code
	Sender Context	ARRAY of octet	Information pertinent only to the sender of an encapsulation command. Length of 8.
	Options	UDINT	Options flags
Command specific data	Encapsulated data	ARRAY of 0 to 65511 octet	The encapsulation data portion of the message is required only for certain commands

The encapsulation message length shall not override length restrictions imposed by the encapsulated protocol.

NOTE: For example, a CIP UCMM message is still limited to 504 bytes even when encapsulated. See chapter 3 of the CIP Common Specification.

Multi-byte integer fields in the encapsulation messages shall be transmitted in little-endian byte order.

NOTE: This is different from the byte ordering used in standard Internet network protocols, which is big-endian.

Although the header contains no explicit information to distinguish between a request and a reply, this information shall be determined in either of two ways:

- implicitly, by the command and the context in which the message is generated. (For example, in the case of the RegisterSession command, the request is generated by an originator and the target generates the reply);
- explicitly, by the contents of an encapsulated protocol packet in the data part of the message.

2-3.2 Command Field

The allocation of command codes shall be as follows:

Table 2-3.2 Encapsulation Commands

Code	Name	Comment
0x0000	NOP	(may be sent only using TCP)
0x0001	Reserved for legacy (RA)	
0x0002 and 0x0003	Reserved for legacy (RA)	
0x0004	ListServices	(may be sent using either UDP or TCP)
0x0005	Reserved for legacy (RA)	
0x0006 through 0x0062	Reserved for future expansion of this specification (Products compliant with this specification shall not use command codes in this range)	
0x0063	ListIdentity	(may be sent using either UDP or TCP)
0x0064	ListInterfaces	**optional** (may be sent using either UDP or TCP)
0x0065	RegisterSession	(may be sent only using TCP)
0x0066	UnRegisterSession	(may be sent only using TCP)
0x0067 through 0x006E	Reserved for legacy (RA)	
0x006F	SendRRData	(may be sent only using TCP)
0x0070	SendUnitData	(may be sent only using TCP)
0x0071	Reserved for legacy (RA)	
0x0072	IndicateStatus	**optional** (may be sent only using TCP)
0x0073	Cancel	**optional** (may be sent only using TCP)
0x0074 through 0x00C7	Reserved for legacy (RA)	
0x00C8 through 0xFFFF	Reserved for future expansion of this specification (Products compliant with this specification shall not used command codes in this range)	

A device shall accept commands that it does not support without breaking the session or underlying TCP connection. A status code indicating that an unsupported command was received shall be returned to the sender of the message.

NOTE: The establishment of a session is defined in section 2-5. In short, a session makes a TCP/IP connection between originator and target over which encapsulated commands may be sent. Since TCP/IP connections are modeled as a stream of bytes, the encapsulation header is prepended to each encapsulated packet so that the receiving device can know where packets begin and end.

2-3.3 Length Field

The length field in the header shall specify the size in bytes of the data portion of the message. The field shall contain zero for messages that contain no data. The total length of a message shall be the sum of the number contained in the length field plus the 24-byte size of the encapsulation header.

The entire encapsulation message shall be read from the TCP/IP connection even if the length is invalid for a particular command or exceeds the target's internal buffers.

NOTE: Failure to read the entire message can result in losing track of the message boundaries in the TCP byte stream.

2-3.4 Session Handle

The Session Handle shall be generated by the target and returned to the originator in response to a RegisterSession request. The originator shall insert it in all subsequent encapsulated packets (sent using the commands listed in Table 2-3.2) to that particular target. In the case where the target initiates and sends a command to the originator, the target shall include this field in the request that it sends to the originator.

NOTE: Some commands (i.e., NOP) do not require a session handle even if a session has been established. Whether or not a particular command requires a session is noted in the description of that command.

2-3.5 Status Field

The value in the Status field shall indicate whether or not the receiver was able to execute the requested encapsulation command. A value of zero in a reply shall indicate successful execution of the command. In all requests issued by the sender, the Status field shall contain zero. If the receiver receives a request with a non-zero Status field, the request shall be ignored and no reply shall be generated.

NOTE: This field does not reflect errors that are generated by an encapsulated protocol packet contained within the data portion of the message. For example, an error encountered during an end node's processing of a Set Attributes service would be returned via the CIP specified error mechanism (see chapter 3 of the CIP Common specification)..

The status codes shall be as follows:

Table 2-3.3 Error Codes

Status Code	Description
0x0000	Success
0x0001	The sender issued an invalid or unsupported encapsulation command.
0x0002	Insufficient memory resources in the receiver to handle the command. This is not an application error. Instead, it only results if the encapsulation layer cannot obtain memory resources that it needs.
0x0003	Poorly formed or incorrect data in the data portion of the encapsulation message.
0x0004 – 0x0063	Reserved for legacy (RA)
0x0064	An originator used an invalid session handle when sending an encapsulation message to the target.
0x0065	The target received a message of invalid length
0x0066 – 0x0068	Reserved for legacy (RA)
0x0069	Unsupported encapsulation protocol revision.
0x006A – 0xFFFF	Reserved for future expansion (Products compliant with this specification shall not use command codes in this range)

2-3.6 Sender Context Field

The sender of the command shall assign the value in the Sender Context field of the header. The receiver shall return this value without modification in its reply. Commands with no expected reply may ignore this field.

NOTE: The sender of a command may place any value in this field. It could be used to match requests with their associated replies.

2-3.7 Options Field

The sender of an encapsulated packet shall set the options field to zero. The receiver of an encapsulated packet shall verify that the option field is zero. The receiver shall discard encapsulated packets with a non-zero option field.

NOTE: The intent of this field is to provide bits that modify the meaning of the various encapsulation commands. No particular use for this field has not yet been specified.

2-3.8 Command Specific Data Field

NOTE: The structure of the command specific data field depends on the command code. To organize their command specific data field, most commands use either or both of the following two methods:

1) use a fixed structure
2) use the common packet format (described in section 2-6)

The common packet format allows commands to structure their command specific data field in an extensible way.

2-4 Command Descriptions

2-4.1 NOP

Either an originator or a target may send a NOP command. No reply shall be generated by this command. The data portion of the command shall be from 0 to 65511 bytes long. The receiver shall ignore any data that is contained in the message.

NOTE: A NOP provides a way for either an originator or target to determine if the TCP connection is still open.

The NOP encapsulation header shall be as follows:

Table 2-4.1 NOP Header Values

Structure	Field Name	Data Type	Field Value
Encapsulation header	Command	UINT	NOP (0x00)
	Length	UINT	Length of data portion (from 0 to 65511)
	Session handle	UDINT	This field is ignored since the NOP command does not generate a reply.
	Status	UDINT	shall be 0
	Sender Context	ARRAY of octet	This field is ignored since the NOP command does not generate a reply. Length of 8.
	Options	UDINT	shall be 0
Command specific data	Unused data	ARRAY of octet	unused data

2-4.2 ListIdentity

2-4.2.1 General

A connection originator may use the ListIdentity command to locate and identify potential targets. This command shall be sent as a broadcast message using UDP and does not require that a session be established.

2-4.2.2 Request

The ListIdentity request shall be as shown below:

Table 2-4.2 ListIdentity Request

Structure	Field Name	Data Type	Field Value
Encapsulation header	Command	UINT	ListIdentity (0x63)
	Length	UINT	0
	Session handle	UDINT	This field is ignored since a session need not be established before sending the ListIdentity request.
	Status	UDINT	0
	Sender Context	ARRAY of octet	0
	Options	UDINT	0

2-4.2.3 Reply

One reply item is defined for this command, Target Identity, with item type code 0x0C. This item shall be supported (returned) by all CIP capable devices.

Each receiver of the List Identity command shall reply with a standard encapsulation header and data, as shown below. The data portion of the message shall provide the information on the target's identity. The reply shall be sent to the IP address from which the broadcast request was received.

Table 2-4.3 ListIdentity Reply

Structure	Field Name	Data Type	Field Value
Encapsulation header	Command	UINT	List Identity (0x63)
	Length	UINT	0
	Session handle	UDINT	Ignored
	Status	UDINT	0
	Sender Context	ARRAY of octet	0
	Options	UDINT	0
Command specific data	Item Count	UINT	Number of target items to follow
	ListIdentity Items	STRUCT of	Interface Information
		UINT	Item Type Code
		UINT	Item Length
		ARRAY of octet	Item Data

The data portion of the message shall be the Common Packet Format that contains a 2-byte item count followed by an array of items providing the target identity.

At a minimum, the CIP Identity item shall be returned and has the format as defined in Table 2-4.4. Part of this item definition follows the Get Attribute All service response definition of the Identity Object (data returned based on instance one of this object), and may adopt new members if and when new members are added to that service response. Unlike most fields in the Common Packet Format, the Socket Address field shall be sent in big endian order.

Table 2-4.4 CIP Identity Item

Parameter Name	Data Type	Description
Item Type Code	UINT	Code indicating item type of CIP Identity (0x0C)
Item Length	UINT	Number of bytes in item which follow (length varies depending on Product Name string)
Encapsulation Protocol Version	UINT	Encapsulation Protocol Version supported (also returned with Register Sesstion reply).
Socket Address	STRUCT of	Socket Address (see section 2-6.3.2)
	INT	sin_family (**big-endian**)
	UINT	sin_port (**big-endian**)
	UDINT	sin_addr (**big-endian**)
	ARRAY of USINT	sin_zero (length of 8) (big-endian)
Vendor ID1	UINT	Device manufacturers Vendor ID
Device Type1	UINT	Device Type of product
Product Code1	UINT	Product Code assigned with respect to device type
Revision1	USINT[2]	Device revision
Status1	WORD	Current status of device
Serial Number1	UDINT	Serial number of device
Product Name1	SHORT_STRING	Human readable description of device
State1	USINT	Current state of device

1 These parameters are further defined by the corresponding instance attribute of the Identity Object. (see the CIP Common specification, chapter 5, Object Library)

2-4.3 ListInterfaces

2-4.3.1 General

The optional List Interfaces command shall be used by a connection originator to identify potential non-CIP communication interfaces associated with the target. A session need not be established to send this command.

2-4.3.2 Request

The ListInterfaces request shall be as shown below.

Table 2-4.5 ListInterfaces Request

Structure	Field Name	Data Type	Field Value
Encapsulation header	Command	UINT	List Interfaces (0x64)
	Length	UINT	0
	Session handle	UDINT	Ignored
	Status	UDINT	0
	Sender Context	ARRAY of octet	0
	Options	UDINT	0

2-4.3.3 Reply

If supported, the receiver of a ListInterfaces request command shall reply with a standard encapsulation header and data, as shown below. The data portion of the message is structured as a Common Packet Format and shall provide information on the non-CIP communication interfaces associated with the target.

Table 2-4.6 ListInterfaces Reply

Structure	Field Name	Data Type	Field Value
Encapsulation header	Command	UINT	List Interfaces (0x64)
	Length	UINT	0
	Session handle	UDINT	Ignored
	Status	UDINT	0
	Sender Context	ARRAY of octet	0
	Options	UDINT	0
Command specific data	Item Count	UINT	Number of target items to follow
	Target Items	STRUCT of	Interface Information
		UINT	Item Type Code
		UINT	Item Length
		ARRAY of octet	Item Data

The data portion of the message shall be the Common Packet Format, which contains a 2-byte item count followed by an array of items providing interface information. There are no publicly defined items returned with this reply. The vendor-specific item(s) which is/are returned shall, at a minimum, return a 32 bit Interface Handle which is used by other encapsulation commands, for example, the SendRRData command.

2-4.4 RegisterSession

2-4.4.1 General

An originator shall send a RegisterSession command to a target to initiate a session.

NOTE: See section 2-5, for detailed information on establishing and maintaining a session.

2-4.4.2 Request

The RegisterSession request shall be as follows:

Table 2-4.7 RegisterSession Request

Structure	Field Name	Data Type	Field Value
Encapsulation header	Command	UINT	RegisterSession (0x65)
	Length	UINT	4 bytes
	Session handle	UDINT	0
	Status	UDINT	0
	Sender Context	ARRAY of octet	Any sender context. Length of 8.
	Options	UDINT	0
Command specific data	Protocol version	UINT	Requested protocol version shall be set to 1.
	Options flags	UINT	Session options shall be set to 0 Bits 0-7 are reserved for legacy (RA) Bits 8-15 are reserved for future expansion NOTE: This field is not the same as the option flags in the encapsulation header.

2-4.4.3 Reply

The target shall send a RegisterSession reply to indicate that it has registered the originator. The reply shall have the same format as the request as shown below:

Table 2-4.8 RegisterSession Reply

Structure	Field Name	Data Type	Field Value
Encapsulation header	Command	UINT	RegisterSession (0x65)
	Length	UINT	4 bytes
	Session handle	UDINT	handle returned by target
	Status	UDINT	0
	Sender Context	ARRAY of octet	context preserved from the corresponding RegisterSession request. Length of 8.
	Options	UDINT	0
Command specific data	Protocol version	UINT	Requested protocol version shall be set to 1.
	Options flags	UINT	Session options shall be set to 0 Bits 0-7 are reserved for legacy (RA) Bits 8-15 are reserved for future expansion NOTE: This field is not the same as the option flags in the encapsulation header.

The Session Handle field of the header shall contain a target-generated identifier that the originator shall save and insert in the Session Handle field of the header for all subsequent requests to that target. This field shall be valid only if the Status field is zero (0).

The Sender Context field of the header shall contain the same values present in the original sender request. If the originator has been registered with the target, the Status field shall be zero (0). If the target was unable to register, the Status field shall be set to 0x69 (unsupported encapsulation protocol revision).

The Protocol Version field shall equal the requested version if the originator was successfully registered. If the target does not support the requested version of the protocol,

- the session shall not be created;
- the Status field shall be set to unsupported encapsulation protocol 0x69;
- the target shall return the highest supported version in the Protocol Version field.

If all requested options are supported, the Options field shall return the originator's value. This value shall be zero.

2-4.5 UnRegisterSession

Either an originator or a target may send this command to terminate the session. The receiver shall initiate a close of the underlying TCP/IP connection when it receives this command. The session shall also be terminated when the transport connection between the originator and target is terminated. The receiver shall perform any other associated cleanup required on its end. There shall be no reply to this command.

The UnregisterSession command format shall be as follows:

Table 2-4.9 UnregisterSession Command

Structure	Field Name	Data Type	Field Value
Encapsulation header	Command	UINT	UnRegisterSession (0x66)
	Length	UINT	0 bytes
	Session handle	UDINT	handle from RegisterSession reply
	Status	UDINT	shall be 0
	Sender Context	ARRAY of octet	Any sender context. Length of 8.
	Options	UDINT	shall be 0

The Session Handle shall be set to the value obtained by the original RegisterSession reply. Once the client has sent this command, it shall no longer use the handle. .

NOTE: See section 2-5.3 for more detail about terminating a session.

2-4.6 ListServices

2-4.6.1 General

The ListServices command shall determine which encapsulation service classes the target device supports.

NOTE: Each service class has a unique type code, and an optional ASCII name.

2-4.6.2 Request

The ListServices header shall be as follows:

Table 2-4.10 ListServices Request

Structure	Field Name	Data Type	Field Value
Encapsulation header	Command	UINT	ListServices (0x04)
	Length	UINT	0 bytes
	Session handle	UDINT	ignored
	Status	UDINT	0
	Sender Context	ARRAY of octet	Any sender context. Length of 8.
	Options	UDINT	0

2-4.6.3 Reply

The receiver shall reply with a standard encapsulation message, consisting of the header and data, as shown below. The data portion of the message shall provide the information on the services supported.

Table 2-4.11 ListServices Reply

Structure	Field Name	Data Type	Field Value
Encapsulation header	Command	UINT	ListServices (0x04)
	Length	UINT	Length of data portion
	Session handle	UDINT	ignored
	Status	UDINT	0
	Sender Context	ARRAY of octet	Preserved from the corresponding ListServices request. Length of 8.
	Options	UDINT	0
Command specific data	Item Count	UINT	Number of items to follow
	Target Items	STRUCT of	Interface Information
		UINT	Item Type Code
		UINT	Item Length
		UINT	Version of encapsulated protocol shall be set to 1
		UINT	Capability flags
		ARRAY of 16 USINT	Name of service

The Type Code shall identify the service class as follows:

One service class is defined, with type code 0x100 and name "Communications". This service class shall indicate that the device supports encapsulation of CIP packets. All devices that support encapsulating CIP shall support the ListServices request and Communications service class.

NOTE: See section 2-6 for a description of items and a list of all reserved item codes.

The Version field shall indicate the version of the service supported by the target to help maintain compatibility between applications.

Each service shall have a different set of capability flags. Unused flags shall be set to zero.

The Capability Flags, defined for the Communications service, shall be as follows:

Table 2-4.12 Capability Flags

Flag Value	Description
Bits 0 – 4	reserved for legacy (RA)
Bit 5	If the device supports CIP packet encapsulation via TCP this bit shall be set (=1); otherwise, it shall be clear (=0)
Bits 6 – 7	reserved for legacy (RA)
Bit 8	Supports CIP Class 0 or 1 UDP-based connections
Bits 9 – 15	Reserved for future expansion

The Name field shall allow up to a 16-byte, NULL-terminated ASCII string for descriptive purposes only. The 16-byte limit shall include the NULL character.

2-4.7 SendRRData

2-4.7.1 General

A SendRRData command shall transfer an encapsulated request/reply packet between the originator and target, where the originator initiates the command. The actual request/reply packets shall be encapsulated in the data portion of the message and shall be the responsibility of the target and originator.

NOTE: When used to encapsulate the CIP, the SendRRData request and response are used to send encapsulated UCMM messages (unconnected messages). See chapter 3 for more detail.

2-4.7.2 Request

The SendRRData header shall be as follows:

Table 2-4.13 SendRRData Request

Structure	Field Name	Data Type	Field Value
Encapsulation header	Command	UINT	SendRRData (0x6F)
	Length	UINT	Length of data portion
	Session handle	UDINT	Handle returned by RegisterSession
	Status	UDINT	0
	Sender Context	ARRAY of octet	Any sender context. Length of 8.
	Options	UDINT	0
Command specific data	Interface handle	UDINT	shall be 0 for CIP
	Timeout	UINT	operation timeout
	Encapsulated packet	ARRAY of octet	see Common Packet Format specification in section 2-6)

The Interface handle shall identify the Communications Interface to which the request is directed. This handle shall be 0 for encapsulating CIP packets.

The target shall abort the requested operation after the timeout expires. When the “timeout” field is in the range 1 to 65535, the timeout shall be set to this number of seconds. When the “timeout” field is set to 0, the encapsulation protocol shall not have its own timeout. Instead, it shall rely on the timeout mechanism of the encapsulated protocol.

NOTE: When used to encapsulate CIP packets, the timeout field is usually set to 0 since CIP provides its own timeout mechanism for connected messages.

The encapsulated protocol packet shall be encoded in the Common Packet Format as shown in section 2-6.

2-4.7.3 Reply

The SendRRData reply, as shown below, shall contain data in response to the SendRRData request. The reply to the original encapsulated protocol request shall be contained in the data portion of the SendRRData reply.

Table 2-4.14 SendRRData Reply

Structure	Field Name	Data Type	Field Value
Encapsulation header	Command	UINT	SendRRData (0x6F)
	Length	UINT	length of data structure
	Session handle	UDINT	handle returned by RegisterSession
	Status	UDINT	0
	Sender Context	ARRAY of octet	Preserved from the corresponding SendRRData request. Length of 8.
	Options	UDINT	0
Command specific data	Interface handle	UDINT	shall be 0 for CIP
	Timeout	UINT	operation timeout (not used)
	Encapsulated packet	ARRAY of octet	see Common Packet Format specification in section 2-6)

The format of the data portion of the reply message shall be the same as that of the SendRRData request message.

NOTE: Since the request and reply share a common format, the reply message contains a timeout field; however, it is not used.

2-4.8 SendUnitData

The SendUnitData command shall send encapsulated connected messages. This command may be used when the encapsulated protocol has its own underlying end-to-end transport mechanism. A reply shall not be returned. The SendUnitData command may be sent by either end of the TCP connection.

NOTE: When used to encapsulate the CIP, the SendUnitData command is used to send CIP connected data in both the O⇒T and T⇒O directions.

The format of the SendUnitData command shall be as follows:

Table 2-4.15 SendUnitData Command

Structure	Field Name	Data Type	Field Value
Encapsulation header	Command	UINT	SendUnitData (0x70)
	Length	UINT	Length of data portion
	Session handle	UDINT	Handle returned by RegisterSession
	Status	UDINT	0
	Sender Context	ARRAY of octet	Any sender context. Length of 8.
	Options	UDINT	0
Command specific data	Interface handle	UDINT	shall be 0
	Timeout	UINT	shall be 0
	Encapsulated packet	ARRAY of octet	see Common Packet Format specification in section 2-6)

Interface handle and Timeout shall be set the zero. The timeout field is not used since no reply is generated upon receipt of a SendUnitData command.

2-5 Session Management

2-5.1 Phases of a TCP Encapsulation Session

An encapsulation session shall have three phases:

- establishing a session;
- maintaining a session;
- closing a session.

2-5.2 Establishing a Session

Session establishment shall proceed according to the following steps:

- The originator shall open a TCP/IP connection to the target, using the reserved TCP port number (0xAF12), or if specified, the TCP port number from the connection path (the means to specify an alternate TCP port number is described in chapter 3);
- The originator shall send a RegisterSession command to the target (see section 2-4.4 for a description of the RegisterSession command);
- The target shall check the protocol version in the command message to verify it supports the same protocol version as the originator. If not, the target shall return a RegisterSession with an appropriate Status field along with the highest supported protocol version;
- The target shall assign a new (unique) Session ID and shall send a RegisterSession reply to the originator.

2-5.3 Terminating a Session

Either the originator or the target may terminate the session. Sessions shall be terminated in either of two ways:

- The originator or target shall close the underlying TCP connection. The corresponding target or originator shall detect the loss of the TCP connection, and shall close its side of the connection;
- The originator or target shall send an UnRegisterSession command (see section 2-4.5 for a description of the UnregisterSession command) and shall wait to detect the closing of the TCP connection. The corresponding target or originator shall then close its side of the TCP connection. The sender of the UnRegisterSession shall detect the loss of the TCP connection, then it shall close its side of the connection.

NOTE: The second method is preferred since it results in more timely clean up of the TCP connection.

2-5.4 Maintaining a Session

Once a session is established, it shall remain established until one of the following occurs:

- the originator or target closes the TCP connection;
- the originator or target issues the UnRegisterSession command;
- the TCP connection is broken.

2-5.5 TCP Behavior (informative)

TCP is a reliable, connection-oriented protocol. If a process at either end of a connection closes its end of the connection, the TCP at the other end is notified immediately. If a message from one process to the other can not be delivered in a reasonable amount of time, the connection is assumed to be broken and an error is returned to the sender on all subsequent sends and receives on the connection.

If an originator process detects that a target has closed its end of a connection or that a connection is broken, it assumes the session with the target is broken and closes its connection to the target. A new session is then established as described above in order to resume communications with the target.

Although an originator process is notified when the other end of a connection has been closed, a broken connection can only be detected when a process actually attempts to send a message over the connection. In most cases, the originator process sends messages to targets frequently enough that a crash of a target machine is detected in a timely manner. Likewise, targets send messages back to originators frequently enough that terminated originator processes and originator machine crashes are detected quickly. However, it is possible that an originator or target may not have any messages to send on a connection for a relatively long period of time.

The TCP protocol supports keep-alive processing. An application can ask TCP to make sure the connection remains working during periods when the application does not have any messages to send. If this feature is enabled, when the connection has been idle for some period of time, TCP will send a keep-alive message to its peer at the other end of the connection. If TCP sends several keep-alive messages and does not receive a reply, TCP assumes the connection has broken and the application is notified just as if it had sent an actual message that timed out.

Most implementations of TCP/IP retry/timeout processing do not declare a failure on a connection until it has remained unusable for several minutes. This is a feature of the TCP protocol on the originator host; turning keep alives does not modify it.

2-6 Common Packet Format

2-6.1 General

The common packet format shall consist of an item count, followed by an address item, then a data item (in that order) as shown below. Additional optional items may follow.

NOTE: The common packet format defines a standard format for protocol packets that are transported with the encapsulation protocol. The common packet format is a general-purpose mechanism designed to accommodate future packet or address types.

Table 2-6.1 Common Packet Format

Field Name	Data Type	Description
Item count	UINT	Number of items to follow (shall be at least 2)
Address item	Item Struct (see below)	Addressing information for encapsulated packet
Data item	Item Struct (see below)	The encapsulated data packet
Optional additional items		

The address and data item structure shall be as follows:

Table 2-6.2 Data and Address Item Format

Field Name	Data Type	Description
Type ID	UINT	Type of item encapsulated
Length	UINT	Length in bytes of the Data Field
Data	Variable	The data (if length >0)

Table 2-6.3 Item ID Numbers

Item ID number	Item type	Description
0x0000	address	Null (used for UCMM messages). Indicates that encapsulation routing is NOT needed. Target is either local (ethernet) or routing info is in a data Item.
0x0001 – 0x000B		Reserved for legacy (RA)
0x000C		ListIdentity response
0x000D – 0x0083		Reserved for legacy (RA)
0x0084 – 0x0090		Reserved for future expansion of this specification (Products compliant with this specification shall not use command codes in this range)
0x0091		Reserved for legacy (RA)
0x0092 – 0x00A0		Reserved for future expansion of this specification (Products compliant with this specification shall not use command codes in this range)
0xA1	address	Connection-based (used for connected messages)
0x00A2 – 0x00A4		Reserved for legacy (RA)
0x00A5 – 0x00B0		Reserved for future expansion of this specification (Products compliant with this specification shall not use command codes in this range)
0x00B1	data	Connected Transport packet
0x00B2	data	Unconnected message
0x00B3 – 0x00FF		Reserved for future expansion of this specification (Products compliant with this

Item ID number	Item type	Description
		specification shall not use command codes in this range)
0x0100		ListServices response
0x0101 – 0x010F		Reserved for legacy (RA)
0x0110 – 0x7FFF		Reserved for future expansion of this specification (Products compliant with this specification shall not use command codes in this range)
0x8000	data	Sockaddr Info, originator-to-target
0x8001	data	Sockaddr Info, target-to-originator
0x8002		Sequenced Address iteme
0x8003 – 0xFFFF		Reserved for future expansion of this specification (Products compliant with this specification shall not use command codes in this range)

2-6.2 Address Items

2-6.2.1 Null Address Item

The null address item shall contain only the type id and the length as shown below. The length shall be zero. No data shall follow the length. Since the null address item contains no routing information, it shall be used when the protocol packet itself contains any necessary routing information. The null address item shall be used for Unconnected Messages.

Table 2-6.4 Null Address Item

Field Name	Data Type	Field Value
Type ID	UINT	0
Length	UINT	0

2-6.2.2 Connected Address Item

This address item shall be used when the encapsulated protocol is connection-oriented. The data shall contain a connection identifier.

NOTE: Connection identifiers are exchanged in the Forward_Open service of the Connection Manager.

Table 2-6.5 Connected Address Item

Field Name	Data Type	Field Value
Type ID	UINT	0xA1
Length	UINT	4
Data	UDINT	Connection Identifier

2-6.2.3 Sequenced Address Item

This address item shall be used for CIP transport class 0 and class 1 connected data. The data shall contain a connection identifier and a sequence number.

Table 2-6.6 Sequenced Address Item

Field Name	Data Type	Field Value
Type ID	UINT	0x8002
Length	UINT	8
Data	UDINT	Connection Identifier
	UDINT	Sequence Number

2-6.3 Data Items

2-6.3.1 Unconnected Data Item

The data item that encapsulates an unconnected message shall be as follows:

Table 2-6.7 Unconnected Data Item

Field Name	Data Type	Field Value
Type ID	UINT	0xB2
Length	UINT	Length, in bytes, of the unconnected message
Data	Variable	The unconnected message

NOTE: The format of the “data” field is dependent on the encapsulated protocol. When used to encapsulate CIP, the format of the “data” field is that of a Message Router request or Message Router reply. See chapter 3 of this specification for details of the encapsulation of UCMM messages. See chapter 2 of the CIP Common specification for the format of the Message Router request and reply packets.

The context field in the encapsulation header shall be used for unconnected request/reply matching.

2-6.3.2 Connected Data Item

The data item that encapsulates a connected transport packet shall be as follows:

Table 2-6.8 Connected Data Item

Field Name	Data Type	Field Value
Type ID	UINT	0xB1
Length	UINT	Length, in bytes, of the transport packet
Data	Variable	The transport packet

NOTE: The format of the "data" field is dependent on the encapsulated protocol. When used to encapsulate CIP, the format of the "data" field is that of connected packet. See chapter 3 of this specification for details of the encapsulation of connected packets. See chapter 3 of the CIP Common specification for the format of connected packets.

2-6.3.3 Sockaddr Info Item

The Sockaddr Info items shall be used to encapsulate socket address information necessary to send datagrams (the connected data) between the target and originator. There are separate items for originator-to-target and target-to-originator socket information.

The Sockaddr Info items shall have the following structure:

Table 2-6.9 Sockaddr Item

Field Name	Data Type	Field Value
Type ID	UINT	0x8000 for O⇒T, 0x8001 for T⇒O
Length	UINT	16 (bytes)
sin_family	INT	shall be AF_INET = 2. This field shall be sent in big endian order.
sin_port	UINT	shall be set to the TCP or UDP port on which packets for this CIP connection will be sent. This field shall be sent in big endian order.
sin_addr	UDINT	shall be set to the IP address to which packet for this CIP connection will be sent. This field shall be sent in big endian order.
sin_zero	ARRAY of USINT	shall be 0. This field shall be sent in big endian order. Length of 8.

NOTE: The structure of the Sockaddr item has been patterned after the sockaddr_in structure from the Winsock specification, version 1.1.

This page is intentionally left blank

Volume 2: EtherNet/IP Adaptation of CIP

Chapter 3: Mapping of Explicit and I/O Messaging to TCP/IP

Contents

3-1 Introduction

This chapter (chapter 3) of the EtherNet/IP specification describes the application of the encapsulation in chapter 2 to the Common Industrial Protocol (CIP). Specifically, this chapter documents the encapsulation of the UCMM and connected packets; extends the format of the path to include IP addresses; and limits which CIP transport parameters can be used in combination.

3-2 CIP Packets over TCP/IP

When the path of a CIP packet traverses an Ethernet-TCP/IP network, the encapsulated packet shall be transmitted using the TCP/IP protocol suite and the encapsulation protocol defined in chapter 2.

3-2.1 Unconnected Messages

UCMM packets shall be transmitted over a TCP/IP connection, using the encapsulation protocol defined in chapter 2. For example, an encapsulated UCMM request shall be formatted as shown in Table 3-2.1.

Table 3-2.1 UCMM Request

Structure	Field Name		Data Type	Field Value
Encapsulation header	Command		UINT	SendRRData (0x6F)
	Length		UINT	Length of command specific data portion
	Session handle		UDINT	Handle returned by RegisterSession
	Status		UDINT	0
	Sender Context		ARRAY of 8 octet	any sender context
	Options		UDINT	0
Command specific data	Interface handle		UDINT	shall be 0 for CIP
	Timeout		UINT	operation timeout
	Encapsulated packet (in the common packet format)	Item count	UINT	This field shall be 2 since one address item and one data item are used.
		Address Type ID	UINT	This field shall be 0 to indicate a UCMM message.
		Address Length	UINT	This field shall be 0 since UCMM messages use the NULL address item.
		Data Type ID	UINT	This field shall be 0x00B2 to encapsulate the UCMM
		Data Length	UINT	Length of the next field in bytes (length of the MR request packet)
		MR request packet	ARRAY of USINT	This field contains a CIP Message Router request packet as defined in Volume 1, Chapter 2.

Likewise, the UCMM reply shall be formatted as shown in Table 3-2.2.

Table 3-2.2 UCMM Reply

Structure	Field Name		Data Type	Field Value
Encapsulation header	Command		UINT	SendRRData (0x6F)
	Length		UINT	Length of command specific data portion
	Session handle		UDINT	Handle returned by RegisterSession
	Status		UDINT	0
	Sender Context		ARRAY of 8 octet	copied from the corresponding UCMM request
	Options		UDINT	0
Command specific data	Interface handle		UDINT	shall be 0 for CIP
	Timeout		UINT	not used for reply
	Encapsulated packet (in the common packet format)	Item count	UINT	This field shall be 2 since one address item and one data item are used.
		Address Type ID	UINT	This field shall be 0 to indicate a UCMM message.
		Address Length	UINT	This field shall be 0 since UCMM messages use the NULL address item.
		Data Type ID	UINT	This field shall be 0x00B2 to encapsulate the UCMM
		Data Length	UINT	Length of the next field in bytes (length of the MR response packet)
		MR response packet	ARRAY of USINT	This field contains a CIP Message Router reply packet as defined in Volume 1, Chapter 2.

3-2.2 CIP Transport Class 0 and Class 1 Connections

NOTE: Please see Volume 1 for the definition and usage of CIP transport class 0 and class 1 connections.

3-2.2.1 CIP transport Class 0 and Class 1 Packets

Packets for transport class 0 and class 1 connections shall be transmitted using UDP and the Common Packet Format defined in Chapter 2. Packets for multicast connections shall be transmitted using IP multicast.

3-2.2.2 Behavior of Class 0 and Class 1 Connections (informative)

Since Ethernet does not have a mechanism for sending scheduled data, several important aspects of class 0 and class 1 behavior are noted as follows:

On Ethernet, it is possible for a CIP transport class 0 or class 1 connected data packet to be lost, for example due to excessive collisions. By definition, class 0 and class 1 connections do not guarantee the delivery of every packet. Rather, producers simply send data at the specified rate (the API). If a packet is lost on a class 0 or class 1 connection, the consumer receives the next packet from the producer.

The degree to which lost packets can be tolerated is application-specific. Ethernet is not suitable for those applications that cannot tolerate any lost packets.

For CIP transport class 1 connections, the consuming CIP transport can detect packet loss by examining the CIP sequence number in the class 1 packet. For class 0 connections, it is not possible for the application to know that a specific packet has been lost since class 0 does not use a sequence number.

The connection timeout mechanism provides feedback to the application when too many packets are lost. The connection timeout is determined by the Requested Packet Interval (RPI) and by the Connection Timeout Multiplier. If a packet is not received in the time specified by the RPI times the Connection Timeout Multiplier, the connection is broken. For example, if the RPI is 50 ms and the Connection Timeout Multiplier is 4, then the connection will time out if a fresh packet is not received in 200 ms (the equivalent of 4 packets being lost). Receipt of older packets (those with lower CIP sequence numbers) will not sustain the CIP connection.

The degree of packet loss for any particular connection will be dependent upon many factors related to the user's Ethernet network configuration. It is beyond the scope of this specification to address this in further detail.

3-2.2.3 No Linkage with TCP Connections

In order to open a CIP transport class 0 or 1 connection, a TCP connection and an EtherNet/IP encapsulation session must first be established. The TCP connection is used to send the Forward_Open service and receive the Forward_Open response. Once the TCP connection is opened, and CIP transport class 0 and 1 connections are established, it is recommended that EtherNet/IP devices leave the TCP connection open. If the TCP connection is left open, it is then available for subsequent communications such as a Forward_Close or other explicit messages.

Although it is recommended that devices leave the TCP connection open, there shall be no linkage between the TCP connection used to open a transport class 0 or class 1 connection and the resulting class 0 or class 1 connection. If a TCP connection closes, the closing of the TCP connection shall not cause the target or the originator to close any corresponding CIP transport class 0 or class 1 connections.

3-2.3 CIP Transport Class 2 and Class 3 Connections

CIP transport class 2 and class 3 connections shall be transmitted over a TCP connection using the encapsulation protocol defined in chapter 2.

Multiple CIP connections may be sent over a single TCP connection. An implementation need not support a specific number of CIP connections per TCP connection. An implementation may impose an upper bound if it chooses.

Because of the full-duplex nature of TCP, the CIP originator to target (O⇒T) and CIP target to originator (T⇒O) link connections shall use the same TCP connection. However if a target subsequently originates a CIP connection, then it shall be considered an originator, and a different TCP connection shall be used.

NOTE: This standard defines no requirements for management of the TCP connection, such as inactivity timeouts, or closing the TCP connection when all native connections are closed. However, implementations are free to implement these.

3-2.4 CIP Transport Classes 4 Through 6

The encapsulation protocol described in chapter 2 shall not be used to encapsulate CIP transport classes 4, 5 and 6.

3-3 Connection Manager Object

3-3.1 Connection Parameters

NOTE: This section documents the Connection Manager parameters that have requirements specific to the TCP/IP encapsulation. Connection Manager parameters are fully described in Volume 1, Chapter 3.

3-3.2 Connection Type

The CIP connection type shall be NULL, MULTICAST, or POINT2POINT. The MULTICAST connection type shall be supported only for CIP transport class 0 and class 1 connections.

3-3.3 Priority

The CIP priority shall be LOW, or HIGH, or SCHEDULED. At present, SCHEDULED priority shall be treated no differently than HIGH priority.

Targets and originators shall close any CIP transport class 2 or 3 connections when the corresponding originating TCP connection is closed.

NOTE: SCHEDULED priority for Ethernet-TCP/IP connections may be further defined in the future.

3-3.4 Trigger Type

The CIP trigger type shall be CYCLIC, CHANGE_OF_STATE, or APPLICATION. CIP transport class 0 and class 1 connections that use CHANGE_OF_STATE triggering shall use the Production Inhibit Time segment (see Volume 1).

3-3.5 Connection Size

The CIP connection size shall be no larger than 65511 bytes.

NOTE: The Forward_Open request limits the connection size to 511 bytes; however, the optional Ex_Forward_Open allows larger connection sizes.

3-3.6 Connection Request Timeout

To reliably establish a CIP connection that extends onto a TCP/IP link, the connection request time-out shall be large enough to allow the connection to be established, which could involve resolving a host name, or going through multiple gateways.

Because of the large variation in connection request processing over TCP/IP, CIP routers in the connection path shall not subtract anything from the connection request timeout.

3-3.7 Connection Path

The link address portion of a TCP/IP connection path segment shall be encoded within a port segment as a string of ASCII characters. The following forms shall all be supported:

- IP address in dot notation, for example “130.151.132.55” (see RFC 1117 for the format of IP addresses);
- IP address in dot notation, followed by a ":" separator, followed by the TCP port number to be used at the specified IP address;
- Host name, for example “plc.controlnet.org”. The host name shall be resolved via a DNS request to a name server (see RFC 1035 for information on host names and name resolution);
- Host name, followed by a ":" separator, followed by the TCP port number to be used at the specified host.

The port number shall be represented in either hex or decimal. Hex shall be indicated by a leading "0x". When a port number is specified, it shall be used rather than the standard port number used for the encapsulation protocol (0xAF12). Only port 0xAF12 is guaranteed to be available in an EtherNet/IP compliant device.

NOTE: Other TCP port numbers may be implemented; however, this specification does not provide a mechanism to determine which TCP port numbers are supported by a device. The use of other TCP port numbers is therefore discouraged. The guaranteed TCP port number, 0xAF12, has been reserved with the Internet Assigned Numbers Authority (IANA) for use by the encapsulation protocol.

Since port segments must be word-aligned, a pad byte may be required at the end of the string. The pad byte shall be 0x00, and shall not be counted in the Optional Address Size field of the port segment.

NOTE: Examples of port segments are shown in Table 3-3.1 (see Volume 1 for the definition of a port segment).

Table 3-3.1 TCP/IP Link Address Examples

Port Segment	IP address	Notes
[12][0D] [31][33][30][2E] [31][35][31][2E] [31][33][32][2E][31][00]	130.151.132.1	Multi-byte address for port 2, 13 byte string plus a pad byte
[13][12] [70][6C][63][2E] [63][6F][6E][74][72][6F][6C][6E][65][74][2E] [6F][72][67]	plc.controlnet.org	Multi-byte address for port 3, 18 byte string, no pad byte
[16][15] [31][33][30][2E] [31][35][31][2E] [31][33][32][2E] [35][35][3A] [30][78][33][32][31][30][00]	130.151.132.55:0x3210	Multi-byte address for port 6, 21 byte string plus a pad byte
[15][17] [70][6C][63][2E] [63][6F][6E][74][72][6F][6C][6E][65][74][2E] [6F][72][67][3A] [39][38][37][36][00]	plc.controlnet.org:9876	Multi-byte address port 5, 23 byte string plus a pad byte

3-3.7.1 Network Connection ID

3-3.7.1.1 General

For EtherNet/IP connections, the Network Connection ID shall be a 32-bit identifier meaningful to the device that chooses it. The Network Connection ID need not be subdivided into any specific fields.

In general, the consuming device selects the Network Connection ID for a point-to-point connection, and the producing device selects the Network Connection ID for a multicast connection. The following table shows which device, Target or Originator, shall choose the T->O and O->T Network Connection IDs:

Table 3-3.2 Network Connection ID Selection

Connection Type	Which Network Connection ID	Who chooses Connection ID
Point-to-point	Originator -> Target	Target
	Target -> Originator	Originator
Multicast	Originator -> Target	Originator
	Target -> Originator	Target

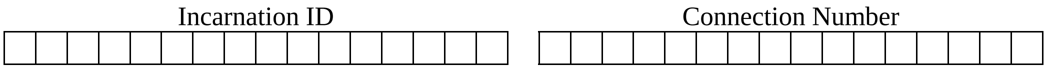
The Network Connection ID shall not be reused until the connection has been closed or has timed out. When a device restarts, it shall not reuse Network Connection IDs from previously opened connections until those connections have been closed or have timed out. A specific connection ID shall not be reused so long as there is the possibility that packets with that connection ID are present in the network.

The following two sections describe possible methods to implement unique Network Connection IDs.

3-3.7.1.2 Using an Incarnation ID (informative)

This section describes one solution that prevents Connection ID reuse when a device restarts. With this solution, the Ethernet device generates Connection IDs for class 0 and class 1 connections with format shown in Figure 3-3.1.

Figure 3-3.1 Connection ID with Incarnation ID



Where:

Connection Number is a 16-bit identifier meaningful to the device choosing the Connection ID.

Incarnation ID is a 16-bit identifier that each device generates before accepting or initiating any connections.

The Incarnation ID persists while the device is powered up and accepting connections. Each successive power-up cycle must cause a new (unique) Incarnation ID to be generated. The following are acceptable methods for generating Incarnation ID's:

Devices may generate a unique Incarnation ID by saving the Incarnation ID in non-volatile storage: when the device powers up, it reads the Incarnation ID from non-volatile storage. This is the Incarnation ID to use for the current cycle. It then increments the Incarnation ID and stores it for the next cycle. Note, however, that non-volatile memory devices generally have a limit to the number of times the device may be written. Depending on the device, it may not be feasible to write the Incarnation ID each powerup.

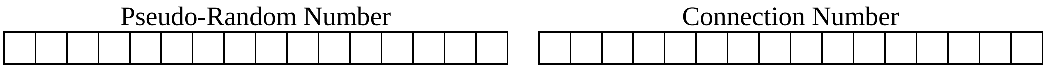
Devices may generate a unique Incarnation ID by generating a pseudo-random number at powerup. This approach requires care. By definition, there is a non-zero probability that the generated Incarnation ID is the same as the previous one. However, if done wisely, the probability is small enough to not be a concern.

Devices such as workstations, because of the large variation in startup time, can safely use the value of the system clock as an Incarnation ID. However, for embedded devices, using the system clock is not reliable since the firmware generally goes through the exact same sequence of instructions at each powerup. This results in the same clock value at the point where it would be selected for the Incarnation ID. For these embedded devices, the Incarnation ID needs to be generated based on random inputs. This is best done using a pseudo-random number generator such as the MD5 algorithm.

3-3.7.1.3 Pseudo-Random Connection ID Per Connection (informative)

This section describes another solution that prevents Connection ID reuse when a device restarts. With this solution, the device generates a pseudo-random Connection ID each time a class 0 or class 1 Connection ID is needed. The Connection ID format for this approach is shown in Figure 3-3.2.

Figure 3-3.2 Pseudo-Random Connection ID



Where:

Connection Number is a 16-bit identifier meaningful to the device choosing the Connection ID.

Pseudo-Random Number is a 16-bit number generated using an appropriate pseudo-random number generator.

With this approach, the device generates the Pseudo-Random Number portion each time a Connection ID is needed. A "strong mixing function" such as the MD5 algorithm [RFC 1321] [RFC 1750] should be used to generate the pseudo-random number. Such functions take multiple input and produce pseudo-random outputs.

In order to prevent Connection IDs from being reused across powerups, the seed values for the inputs to the MD5 algorithm must be unique across successive powerup cycles. The recommended approach is to use the following inputs upon receiving the first incoming connection request:

- Vendor ID, Serial Number, Connection Serial Number
- Contents of the sockaddr_in struct (for the next hop if outbound connection; of the sender if inbound connection)
- Value of system clock

NOTE: This assumes the Ethernet device is a bridge or the Target of the connection and would not be applicable if the device is the connection Originator. For connection originators, the above seed values would likely be the same across successive powerups. Connection originators must use another source for initial seed values, or else use the Incarnation ID approach.

By definition, there is a non-zero probability that a Connection ID conflict may still occur. However, the probability is lowered by:

Using a robust pseudo-random number generator such as the MD5 algorithm.

Ensuring the seed values are different on successive power-up cycles.

3-3.8 Forward_Open for CIP Transport Class 2 and Class 3 Connections

The Forward_Open service for CIP class 2 and class 3 connections shall be sent over a TCP connection using the SendRRData command defined in chapter 2.

3-3.9 Forward_Open for CIP Transport Class 0 and Class 1 Connections

3-3.9.1 General

The Forward_Open service for CIP transport class 0 and class 1 connections shall be sent over a TCP connection using the SendRRData command defined in chapter 2. As part of the Forward_Open dialog, the producer and consumer shall exchange the UDP port numbers and IP multicast address (for multicast connections) necessary to send the CIP transport class 0 and class 1 connected data. The Sockaddr Info item defined in Chapter 2 shall be used to encode the UDP port numbers and IP multicast address. The use of the Sockaddr Info item varies depending on whether the connection is multicast or point-to-point, and whether the connection originator or the connection target is the producer.

For multicast connections, the producer shall choose an IP multicast address to which to send the connected data. The port number shall be the registered UDP port number (0x08AE) assigned by the IANA. The IP multicast address and UDP port number shall be encoded via a Sockaddr Info item. A Sockaddr Info item shall be sent with Forward_Open (if the connection originator is the producer), or with the Forward_Open_reply (if the connection target is the producer).

For point-to-point connections, the consumer shall choose a UDP port number to which the connected data shall be sent. The port number may be the registered port number (0x08AE) or may be a port number chosen by the consumer. The port number shall be encoded in a Sockaddr Info item and shall be sent with the Forward_Open (if the connection originator is the consumer), or with the Forward_Open_reply (if the connection target is the consumer).

The Sockaddr Info item(s) shall be placed after the Forward_Open and/or Forward_Open_reply data in the SendRRData command/reply. If the Sockaddr Info items are not present, or are in error, a Forward_Open_reply shall be returned with status code 0x01 and extended status 0x205.

3-3.9.2 Mapping Connections to IP Multicast Addresses

NOTE: It is recommended, though not required, that producers use a unique IP multicast address for each active multicast connection. Depending upon the implementation, this can reduce the amount of connection screening on the part of the consumer. It also allows the consumer to more evenly service incoming connected data from multiple connections.

Since a unique IP multicast address per multicast connection is not required, consumers shall be able to handle the situation in which packets from multiple multicast connections are being sent to the same IP multicast address. Consumers shall be able to screen the incoming packets based on the Connection ID and source IP address.

NOTE: Requirements for screening connected data are defined in section 3-4.3.

3-3.9.3 Completing the Multicast Connection (informative)

After receiving the Forward_Open_reply the consuming Ethernet devices should join the desired IP Multicast Group in order to receive the IP Multicast datagrams. The exact method for doing this depends on the TCP/IP application programming interface in use on the device.

3-4 CIP Transport Class 0 and Class 1 Connected Data

3-4.1 UDP Datagrams

CIP transport class 0 and class 1 packets shall be sent in UDP datagrams using the Common Packet Format defined in chapter 2. The data portion of the UDP datagram for CIP transport class 0 and class 1 packets shall be as shown in Table 3-4.1.

Table 3-4.1 UDP Data Format for Class 0 and Class 1

Field Name	Type	Value
Item Count	UINT	2
Type ID	UINT	0x8002 (Sequenced Address Type)
Length	UINT	8
Address Data	UDINT	Connection ID (from Forward_Open reply)
	UDINT	Sequence Number
Type ID	UINT	0x00B1 (Connected Data Type)
Length	UINT	Number of bytes in packet to follow
Data		class 0 or class 1 packet

3-4.2 CIP Transport Class 0 and Class 1 Packet Ordering

NOTE: By definition, CIP class 0 and class 1 transports do not detect out-of-order packets. For class 0, every packet is considered to be new data. For class 1, only duplicate data is detected. If a new packet arrives and the sequence number in the transport packet is different from the previous packet, then the new packet is considered to be new data even if the new packet has a sequence number that is less than the previous sequence number.

NOTE: When using UDP to transport CIP class 0 and class 1 connected data, there is no guarantee that packets arrive in the same order that they were sent. When both sender and receiver are on the same subnet, packets typically arrive in order. However, when going through routers, when there are multiple paths that a packet could take, it is possible for packets to arrive out of order.

For class 0 and class 1 connections over Ethernet, devices shall maintain a sequence number in the UDP payload defined in section 3-4.1. The sequence number shall be maintained per connection. Each time an Ethernet device sends a CIP class 1 packet, it shall increment the sequence number for that connection. If the receiving Ethernet device receives a packet whose sequence number is less than the previously received packet, the packet with the smaller sequence number shall be discarded. Duplicate packets shall be accepted and given to the transport layer.

The sequence number shall be operated on with modular arithmetic to deal with sequence rollover.

NOTE: Dealing with 32-bit sequence numbers is described in RFC793 (the TCP definition), as follows:

It is essential to remember that the actual sequence number space is finite, though very large. This space ranges from 0 to 2**32 - 1. Since the space is finite, all arithmetic dealing with sequence numbers must be performed modulo 2**32. This unsigned arithmetic preserves the relationship of sequence numbers as they cycle from 2**32 - 1 to 0 again. There are some subtleties to computer modulo arithmetic, so great care should be taken in programming the comparison of such values. The symbol "=<" means "less than or equal" (modulo 2**32).

Example macros show how this may be done:

```
/*
 * TCP sequence numbers are unsigned 32 bit integers operated
 * on with modular arithmetic.  These macros can be
 * used to compare such integers.
 */

#define SEQ_LT(a,b)     ((int)((a)-(b)) < 0)
#define SEQ_LEQ(a,b)    ((int)((a)-(b)) <= 0)
#define SEQ_GT(a,b)     ((int)((a)-(b)) > 0)
#define SEQ_GEQ(a,b)    ((int)((a)-(b)) >= 0)
```

3-4.3 Screening Incoming Connected Data

Ethernet devices that receive class 0 and class 1 connected data shall screen incoming packets based on the network connection ID and IP address of the sending device. This is necessary for the following reasons:

- For multicast connections, there is no guaranteed mechanism to prevent multiple devices from using the same IP multicast address. Consequently, a device could receive (bogus) multicast connected data from a device with which it has not established a connection.
- For multicast connections, a device is allowed to use the same IP multicast address for multiple class 0 and class 1 multicast connections.
- To prevent network connection ID conflicts.

When a class 0 or class 1 connection is established, the target and originating Ethernet devices shall record the network connection ID on which they will receive connected data, coupled with the IP address of the device at the other end of the connection. When a device receives connected data, it shall confirm that the network connection ID is valid for the IP address of the sending device. If not, the packet shall be discarded.

3-5 IP Multicast Scoping and Address Allocation

3-5.1 Background (informative)

3-5.1.1 General

Two issues related to IP multicast must be considered when implementing EtherNet/IP multicast connections: IP multicast scoping and IP multicast address allocation.

IP multicast scoping refers to the practice of limiting how widely a given multicast datagram is propagated across the network. IP multicast address allocation refers to the problem of how applications select IP multicast addresses that are used to send and receive IP multicast datagrams.

The following subsections on multicast scoping and allocation practices are informative, and are intended to set the general context for considering the issues of scoping and address allocation. Specific requirements for EtherNet/IP devices follow in subsequent sections.

3-5.1.2 IP Multicast Scoping Practices

In general, most currently deployed networks use the practice of "TTL scoping" in conjunction with router and/or switch configuration to confine multicast traffic to desired network boundaries.

TTL scoping refers to the practice of using the "Time to Live" (TTL) field in the IP header to limit the number of network hops over which the multicast packet is propagated. When sending an IP multicast datagram, a host can set the TTL field in the IP header to an appropriate value based on how widely the datagram should be propagated. As the datagram is routed through the network, each hop decrements the TTL field. Routers can be configured with TTL thresholds such that they will not forward a packet unless the TTL is greater than the threshold.

Note that a multicast datagram with an initial TTL of 1 limits the datagram to the local subnet. Other common TTL values are 16 for multicast within a site and 64 for multicast within a region.

In addition to TTL scoping, multicast routing protocols and other methods are commonly used to control the propagation of multicast traffic. Routers commonly support multicast protocols such as PIM, DVMRP, etc. Switches that implement "IGMP snooping" can limit the multicast packets sent on a port to only those multicast addresses for which the end device has issued an IGMP membership message. Configuration of switches and routers is usually done by knowledgeable staff.

3-5.1.3 IP Multicast Address Allocation Practices

The entire IP multicast address space is 224.0.0.0 through 239.255.255.255. The Internet Assigned Numbers Authority (www.iana.org) is responsible for allocation of the IP multicast address space. IP multicast addresses have been assigned to particular organizations, and for particular protocols. In addition there is a large block of IP multicast addresses allocated for "administratively scoped" multicast, from which applications may allocate addresses, and for which a suite of allocation and scoping protocols are being developed by the Internet Engineering Task Force (IETF). The administratively scoped range is from 239.0.0.0 through 239.255.255.255 (and is further partitioned into additional ranges).

Unfortunately at present there are no widely deployed standard mechanisms for allocating and assigning multicast addresses to applications. For example, when a network administrator deploys a video streaming application, the application will have its own specific mechanism for assigning IP multicast addresses.

3-5.2 Multicast Scoping for EtherNet/IP

By default, EtherNet/IP devices shall use a TTL equal to 1 for transport class 0 and 1 multicast packets. The use of a TTL value of 1 prevents multicast packets from propagating beyond the local subnet. When TTL is equal to 1, both the EtherNet/IP producer and consumer must be on the same subnet.

EtherNet/IP devices are strongly encouraged to support the explicit configuration of the TTL value for IP multicast packets. If supported, devices shall use the TCP/IP Interface Object (class 0xF5) as the mechanism to configure the TTL value. When a TTL value greater than 1 is configured, then the producer and consumer may be on different subnets.

If the TTL value has not been configured to be greater than 1 and if a multicast connection request is received from an originator on a different subnet, then the device shall return General Status 0x01 and Extended Status 0x813 in the Forward_Open Reply.

When the TTL value is explicitly configured, it shall be used for all EtherNet/IP multicast packets.

3-5.3 Multicast Address Allocation for EtherNet/IP

EtherNet/IP defines two mechanisms for allocation of IP multicast addresses used for EtherNet/IP multicast packets: using an algorithm based on the device's IP address, and explicit configuration via the TCP/IP Interface Object. EtherNet/IP devices shall implement the algorithm-based method by default. Devices are also strongly encouraged to implement the method for explicit configuration of multicast addresses. Both methods are described below:

1. Allocation algorithm based on the device's IP address:

   The overall IP multicast address range shall be the Organizational Local Scope, and shall start at 239.192.1.0. Each device shall use a block of (at most) 32 multicast addresses from this range. Each device shall calculate the block of multicast addresses via an algorithm, described further below, that uses the Host Id portion of the device's IP address.

   A device's Host Id shall be determined by applying the subnet mask to the device's IP address. If a subnet mask is configured for the device, the subnet mask shall be applied to the IP address to determine the Host Id. If no subnet mask is in use, then the class of the device's IP address shall be used to determine the Host Id (e.g., for Class C addresses 8 bits of Host Id shall be used, for Class B addresses 16 bits of host id shall be used, and for Class A addresses 24 bits of Host Id shall be used).

   In order to keep the IP multicast addresses within the IPv4 Organization Local Scope, and to put a reasonable bounds on the number of multicast addresses in use, devices shall use at most the low-order 10 bits of the Host Id in generating the range of multicast addresses. This allows for 1024 unique Host Ids.

   The following pseudo-code shows the algorithm to determine the device's starting and ending multicast addresses:

   ```
   CIP_Mcast_Base_Addr = 0xEFC00100 // 239.192.1.0 is the starting address
   CIP_Host_Mask = 0x3FF            // 10 bits of host id
   ```

```
if Subnet_mask configured then
   Netmask = Subnet_mask
else
   if IP_address is Class A then
      Netmask = 255.0.0.0
   else if IP_address is Class B then
      Netmask = 255.255.0.0
   else if IP_address is Class C then
      Netmask = 255.255.255.0
end_else

Host_id = IP_addr & (~Netmask)
Mcast_index = Host_id - 1
Mcast_index = Mcast_index & (CIP_Host_Mask)

Mcast_start_addr = CIP_Mcast_Base_Addr + (Mcast_index * 32)
Mcast_end_addr = Mcast_start_addr + 31
```

The following table shows example multicast assignments.

Subnet mask	IP address	Multicast addresses
None configured (8 bits of Host Id used, since 192.168.x.x is Class C)	192.168.1.1	239.192.1.0 - 239.192.1.31
	192.168.1.2	239.192.1.32 - 239.192.1.63
	192.168.1.3	239.192.1.64 - 239.192.1.95
255.255.248.0 (11 bits of Host Id)	10.10.16.1	239.192.1.0 - 239.192.1.31
	10.10.16.2	239.192.1.32 - 239.192.1.63
	10.10.16.3	239.192.1.64 - 239.192.1.95
	10.10.20.0 (Host Id = 1024; lower 10 bits all 0)	239.192.128.224 - 239.192.128.255 (this is the highest multicast address range that would result using the algorithm)

Since there are a finite number of unique Host Ids, it is possible for different devices to produce data using the same multicast address. Consequently, devices that receive packets on EtherNet/IP multicast connections shall screen the incoming packets based on the CIP Connection Id as well as the IP address of the sending device. This is described in Chapter 3-4.3.

Note that when the device's IP address or subnet mask changes, the IP multicast addresses generated by the algorithm also shall change accordingly.

2. Explicit configuration of IP multicast addresses:

   IP multicast addresses are configured via the TCP/IP Interface Object (class 0xF5). The user (or software) can configure a starting multicast address and number of multicast addresses to allocate. The configured multicast addresses shall then be used for EtherNet/IP multicast packets.

EtherNet/IP devices shall at least implement the algorithmic method for allocating IP multicast addresses, and are encouraged to implement both methods. If both methods are implemented, the algorithmic method shall be the default "out of box" method.

3-5.4 User Considerations (informative)

This section is informational, and is meant to be an aid to vendors and users in the practical deployment of EtherNet/IP applications. When deploying an EtherNet/IP system that uses multicast connections, the user should consider a number of aspects in order to achieve satisfactory application performance.

1. When devices allocate IP multicast addresses according to the default algorithm in section 3-6.3, the assignment of IP addresses to devices affects the way that IP multicast addresses are selected. Users should be aware that only 10 bits the IP address are used to generate the Host Id, which in turn determines the device's range of IP addresses. If the user's subnet mask is larger than 10 bits, there is the potential for multiple devices to use the same IP multicast address when producing data. While this does not result in incorrect operation, it can result in devices experiencing performance degradation due to the receipt of additional multicast packets that must be discarded.

2. When multicast addresses are explicitly configured, care should be taken so that devices in the same subnet have unique blocks of multicast addresses. Further, if multicast connections will cross subnet boundaries, then care must be taken to ensure that all devices in the network have unique blocks of multicast addresses. When configuring multicast addresses, it is recommended that addresses from the IPv4 Local Scope be used (239.255.0.0 – 239.255.255.255) so as not to conflict with multicast address that may be generated algorithmically.

3. Some routers experience performance degradation when they must handle many multicast packets with TTL equal to 1. In such situations, users may configure TTL to be greater than 1 even though I/O connections do not need to cross subnets. When setting TTL greater than 1, it is recommended that users also configure multicast addresses for each device. If multicast addresses are not explicitly configured, they are generated according to the algorithm in the specification. Care must be taken in this situation since devices in different subnets could generate the same IP multicast addresses. The multicast packets sent from one subnet would then be received on the other subnet, possibly impacting performance. In order to prevent unwanted multicast propagation, the user must perform additional router configuration to constrain the EtherNet/IP multicast packets to the subnet on which they originate. There are several techniques for constraining multicast at the router. Router configuration is beyond the scope of this specification.

4. Users are strongly recommended to use switches that implement IGMP snooping. When IGMP snooping is used, devices will only receive the multicast packets in which they are interested (i.e., for which they have issued an IGMP membership message).

3-5.5 Future Directions for EtherNet/IP (informative)

There are a number of Internet standards regarding IP multicast allocation and scoping. While these standards have not yet been widely deployed, they are expected to have an impact on future EtherNet/IP mechanisms for using IP multicast. Three RFC's that seem most relevant to EtherNet/IP are listed below:

- "The Internet Multicast Address Allocation Architecture", RFC 2908
- "Administratively Scoped Multicast", RFC 2365
- "Multicast Address Dynamic Client Allocation Protocol (MADCAP)", RFC 2730

3-6 IGMP Usage

3-6.1 Background (informative)

The Internet Group Management Protocol (IGMP) is a standard protocol used by hosts to report their IP multicast group memberships and must be implemented by any host that wishes to receive IP multicast datagrams. IGMP messages are used by multicast routers to learn which multicast groups have members on their attached networks. IGMP messages are also used by switches capable of supporting "IGMP snooping" whereby the switch listens to IGMP messages and only sends the multicast packets to ports that have joined the multicast group.

There are two versions of IGMP:

- IGMP V1 is defined in RFC1112. It defines two messages: Host Membership Query and Host Membership Report (commonly referred to as a "join")
- IGMP V2 is defined in RFC2236. It defines additional messages and behavior, notably the Leave Group message.

IGMP V2 is backward compatible with V1. RFC2236 discusses the interaction between IGMP V1 and V2 hosts and routers.

Since EtherNet/IP devices make extensive use of IP multicast for CIP transport class 0 and 1 connections, consistent IGMP usage by EtherNet/IP devices is essential in order to create well-functioning EtherNet/IP application networks.

3-6.2 IGMP Membership Report Messages

EtherNet/IP devices shall issue a Membership Report message when opening a CIP connection on which they will receive multicast packets. Specifically, devices shall adhere to the following behavior:

1. When the T->O Connection Type is multicast (originator is multicast consumer), the originator shall issue a Membership Report upon receipt of a successful Forward_Open_reply. The Membership Report shall include the IP multicast address as communicated in the Forward_Open_reply.
2. When the O->T Connection Type is multicast (target is multicast consumer), the target shall issue a Membership Report upon sending a successful Forward_Open_reply. The Membership Report shall include the IP multicast address as communicated in the Forward_Open.

If the device has already issued a Membership Report for the IP multicast address (e.g., if the multicast address is being used with an existing connection) the device may, but is not required to, issue another Membership Report.

Devices shall also send Membership Report messages in response to Membership Query messages, per the IGMP RFCs.

3-6.3 IGMP Leave Group messages

Devices that support IGMP V2 shall issue a Leave Group when all the CIP connections associated with a consuming IP multicast address have either closed or timed out. Specifically, devices shall adhere to the following behavior:

1. When the T->O Connection Type is multicast (originator is multicast consumer), the originator shall issue a Leave Group upon receipt of a successful Forward_Close reply if the originator has no other open connections consuming on that IP multicast address.
2. When the O->T Connection Type is multicast (target is multicast consumer), the target shall issue a Leave Group upon sending a successful Forward_Close reply if the target has no other open connections consuming on that IP multicast address.
3. In the event of a connection timeout, the multicast consumer (whether target or originator) shall issue a Leave Group message if the multicast consumer has no other connections consuming on that IP multicast address.

This page is intentionally left blank

Volume 2: EtherNet/IP Adaptation of CIP

Chapter 4: Object Model

Contents

4-1 Introduction

This chapter of the EtherNet/IP specification contains additions to the CIP object model that are EtherNet/IP specific. At this time, no such additions exist.

This page is intentionally left blank

Volume 2: EtherNet/IP Adaptation of CIP

Chapter 5: Object Library

Contents

5-1 Introduction

In this standard, object modeling is used to represent the network visible behavior of devices. Devices are modeled as a collection of objects. Each class of objects is a collection of related services, attributes and behaviors. Services are the procedures that an object performs. Attributes are characteristics of objects represented by values, which can vary. An object’s behavior is an indication of how the object responds to particular events.

This chapter of the specification contains the object descriptions specific to EtherNet/IP. The rest of the object descriptions can be found in the Volume 1, Chapter 5. With respect to the OSI reference model, CIP objects perform the Layer 7 Application functions. They also provide a mechanism to access station management counters via the network.

5-2 Reserved Class Codes

The rest of the class codes are defined in Volume 1 of the CIP Networks Library.

5-3 TCP/IP Interface Object

Class Code: F5 Hex

5-3.1 Scope

The TCP/IP Interface Object provides the mechanism to configure a device’s TCP/IP network interface. Examples of configurable items include the device’s IP Address, Network Mask, and Gateway Address.

The underlying physical communications interface associated with the TCP/IP Interface Object shall be any interface that supports the TCP/IP protocol. For example, a TCP/IP Interface Object may be associated any of the following: an Ethernet 802.3 interface, an ATM interface, a serial port running SLIP, a serial port running PPP, etc. The TCP/IP Interface Object provides an attribute that identifies the link-specific object for the associated physical communications interface. The link-specific object is generally expected to provide link-specific counters as well as any link-specific configuration attributes.

Each device shall support exactly one instance of the TCP/IP Interface Object for each TCP/IP-capable communications interface on the module.

5-3.2 Attributes

5-3.2.1 Class Attributes

Table 5-3.1 Class Attributes

Attribute ID	Need in Implementation	Access Rule	Name	Data Type	Description of Attribute	Semantics of values
2	Conditional [1]	Get	Max Instance	UINT	Maximum instance number of an object currently created in this class level of the device.	The largest instance number of a created object at this class hierarchy level.
3	Conditional [1]	Get	Number of Instances	UINT	Number of object instances currently created at this class level of the device.	The number of object instances at this class hierarchy level
1, 4 thru 7	These class attributes are optional and are described in Volume 1, Chapter 4.					
1	Required if the number of instances is greater than 1.					

5-3.2.2 Instance Attributes

Table 5-3.2 Instance Attributes

Attr ID	Need In Implem	Access Rule	Name	Data Type	Description of Attribute	Semantics of Values
1	Required	Get	Status	DWORD	Interface status	See section 5-3.2.2.1.
2	Required	Get	Configuration Capability	DWORD	Interface capability flags	Bit map of capability flags. See section 5-3.2.2.2.
3	Required	Set	Configuration Control	DWORD	Interface control flags	Bit map of control flags. See section 5-3.2.2.3

Attr ID	Need In Implem	Access Rule	Name	Data Type	Description of Attribute	Semantics of Values
4	Required	Get	Physical Link Object	STRUCT of:	Path to physical link object	See section 5-3.2.2.4
			Path size	UINT	Size of Path	Number of 16 bit words in Path
			Path	Padded EPATH	Logical segments identifying the physical link object	The path is restricted to one logical class segment and one logical instance segment. The maximum size is 12 bytes. See Appendix C of Volume 1, Logical Segments.
5	Required	Set	Interface Configuration	STRUCT of:	TCP/IP network interface configuration.	See section 5-3.2.2.5
			IP Address	UDINT	The device's IP address.	Value of 0 indicates no IP address has been configured. Otherwise, the IP address shall be set to a valid Class A, B, or C address and shall not be set to the loopback address (127.0.0.1).
			Network Mask	UDINT	The device's network mask	Value of 0 indicates no network mask address has been configured.
			Gateway Address	UDINT	Default gateway address	Value of 0 indicates no IP address has been configured. Otherwise, the IP address shall be set to a valid Class A, B, or C address and shall not be set to the loopback address (127.0.0.1).
			Name Server	UDINT	Primary name server	Value of 0 indicates no name server address has been configured. Otherwise, the name server address shall be set to a valid Class A, B, or C address.
			Name Server 2	UDINT	Secondary name server	Value of 0 indicates no secondary name server address has been configured. Otherwise, the name server address shall be set to a valid Class A, B, or C address.
			Domain Name	STRING	Default domain name	ASCII characters. Maximum length is 48 characters. Shall be padded to an even number of characters (pad not included in length). A length of 0 shall indicate no Domain Name is configured.

Attr ID	Need In Implem	Access Rule	Name	Data Type	Description of Attribute	Semantics of Values
6	Required	Set (Optional. See section 5-3.2.2.6)	Host Name	STRING	Host name	ASCII characters. Maximum length is 64 characters. Shall be padded to an even number of characters (pad not included in length). A length of 0 shall indicate no Host Name is configured. See section 5-3.2.2.6.
7	Conditional[1]		Safety Network Number	6 octets	See CIP Safety Specfication, Volume 5, Chapter 3	
8	Conditional[2]	Get Set is conditional[3]	TTL Value	USINT	TTL value for EtherNet/IP multicast packets	Time-to-Live value for IP multicast packets. Default value is 1. Minimum is 1; maximum is 255 See Chapter 5-3.2.2.7.
9	Conditional[2]	Get Set is conditional[3]	Mcast Config	STRUCT of:	IP multicast address configuration	See Chapter 5-3.2.2.8.
			Alloc Control	USINT	Multicast address allocation control word. Determines how addresses are allocated.	See Chapter 5-3.2.2.8 for details. Determines whether multicast addresses are generated via algorithm or are explicitly set.
			Reserved	USINT	Reserved for future use	Shall be 0.
			Num Mcast	UINT	Number of IP multicast addresses to allocate for EtherNet/IP	The number of IP multicast addresses allocated, starting at “Mcast Start Addr”. Maximum value is device specific, however shall not exceed the number of EtherNet/IP multicast connections supported by the device.
			Mcast Start Addr	UDINT	Starting multicast address from which to begin allocation.	IP multicast address (Class D). A block of “Num Mcast” addresses is allocated starting with this address.

1 – This attribute is required for EtherNet/IP safety devices. Non-safety devices shall not implement this attribute.

2 – If either TTL Value or Mcast Config is implemented, both must be implemented.

3 – If either TTL Value or Mcast Config is implemented as settable, both must be implemented as settable.

5-3.2.2.1 Status Instance Attribute

The **Status** attribute is a bitmap that shall indicate the status of the TCP/IP network interface. Refer to the state diagram in section 5-3.4, Behavior, for a description of object states as they relate to the Status attribute.

Table 5-3.3 Status Attribute

Bit(s):	Called:	Definition	
0-3	Interface Configuration Status	Indicates the status of the Interface Configuration attribute.	0 = The Interface Configuration attribute has not been configured. 1 = The Interface Configuration attribute contains valid configuration obtained from BOOTP, DHCP or non-volatile storage. 2 = The Interface Configuration attribute contains valid configuration, obtained from hardware settings (e.g.: pushwheel, thumbwheel, etc.) 3-15 = Reserved for future use.
4	Mcast Pending	Indicates a pending configuration change in the TTL Value and/or Mcast Config attributes. This bit shall be set when either the TTL Value or Mcast Config attribute is set, and shall be cleared the next time the device starts.	
5-31	Reserved	Reserved for future use and shall be set to zero.	

5-3.2.2.2 Configuration Capability Instance Attribute

The **Configuration Capability** attribute is a bitmap that indicates the device's support for optional network configuration capability. Devices are not required to support any one particular item, however must support at least one method of obtaining an initial IP address.

Table 5-3.4 Configuration Capability Attribute

Bit(s):	Called:	Definition
0	BOOTP Client	1 (TRUE) shall indicate the device is capable of obtaining its network configuration via BOOTP.
1	DNS Client	1 (TRUE) shall indicate the device is capable of resolving host names by querying a DNS server.
2	DHCP Client	1 (TRUE) shall indicate the device is capable of obtaining its network configuration via DHCP.
3	DHCP-DNS Update	1 (TRUE) shall indicate the device is capable of sending its host name in the DHCP request as documented in Internet draft <draft-ietf-dhc-dhcp-dns-12.txt>.
4	Configuration Settable	1 (TRUE) shall indicate the Interface Configuration attribute is settable. Some devices, for example a PC or workstation, may not allow the Interface Configuration to be set via the TCP/IP Interface Object.
5-31	Reserved	Reserved for future use and shall be set to zero.

5-3.2.2.3 Configuration Control Instance Attribute

The **Configuration Control** attribute is a bitmap used to control network configuration options.

Table 5-3.5 Configuration Control Attribute

Bit(s):	Called:	Definition	
0-3	Startup Configuration	Determines how the device shall obtain its initial configuration at start up.	0 = The device shall use the interface configuration values previously stored (for example, in non-volatile memory or via hardware switches, etc). 1 = The device shall obtain its interface configuration values via BOOTP. 2 = The device shall obtain its interface configuration values via DHCP upon start-up. 3-15 = Reserved for future use.
4	DNS Enable	If 1 (TRUE), the device shall resolve host names by querying a DNS server.	
5-31	Reserved	Reserved for future use and shall be set to zero.	

Devices are not required to support any of the particular values of the Startup Configuration bits, however a device must support at least one method of obtaining an initial TCP/IP interface configuration.

Some devices, in particular low-end devices, may choose to obtain network interface configuration via BOOTP or DHCP only. The use of BOOTP and/or DHCP may not be appropriate for all devices. DHCP in particular supports dynamically allocated IP addresses, which could result in a device getting a different IP address each time it powers up. This behavior is not appropriate for devices that need to have static IP addresses.

A device may obtain its initial IP address via BOOTP or DHCP and then have that address retained in non-volatile storage. After receiving an IP address via BOOTP or DHCP, the address can be retained by setting the Startup Configuration bits to 0.

Out of the box, a device may wish to obtain its initial configuration via some method other than BOOTP or DHCP. For example, the device may wish to obtain its initial configuration over an attached serial port. In this case, the device should have its Startup Configuration bits set to 0, and its Interface Configuration attribute fields to all 0s. The device should then wait to be configured.

Once a device is up and running, when the value of the Startup Configuration bits is 0, a request to set the Interface Configuration attribute shall cause the device to store the contents of the Interface Configuration attribute in non-volatile storage if supported by the device. The Startup Configuration bits shall not be set to 0 unless the Interface Configuration attribute minimally contains a valid IP address. Otherwise the device could be rendered unable to communicate on the network.

Additional standard configuration methods are may be adopted in the future as they are developed and accepted by the Internet community. Non-standard techniques, including various forms of "IP Gleaning," which rely upon arrival of unusual sequences of messages shall not be used for configuration of EtherNet/IP nodes.

5-3.2.2.4 Physical Link Object

This attribute identifies the object associated with the underlying physical communications interface (e.g., an 802.3 interface). There are two components to the attribute: a Path Size (in UINTs) and a Path. The Path shall contain a Logical Segment, type Class, and a Logical Segment, type Instance that identifies the physical link object. The maximum Path Size is 6 (assuming a 32 bit logical segment for each of the class and instance).

The physical link object itself typically maintains link-specific counters as well as any link-specific configuration attributes. If the CIP port associated with the TCP/IP Interface Object has an Ethernet physical layer, this attribute shall point to an instance of the Ethernet Link Object (class code = 0xF6). When there are multiple physical interfaces that correspond to the TCP/IP interface, this attribute shall either contain a Path Size of 0, or shall contain a path to the object representing an internal communications interface (often used in the case of an embedded switch).

For example, the path could be as follows:

Table 5-3.6 Example Path

Path	Meaning
[20][F6][24][01]	[20] = 8 bit class segment type; [F6] = Ethernet Link Object class; [24] = 8 bit instance segment type; [01] = instance 1.

5-3.2.2.5 Interface Configuration

This attribute contains the configuration parameters required to operate as a TCP/IP node. In order to prevent incomplete or incompatible configuration, the parameters making up the Interface Configuration attribute cannot be set individually. To modify the Interface Configuration attribute, the user should first Get the Interface Configuration attribute, change the desired parameters then set the attribute.

The TCP/IP Interface Object shall apply the new configuration upon completion of the Set service. If the value of the Startup Configuration bits (Configuration Control attribute) is 0, the new configuration shall be stored in non-volatile memory. The device shall not reply to the set service until the values are safely stored to non-volatile storage. An attempt to set any of the components of the Interface Configuration attribute to invalid values (see Semantics of Values in Table 5-3.2) shall result in an error (status code 0x09) returned from the Set service.

If initial configuration is to be obtained via BOOTP or DHCP, the Interface Configuration attribute components shall be all zeros until the BOOTP or DHCP reply is received. Upon receipt of the BOOTP or DHCP reply, the Interface Configuration attribute shall show the configuration obtained via BOOTP/DHCP.

Devices are not required to support the Set service. Some implementations, for example those running on a PC or Workstation, need not support setting the network interface configuration via the TCP/IP Interface Object.

Components of the interface configuration attributes are described below:

Table 5-3.7 Interface Configuration Attribute

Name	Meaning
IP address	The device's IP address.
Network mask	The device's network mask. The network mask is used when the IP network has been partitioned into subnets. The network mask is used to determine whether an IP address is located on another subnet.
Gateway address	The IP address of the device's default gateway. When a destination IP address is on a different subnet, packets are forwarded to the default gateway for routing to the destination subnet.
Name server	The IP address of the primary name server. The name server is used to resolve host names. For example, that might be contained in a CIP connection path.
Name server 2	The IP address of the secondary name server. The secondary name server is used when the primary name server is not available, or is unable to resolve a host name.
Domain name	The default domain name. The default domain name is used when resolving host names that are not fully qualified. For example, if the default domain name is "odva.org", and the device needs to resolve a host name of "plc", then the device will attempt to resolve the host name as "plc.odva.org".

For additional information on IP addressing, subnetworks, gateways, etc. refer to Comer, Douglas E.*; Internetworking with TCP/IP, Volume 1: Protocols and Architecture*; Englewood Cliffs, NJ; Prentice-Hall, 1990.

5-3.2.2.6 Host Name

The **Host Name** attribute contains the device's host name. The host name attribute is used when the device supports the DHCP-DNS Update capability and has been configured to use DHCP upon start up. The DHCP-DNS Update mechanism is specified Internet draft <draft-ietf-dhc-dhcp-dns-12.txt>, and is supported in Windows 2000. The mechanism allows the DHCP client to transmit its host name to the DHCP server. The DHCP server then updates the DNS records on behalf of the client.

The host name attribute does not need to be set for the device to operate normally. The value of the Host Name attribute, if it is configured, shall be used for the value of the FQDN option in the DHCP request. If the Host Name attribute has not been configured then the device shall not include the FQDN option in the DHCP request.

For devices that do not support the DHCP-DNS capability, or are not configured to use DHCP, then the host name can be used for informational purposes.

The set access is optional when the Interface Configuration attribute is not settable. Some devices, for example, a PC or workstation, may not allow the Interface Configuration or Host name to be set via the TCP/IP interface object. If the set is not implemented, an "Attribute not settable" (0x0E) error shall be returned in response to a "Set attribute single" request.

5-3.2.2.7 TTL Value

TTL Value is value the device shall use for the IP header Time-to-Live field when sending EtherNet/IP packets via IP multicast. By default, TTL Value shall be 1. The maximum value for TTL is 255. Note that unicast packets shall use the TTL as configured for the TCP/IP stack, and not the TTL Value configured in this attribute.

When set, the TTL Value attribute shall be saved in non-volatile memory, and shall take affect the next time the device starts. Setting the TTL Value attribute shall also cause the Mcast Pending bit in the Interface Status attribute to be set, indicating that there is pending configuration. When a new TTL Value is pending, Get_Attribute_Single or Get_Attributes_All requests shall return the pending value. The Mcast Pending bit shall be cleared the next time the device starts.

Users should exercise caution when setting the TTL Value greater than 1, to prevent unwanted multicast traffic from propagating through the network. Chapter 3 includes a discussion on user considerations when using multicast.

5-3.2.2.8 Mcast Config

The **Mcast Config** attribute contains the configuration of the device's IP multicast addresses to be used for EtherNet/IP multicast packets. There are three elements to the Mcast Config structure: Alloc Control, Num Mcast, and Mcast Start Addr.

Alloc Control determines how the device shall allocate IP multicast addresses (e.g., whether by algorithm, whether they are explicitly set, etc.) Table 5-3.8 shows the details for Alloc Control.

Table 5-3.8 Alloc Control

Value	Definition
0	Multicast addresses shall be generated using the default allocation algorithm specified in Chapter 3. When this value is specified on a set-attribute or set-attributes-all, the values of Num Mcast and Mcast Start Addr in the set-attribute request shall be 0.
1	Multicast addresses shall be allocated according to the values specified in Num Mcast and Mcast Start Addr.
2	Reserved

Num Mcast is the number of IP multicast addresses allocated. The maximum number of multicast addresses is device specific, but shall not exceed the number of EtherNet/IP multicast connections supported by the device.

Mcast Start Addr is the starting multicast address from which Num Mcast addresses are allocated.

When set, the Mcast Config attribute shall be saved in non-volatile memory, and shall take affect the next time the device starts. Setting the Mcast Config attribute shall also cause the Mcast Pending bit in the Interface Status attribute to be set, indicating that there is pending configuration. When a new Mcast Config value is pending, Get_Attribute_Single or Get_Attributes_All requests shall return the pending value. The Mcast Pending bit shall be cleared the next time the device starts.

When the multicast addresses are generated using the default algorithm, Num Mcast and Mcast Start Addr shall report the values generated by the algorithm.

5-3.3 Common Services

5-3.3.1 All Services

The TCP/IP Interface Object shall provide the following common services.

Table 5-3.9 Common Services

Service	Need in Implementation			
Code	Class	Instance	Service name	Description of Service
0x01	Optional	Optional	Get_Attribute_All	Returns a predefined listing of this objects attributes (See the Get_Attribute_All response definition in section 5-3.3.2)
0x02	n/a	Optional	Set_Attribute_All	Modifies all settable attributes.
0x0E	Conditional	Required	Get_Attribute_Single	Returns the contents of the specified attribute.
0x10	n/a	Required	Set_Attribute_Single	Modifies a single attribute.

5-3.3.2 Get_Attribute_All Response

For class attributes, (since there is only one class attribute) class attribute #1 shall be returned.

For instance attributes, attributes shall be returned in numerical order up to the last implemented attribute. The Get_Attribute_All reply shall be as follows:

Table 5-3.10 Get_Attribute_All

Attribute ID	Size in Bytes	Contents
1	4	Status
2	4	Configuration Capability
3	4	Configuration Control
4	2	Physical Link Object, Path Size
	Variable, 12 bytes max	Physical Link Object, Path (if Path Size is non-zero)
5	4	IP Address
	4	Network Mask
	4	Gateway Address
	4	Name Server
	4	Secondary Name Server
	2	Domain Name Length
	Variable, equal to Domain Name Length	Domain Name
	1	Pad byte only if Domain Name Length is odd
6	2	Host Name Length
	Variable, equal to Host Name Length	Host Name
	1	Pad byte only if Host Name Length is odd
7	6 octets	See CIP Safety Specification Volume 5, Chapter 3. Not present if attribute 7 is not implemented. Shall be 0 when additional attributes greater than attribute ID 7 are included.

Attribute ID	Size in Bytes	Contents
8	1 octet	TTL Value. Not present if attribute 8 is not implemented. Shall be 0 when additional attributes greater than attribute ID 8 are included.
9	8 octets	Mcast Config. Not present if attribute 9 is not implemented. Shall be 0 when additional attributes greater than attribute ID 9 are included.

The lengths of the Physical Link Object path, Domain Name, and Host Name are not known before issuing the Get_Attribute_All service request. Implementers shall be prepared to accept a response containing the maximum sizes of the Physical Link Object path (6 UINTs), the Domain Name (48 USINTs), and Host Name (64 USINTs).

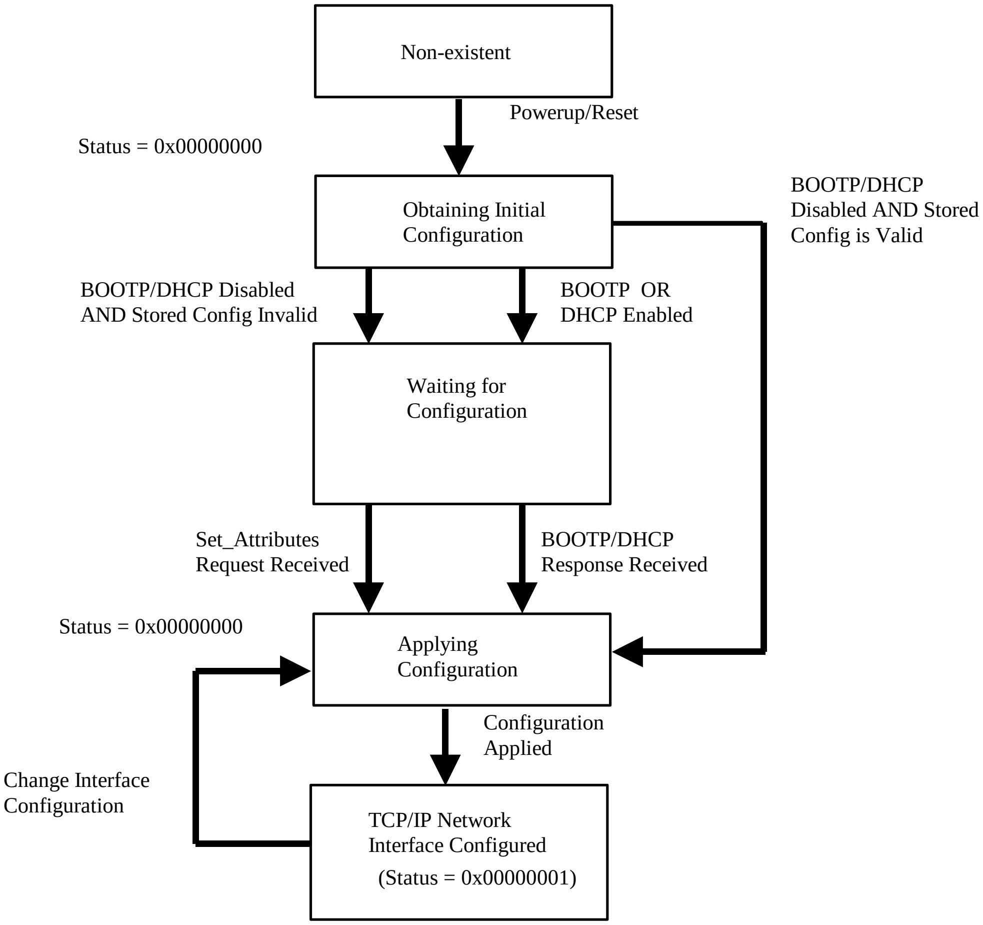
5-3.3.3 Set_Attribute_All Request

The instance Set_Attribute_All request contains the Configuration Control attribute, followed by the Interface Configuration attribute.

5-3.4 Behavior

The behavior of the TCP/IP Interface Object shall be as illustrated in the State Transition Diagram below. Note that the act of obtaining an initial executable image via BOOTP/TFTP shall not be considered within the scope of the TCP/IP Interface Object behavior. Devices are free to implement such behavior, however it shall be considered to have occurred in the “Non-existent” state.

Figure 5-3.1 State Diagram Showing the Behavior of the TCP/IP Object


Non-existent
Powerup/Reset
Status = 0x00000000
Obtaining Initial Configuration
BOOTP/DHCP Disabled AND Stored Config is Valid
BOOTP/DHCP Disabled AND Stored Config Invalid
BOOTP OR DHCP Enabled
Waiting for Configuration
Set_Attributes Request Received
BOOTP/DHCP Response Received
Status = 0x00000000
Applying Configuration
Configuration Applied
Change Interface Configuration
TCP/IP Network Interface Configured
(Status = 0x00000001)

5-4 Ethernet Link Object

Class Code: F6 $_{\text{hex}}$

5-4.1 Scope

The Ethernet Link Object maintains link-specific counters and status information for a Ethernet 802.3 communications interface. Each device shall support exactly one instance of the Ethernet Link Object for each Ethernet IEEE 802.3 communications interface on the module. Devices may use an Ethernet Link Object instance for an internally accessible interface, such as an internal port for an embedded switch, Refer to Chapter 6 (Device Profiles) for additional information on multi-port EtherNet/IP devices.

5-4.2 Revision History

Since the initial release of this object class definition changes have been made that require a revision update of this object class. The table below represents the revision history.

Table 5-4.1 Revision History

Revision	Reason for Object Definition Update
1	Initial revision of this object definition
2	Add Instance Attribute 6, Interface Control
3	Add new instance attributes 7-10 providing support for multiple port Ethernet devices

5-4.3 Attributes

5-4.3.1 Class Attributes

The Ethernet Link Object shall support the following class attributes.

Table 5-4.2 Class Attributes

Attribute ID	Need in Implementation	Access Rule	Name	Data Type	Description of Attribute	Semantics of Values
1	Required	Get	Revision	UINT	Revision of this object	The current value assigned to this attribute is three
2	Conditional [1]	Get	Max Instance	UINT	Maximum instance number of an object currently created in this class level of the device	The largest instance number of a created object at this class hierarchy level
3	Conditional [1]	Get	Number of Instances	UINT	Number of object instances currently created at this class level of the device	The number of object instances at this class hierarchy level
4 thru 7	These class attributes are optional and are described in Volume 1, Chapter 4.					
1 Required if the number of instances is greater than 1.						

An error reading the Class Revision attribute implies this is a revision 1 only implementation.

5-4.3.2 Instance Attributes

The Ethernet Link Object shall support the following instance attributes.

Table 5-4.3 Instance Attributes

Attr ID	Need in Implementation	Access Rule	Name	Data Type	Description of Attribute	Semantics of Values
1	Required	Get	Interface Speed	UDINT	Interface speed currently in use	Speed in Mbps (e.g., 0, 10, 100, 1000, etc.)
2	Required	Get	Interface Flags	DWORD	Interface status flags	Bit map of interface flags. See section 5-4.3.2.1
3	Required	Get	Physical Address	ARRAY of 6 USINTs	MAC layer address	See section 5-4.3.2.3
4	Conditional[1]	Get	Interface Counters	STRUCT of:		See section 5-4.3.2.4
			In Octets	UDINT	Octets received on the interface	
			In Ucast Packets	UDINT	Unicast packets received on the interface	
			In NUcast Packets	UDINT	Non-unicast packets received on the interface	
			In Discards	UDINT	Inbound packets received on the interface but discarded	
			In Errors	UDINT	Inbound packets that contain errors (does not include In Discards)	
			In Unknown Protos	UDINT	Inbound packets with unknown protocol	
			Out Octets	UDINT	Octets sent on the interface	
			Out Ucast Packets	UDINT	Unicast packets sent on the interface	
			Out NUcast Packets	UDINT	Non-unicast packets sent on the interface	
			Out Discards	UDINT	Outbound packets discarded	
			Out Errors	UDINT	Outbound packets that contain errors	

Attr ID	Need in Implementation	Access Rule	Name	Data Type	Description of Attribute	Semantics of Values
5	Optional	Get	Media Counters	STRUCT of:	Media-specific counters	See section 5-4.3.2.5
			Alignment Errors	UDINT	Frames received that are not an integral number of octets in length	
			FCS Errors	UDINT	Frames received that do not pass the FCS check	
			Single Collisions	UDINT	Successfully transmitted frames which experienced exactly one collision	
			Multiple Collisions	UDINT	Successfully transmitted frames which experienced more than one collision	
			SQE Test Errors	UDINT	Number of times SQE test error message is generated	
			Deferred Transmissions	UDINT	Frames for which first transmission attempt is delayed because the medium is busy	
			Late Collisions	UDINT	Number of times a collision is detected later than 512 bit-times into the transmission of a packet	
			Excessive Collisions	UDINT	Frames for which transmission fails due to excessive collisions	
			MAC Transmit Errors	UDINT	Frames for which transmission fails due to an internal MAC sublayer transmit error	
			Carrier Sense Errors	UDINT	Times that the carrier sense condition was lost or never asserted when attempting to transmit a frame	
			Frame Too Long	UDINT	Frames received that exceed the maximum permitted frame size	
			MAC Receive Errors	UDINT	Frames for which reception on an interface fails due to an internal MAC sublayer receive error	
6	Optional	Set	Interface Control	STRUCT of:	Configuration for physical interface	See section 5-4.3.2.6
			Control Bits	WORD	Interface Control Bits	
			Forced Interface Speed	UINT	Speed at which the interface shall be forced to operate	Speed in Mbps (10, 100, 1000, etc.)
7	Optional	Get	Interface Type	USINT	Type of interface: twisted pair, fiber, internal, etc	See section 5-4.3.2.7
8	Optional	Get	Interface State	USINT	Current state of the interface: operational, disabled, etc	See section 5-4.3.2.8

Attr ID	Need in Implementation	Access Rule	Name	Data Type	Description of Attribute	Semantics of Values
9	Optional	Set	Admin State	USINT	Administrative state: enable, disable	See section 5-4.3.2.9
10	Conditional [2]	Get	Interface Label	SHORT_STRING	Human readable identification	See section 5-4.3.2.10

1 The Interface Counters attribute is required if the Media Counters attribute is implemented.

2 Required if number of instances is greater than 1.

5-4.3.2.1 Interface Flags

The Interface Flags attribute contains status and configuration information about the physical interface and shall be as follows:

Table 5-4.4 Interface Flags

Bit(s):	Called:	Definition
0	Link Status	Indicates whether or not the Ethernet 802.3 communications interface is connected to an active network. 0 indicates an inactive link; 1 indicates an active link. The determination of link status is implementation specific. In some cases devices can tell whether the link is active via hardware/driver support. In other cases, the device may only be able to tell whether the link is active by the presence of incoming packets.
1	Half/Full Duplex	Indicates the duplex mode currently in use. 0 indicates the interface is running half duplex; 1 indicates full duplex. Note that if the Link Status flag is 0, then the value of the Half/Full Duplex flag is indeterminate.
2-4	Negotiation Status	Indicates the status of link auto-negotiation 0 = Auto-negotiation in progress. 1 = Auto-negotiation and speed detection failed. Using default values for speed and duplex. Default values are product-dependent; recommended defaults are 10Mbps and half duplex. 2 = Auto negotiation failed but detected speed. Duplex was defaulted. Default value is product-dependent; recommended default is half duplex. 3 = Successfully negotiated speed and duplex. 4 = Auto-negotiation not attempted. Forced speed and duplex.
5	Manual Setting Requires Reset	0 indicates the interface can activate changes to link parameters (auto-negotiate, duplex mode, interface speed) automatically. 1 indicates the device requires a Reset service be issued to its Identity Object in order for the changes to take effect.
6	Local Hardware Fault	0 indicates the interface detects no local hardware fault; 1 indicates a local hardware fault is detected. The meaning of this is product-specific. Examples are an AUI/MII interface detects no transceiver attached or a radio modem detects no antennae attached. In contrast to the soft, possible self-correcting nature of the Link Status being inactive, this is assumed a hard-fault requiring user intervention.
7-31	Reserved	Shall be set to zero

5-4.3.2.2 Interface Speed

The Interface Speed attribute shall indicate the speed at which the interface is currently running (e.g., 10 Mbps, 100 Mbps, 1 Gbps, etc.) A value of 0 shall be used to indicate that the speed of the interface is indeterminate. The scale of the attribute is in Mbps, so if the interface is running at 100 Mbps then the value of Interface Speed attribute shall be 100. The Interface Speed is intended to represent the media bandwidth; the attribute shall not be doubled if the interface is running in full-duplex mode.

5-4.3.2.3 Physical Address

The Physical Address attribute contains the interface's MAC layer address. The Physical Address is an array of octets. The recommended display format is "XX-XX-XX-XX-XX-XX", starting with the first octet. Note that the Physical Address is not a settable attribute. The Ethernet address shall be assigned by the manufacturer, and shall be unique per IEEE 802.3 requirements. Devices with multiple ports but a single MAC interface (e.g., a device with a embedded switch technology) may use the same value for this attribute in each instance of the Ethernet Link Object. The general requirement is that the value of this attribute shall be the MAC address used for packets to and from the device's own MAC interface over this physical port.

5-4.3.2.4 Interface Counters

The Interface Counters attribute contains counters relevant to the receipt of packets on the interface. These counters shall be as defined in RFC 1213 "MIB-II Management Information Base". The Interface Counters are a conditional attribute; they shall be implemented if the Media Counters attribute is implemented. Multi-port devices with a single MAC interface (e.g., device with an embedded switch) shall maintain counter values in one of three ways:

1. In each instance, count the MAC frames sent/received by the device itself over the port represented by that instance (i.e., each physical interface counts the MAC frames sent/received over that interface). This is the preferred solution.

2. Use counter values of 0 for those instances that correspond to the external switch ports; count MAC frames in the instance that corresponds to the internal device interface

3. Use the same counter values for all instances, counting MAC frames sent/received by the device itself

5-4.3.2.5 Media Counters

The Media Counters attribute contains counters specific to Ethernet media. These counters shall be as defined by RFC 1643, "Definitions of Managed Objects for Ethernet-Like Interface Types". If this attribute is implemented the Interface Counters shall also be implemented. Instances that refer to internal interfaces may set the values of the Interface Counters to 0.

Note: some underlying hardware or system implementations may not provide all of the Media Counters. In the case of fiber media, some of the counters do not apply (e.g., collision counters). Devices shall use values of 0 for counters that are not implemented.

5-4.3.2.6 Interface Control

The Interface Control attribute is a structure consisting of Control Bits and Forced Interface Speed and shall be as follows:

5-4.3.2.6.1 Control Bits

Table 5-4.5 Control Bits

Bit(s):	Called:	Definition
0	Auto-negotiate	0 indicates 802.3 link auto-negotiation is disabled. 1 indicates auto-negotiation is enabled. If auto-negotiation is disabled, then the device shall use the settings indicated by the Forced Duplex Mode and Forced Interface Speed bits.
1	Forced Duplex Mode	If the Auto-negotiate bit is 0, the Forced Duplex Mode bit indicates whether the interface shall operate in full or half duplex mode. 0 indicates the interface duplex should be half duplex. 1 indicates the interface duplex should be full duplex. Interfaces not supporting the requested duplex shall return a GRC hex 0x09 (Invalid Attribute Value). If auto-negotiation is enabled, attempting to set the Forced Duplex Mode bits shall result in a GRC hex 0x0C (Object State Conflict).
2-15	Reserved	Shall be set to zero

5-4.3.2.6.2 Forced Interface Speed

If the Auto-negotiate bit is 0, the Forced Interface Speed bits indicate the speed at which the interface shall operate. Speed is specified in megabits per second (e.g., for 10 Mbps Ethernet, the Interface Speed shall be 10). Interfaces not supporting the requested speed should return a GRC hex 0x09 (Invalid Attribute Value).

If auto-negotiation is enabled, attempting to set the Forced Interface Speed shall result in a GRC hex 0x0C (Object State Conflict).

5-4.3.2.7 Interface Type

The Interface Type attribute shall indicate the type of the physical interface. Table 5-4.6 shows the Interface Type values. This attribute shall be stored in non-volatile memory.

Table 5-4.6 Interface Type

Value	Type of interface
0	Unknown interface type.
1	The interface is internal to the device, for example, in the case of an embedded switch.
2	Twisted-pair (e.g., 10Base-T, 100Base-TX, 1000Base-T, etc.)
3	Optical fiber (e.g., 100Base-FX)
4-256	Reserved.

5-4.3.2.8 Interface State

The Interface State attribute shall indicate the current operational state of the interface. Table 5-4.7 shows the Interface State values. This attribute shall be stored in volatile memory.

Table 5-4.7 Interface State

Value	Interface State
0	Unknown interface state
1	The interface is enabled and is ready to send and receive data
2	The interface is disabled
3	The interface is testing
4-256	Reserved.

5-4.3.2.9 Admin State

The Admin State attribute shall allow administrative setting of the interface state. Table 5-4.8 shows the Admin State values. This attribute shall be stored in non-volatile memory.

Table 5-4.8 Admin State

Value	Admin State
0	Reserved
1	Enable the interface
2	Disable the interface. If this is the only CIP communications interface, a request to disable the interface shall result in an error (status code 0x09).
3-256	Reserved.

5-4.3.2.10 Interface Label

The Interface Label attribute shall be a text string that describes the interface. The content of the string is vendor specific. For internal interfaces the text string should include “internal” somewhere in the string. The maximum number of characters in this string is 64. This attribute shall be stored in non-volatile memory.

5-4.4 Common Services

5-4.4.1 All Services

The Ethernet Link Object shall provide the following common services.

Table 5-4.9 Common Services

Service	Need in Implementation			
Code	Class	Instance	Service Name	Description of Service
0x01	Optional	Optional	Get_Attribute_All	Returns a predefined listing of this objects attributes (See the Get_Attribute_All response definition in section 5-4.4.2)
0x0E	Conditional	Required	Get_Attribute_Single	Returns the contents of the specified attribute.
0x10	n/a	Conditional	Set_Attribute_Single	Modifies a single attribute.

The Get_Attribute_Single shall be implemented for the class attribute if the class attribute is implemented.

The Set_Attribute_Single service shall be implemented if the Interface Control attribute is implemented.

5-4.4.2 Get_Attribute_All Response

For class attributes, since there is only one possible attribute, the Get_Attribute_All response is the same as the Get_Attribute_Single response. If no class attributes are implemented, then no data is returned in the data portion of the reply.

For instance attributes, attributes shall be returned in numerical order, up to the last implemented attribute. All 0's shall be returned for the Interface Counters and Media Counters attributes if they are not implemented but the Interface Control attribute is implemented.

5-4.5 Class-Specific Services

The Ethernet Link Object shall support the following class-specific services:

Table 5-4.10 Class Specific Services

Service	Need in Implementation			
Code	Class	Instance	Service Name	Description of Service
0x4C	n/a	Conditional[1]	Get_and_Clear	Gets then clears the specified attribute (Interface Counters or Media Counters).

[1] The Get_and_Clear service shall only be implemented if the Interface Counters and Media Counters are implemented.

5-4.5.1 Get_and_Clear Service

The Get_and_Clear service is a class-specific service. It is only supported for the Interface Counters and Media Counters attributes. The Get_and_Clear response shall be the same as the Get_Attribute_Single response for the specified attribute. After the response is built, the value of the attribute shall be set to zero.

Volume 2: EtherNet/IP Adaptation of CIP

Chapter 6: Device Profiles

Contents

6-1 Introduction

This chapter of the EtherNet/IP specification contains device profiles that are EtherNet/IP specific.

6-2 Required Objects

At minimum, every EtherNet/IP device shall implement instance number one of each of the following objects:

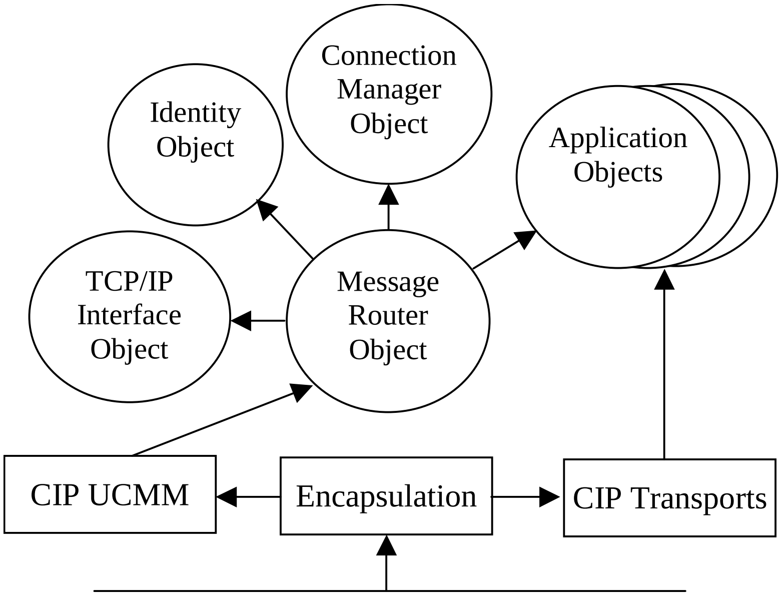
- Identity Object (class code = 0x01)
- Message Router Object (class code = 0x02)
- Connection Manager Object (class code = 0x06)
- TCP/IP Interface Object (class code = 0xF5) and a corresponding link object

If an Ethernet medium is used, the corresponding link object shall be the Ethernet Link Object (class code = 0xF6). If any other medium is used, the vendor shall define a vendor specific link object.

NOTE: This specification permits the use of any medium that supports TCP/IP; however, only the Ethernet medium has been completely standardized here. It is likely in the future that ODVA/CI will standardize other link objects for frequently used TCP/IP media. For example, in the future, a standardized PPP object may be defined.

Although it does not have an object class code, each device shall also implement the CIP Unconnected Message Manager (UCMM).

Figure 6-2.1 Base Device Object Model



6-3 Devices with Multiple Interfaces

EtherNet/IP devices may implement multiple network interfaces, for example:

- A device with a single IP address with two Ethernet ports implemented as an embedded switch
- An EtherNet/IP-enabled switch with multiple Ethernet ports and with EtherNet/IP communications for the switch itself
- A device with a single Ethernet interface and multiple IP addresses

Note: The specification does not address any behavior related to the Ethernet switching function in the device examples mentioned above. The intent of the specification at present is only to specify to allowable configurations of TCP/IP Interface Object and Ethernet Link Object instances in order to support the device possibilities above.

Devices with multiple interfaces shall implement multiple instances of the TCP/IP interface Object and physical link objects (e.g., Ethernet Link Object) as applicable to the device function as listed below:

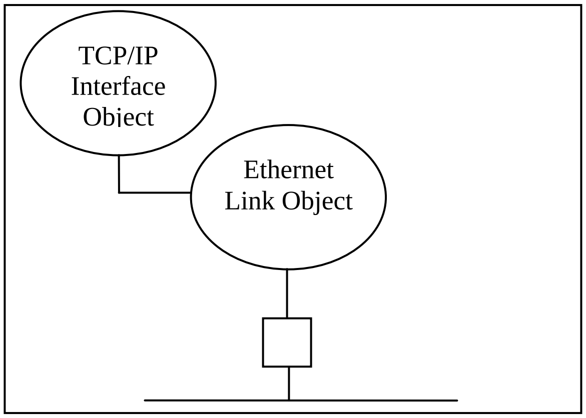
- 1 instance of the TCP/IP Interface Object for each TCP/IP interface (i.e., for each IP address).
- 1 instance of a physical link object (e.g., Ethernet Link Object) for each physical interface exposed via EtherNet/IP. Devices may elect not to expose all interfaces,
- Devices may use a physical link object instance (e.g., Ethernet Link Object) for an internal interface such as the internal device port of an embedded switch.
- Devices with multiple IP addresses may elect to represent each IP interface as a different CIP port, and allow CIP messages to be routed from one port to the other. For example, the CIP connection path would enter one of the ports and exit the other port. In this scenario, the device shall implement one instance of the Port Object for each CIP port, and shall implement the CIP routing mechanism described in Volume 1. Devices are not required to implement the Port Object unless they implement multiple CIP ports.

The following sections illustrate some of the different possibilities for devices with varying numbers of Ethernet interfaces.

6-3.1 Case 1: Single Port Device, 1 IP Address

This is the normal case for most EtherNet/IP devices : single Ethernet interface, single IP address. In this case, there is one instance of the TCP/IP Interface Object, and one instance of the Ethernet Link Object that represents the physical interface.

Figure 6-3.1 Case 1 Illustration

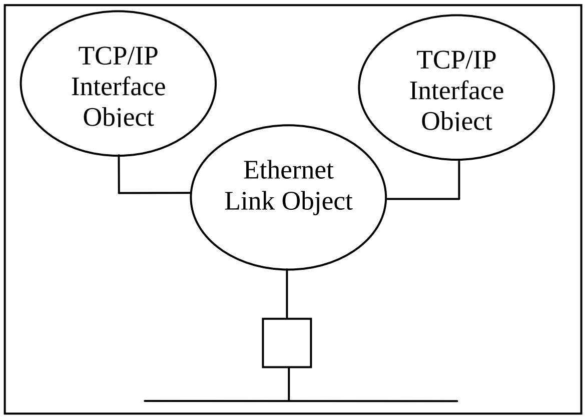


6-3.2 Case 2, Single Port Device, Multiple IP Addresses

Example : A device with a single Ethernet interface wishes to expose a second IP address.

In this example, there are 2 instances of the TCP/IP Interface Object, and one instance of the Ethernet Link Object that represents the physical interface. TCP/IP Interface Object class attributes 2 (Max. instances) and 3 (Number of instances) are used. The instance attribute 4 (Physical Link Object) of both instances refers to the same Ethernet Link Object

Figure 6-3.2 Case 2 Illustration

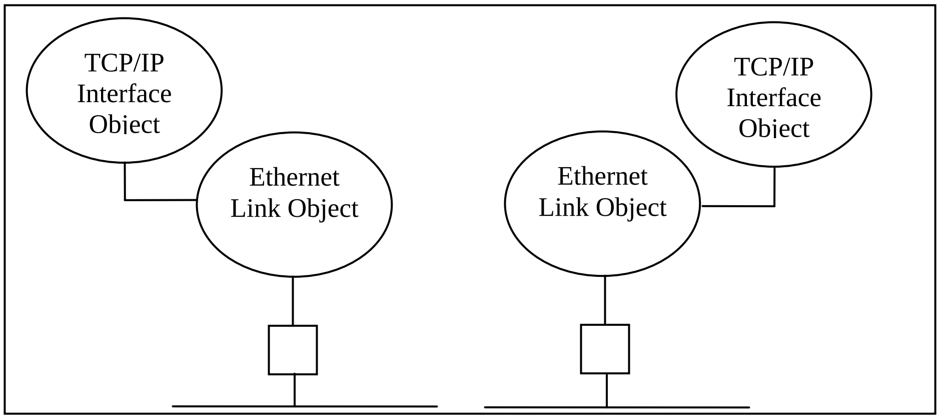


6-3.3 Case 3, Device with 2 Ethernet ports. Each port has an associated IP Address

Example : A device with a multiple Ethernet interfaces, each with an associated IP address. Each interface would be a different CIP-addressable port (i.e., there would be a Port Object instance per interface).

In this example, there are 2 instances of the TCP/IP Interface Object, and 2 instances of the Ethernet Link Object that represent each the physical interface. TCP/IP Interface Object / Ethernet Link Object class attributes 2 (Max. instances) and 3 (Number of instances) are used. The new Ethernet Link instance attribute 10 (Interface Label) is used to get the correlation between the Ethernet Link Object and the physical port.

Figure 6-3.3 Case 3 Illustration



Note: this example can be generalized to more than 2 ports.

6-3.4 Case 4, Device with Multiple Ethernet interfaces and a single IP Address and CIP Interface

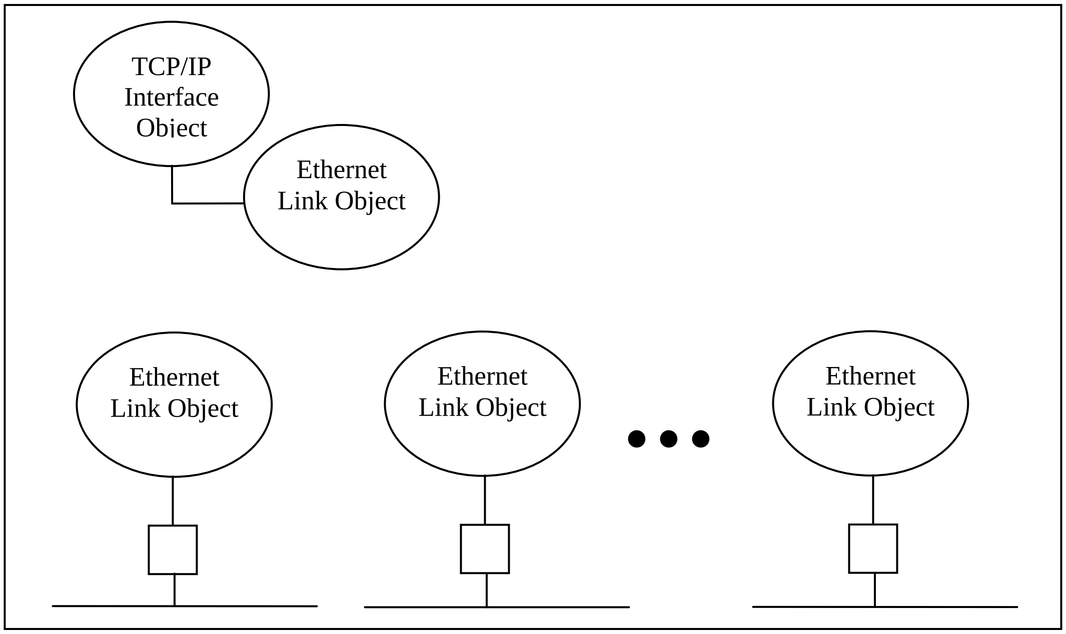
Example:An example is a device with embedded switch technology (to support linear topology), or an EtherNet/IP-enabled switch. In this case, the device has multiple Ethernet interfaces, but the interfaces are not CIP-addressable ports (i.e., they do not have corresponding CIP port numbers or Port Object instances).

It is however useful to allow configuration of the Ethernet interfaces, for example to set port speed and duplex via the Ethernet Link Object. Note that there is no intent to specify switching behavior of the device.

In this case, there is a single instance of the TCP/IP Interface Object, an optional “internal” instance of the Ethernet Link Object (corresponding to the internal device port), and then Ethernet Link Object instances for each of the physical interfaces.

New Ethernet Link Object class attributes 2 (Max. instances) and 3 (Number of instances) are used. The new Ethernet Link instance attribute 10 (Interface Label) is used to get the correlation between the Ethernet Link Object and the physical port. The new Ethernet Link instance attribute 7 (Interface Type) defines the kind of object (internal/external).

Figure 6-3.4 Case 4 Illustration



Refer to Chapter 5 (Object Library) for specific EtherNet/IP object definitions and requirements.

This page is intentionally left blank

Volume 2: EtherNet/IP Adaptation of CIP

Chapter 7: Electronic Data Sheets

Contents

7-1 Introduction

This chapter of the EtherNet/IP specification contains additions to the definition of electronic data sheets (EDS) that are EtherNet/IP specific. See the CIP Common specification for more information about the format of electronic data sheets and the definition of EDS related terms such as EDS section, EDS entry and EDS field.

7-2 [Device Classification] Section

In the [Device Classification] section of the EDS, for any EtherNet/IP compliant device, there shall be at least one ClassN keyword entry with its first field set to EthernetIP. As shown in Figure 7-3.1, no sub-classifications shall be present.

7-3 [Port] Section

In the [Port] section of the EDS (see Figure 7-3.1 for an example), the PortN entry corresponding to the EtherNet/IP compliant port shall be set as follows:

The “Port Type” field shall have a value of “TCP”.

The optional “Port Object” field shall be set to the path of the TCP Object for this port.

No additional requirements, beyond those in the CIP Common Specification (Volume 1), are placed on the “Name” and “Port Number” fields. .

NOTE: An EDS for an EtherNet/IP device does not directly refer to the link object for the EtherNet/IP port (for example, the Ethernet Link Object) since it can be referenced through the TCP Object for the port.

Figure 7-3.1 Example EDS of an EtherNet/IP Device

```
[File]
        DescText = "Widget EDS File";
        CreateData = 02-07-2001;
        CreateTime = 17:51:44;
        ModDate = 04-06-1997;
        ModTime = 22:07:30;
        Revision = 2.1;
        HomeURL = "http://www.controlnet.org/EDS/12345.eds";
[Device]
        VendCode = 65535;
        VendName = "Widget-Works, Inc.";
        ProdType = 0;
        ProdTypeStr = "Generic";
        ProdCode = 10;
        MajRev = 1;
        MinRev = 1;
        ProdName = "Smart-Widget";
        Catalog = "1492-SW";
        Icon = "widget.ico";
[Device Classification]
        Class1 = EthernetIP;

[Port]
        Port1 =
                TCP,
                "EtherNet/IP port",
                "20 F5 24 01",
                1;
```

Volume 2: EtherNet/IP Adaptation of CIP

Chapter 8: Physical Layer

Contents

8-1 Introduction

Chapter 8 specifies EtherNet/IP media and physical layer requirements for EtherNet/IP installations and active devices. In some cases, industrial environmental requirements may exceed those used in office environments. Products and components may need to be enhanced to provide the level of performance required to support industrial applications. Some of these enhancements include noise rejection, sealing, voltage isolation, chemical resistance, shock, vibration and wide/dynamic temperature ranges.

8-2 General

The following sections will delineate physical layer media variants for EtherNet/IP. This standard does not define requirements for coaxial Ethernet components or commercial off the shelf components (COMMERCIAL). Requirements for these components can be found in ANSI IEEE 802.3 standard and TIA 568 standards. In this chapter, components that fall under these standards will be referred to as “standard components”. Systems constructed of standard components have been deployed in industrial environments primarily in information systems and limited control applications. These systems, for the most part, have been successfully providing services at 10 Mbps. Whether providing services at 10 Mbps or 100 Mbps, standard components are recognized and acceptable for use within the guidelines of this specification. However, because testing has shown that in order to survive harsh environments both in high noise, diverse temperatures and the presence of chemicals both in liquid and solid forms, there are system and component enhancements that are required.

This document defines component performance up to 100 Mbps. The component specifications herein are optimized for data rates of 10 Mbps and 100 Mbps. The copper variant shall include both shielded and unshielded twisted pair cable technologies. The signaling and coupling for copper twisted pair methods are described in section 8-9.2.1. Products constructed with Commercial components are not eligible for the industrial conformance check mark since the Commercial products cannot be directly mapped into any controlling standards controlled by ODVA. Products constructed of Commercial components are eligible for the Commercial checkmark.

8-3 Grounding (earthing) and Bonding

Grounding and bonding in the communications coverage area is critical to the performance of EtherNet/IP networks. This topic is a subject of further study.

8-4 Environmental Compatibility

Products and components installed in EtherNet/IP networks shall be compatible with the local environmental conditions ether by design or a combination of design and mitigation.

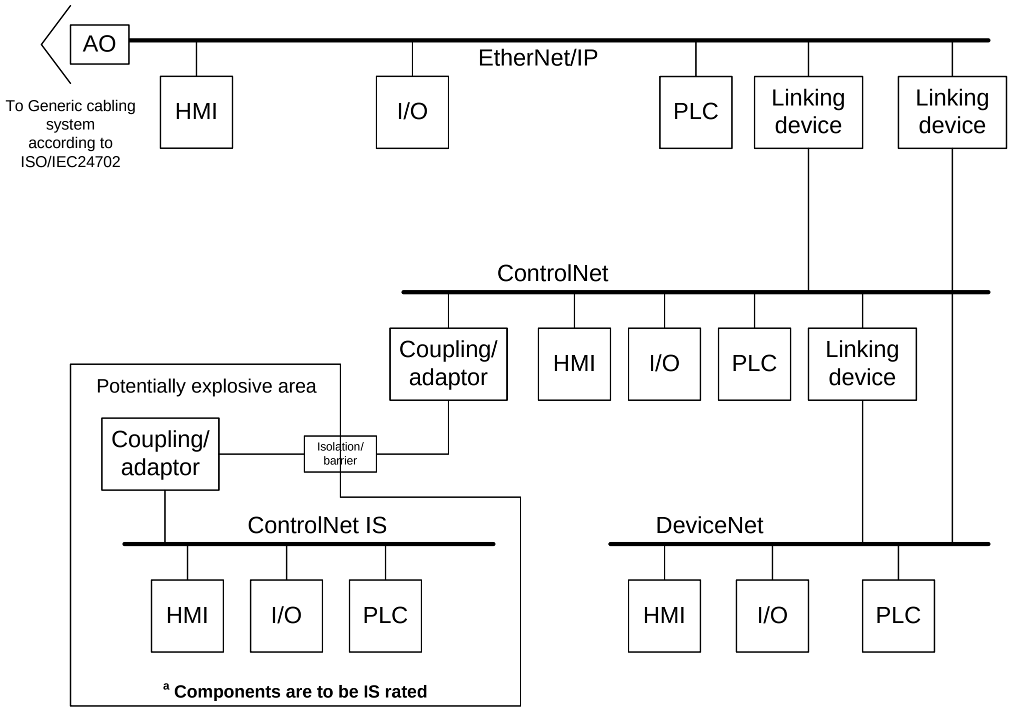
8-5 Auxiliary Power

Auxiliary Power network is being defined as a collective effort by the Physical Layer SIGs and JSIGS within ODVA. This topic is currently under study. Designers of active components planning to use Auxiliary power should review the current revision of this work before implementing connectors and connector pin out schemes in their products.

8-6 Supported Physical Topologies and relation to other networks

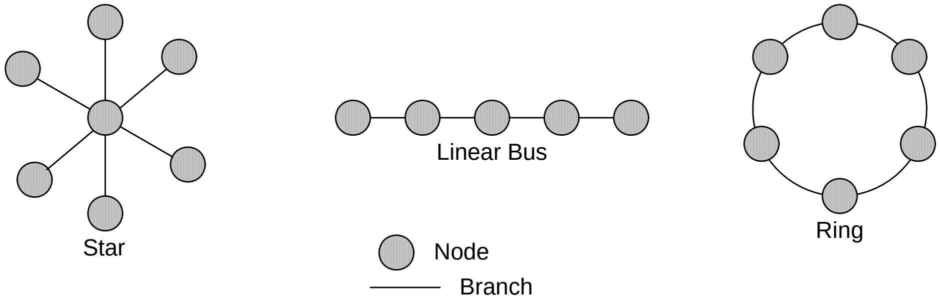
Figure 8-6.1 shows how EtherNet/IP maybe connected to other CIP networks and the generic telecommunications infrastructure. Connection to the generic telecommunications infrastructure should be through an appropriate security device to prevent inadvertent interruption to the control networks. The industrial generic cabling systems may not provide a level of performance required for control of industrial machinery and processes. The cabling is connected to the generic cabling as defined by ISO/IEC 24702 and TIA 1005 through an Automation Outlet (AO). The requirements of the automation outlet are defined in this chapter.

Figure 8-6.1 Relationship to Other Networks



There are three basic topologies for Ethernet based networks. EtherNet/IP supports all three of the standard active physical topologies as detailed in Figure 8-6.2.

Figure 8-6.2 Active Physical Topologies



A switching device is expected to be a dedicated device usually located in the center of the star physical topology. For other topologies such as the physical linear bus and ring the switching device may be embedded in a node. A physical linear bus and ring device requires two physical connections to the cabling infrastructure.

8-7 Performance Levels

Sections 8-7.1and 8-7.2 define two levels of product performance: Commercial EtherNet/IP and Industrial EtherNet/IP enhanced components. COMMERCIAL cabling components may require mitigation in the form of isolation and separation from the various harsh elements found in a typical industrial environment. The EtherNet/IP industrial connectivity component requirements have added enhancements to reduce the level of mitigation that might otherwise be required. This clause also details the requirements for active devices both based on COMMERCIAL and industrially hardened in the same way as the cabling components.

NOTE: There are many sections within this chapter that specify optional requirements. This section distils these requirements into two distinct levels: commercial copper and fiber and industrial EtherNet/IP copper and fiber.

8-7.1 COMMERCIAL based EtherNet/IP products

8-7.1.1 Copper and Fiber Cabling Components

The requirements for COMMERCIAL cabling components are defined ANSI/TIA/EIA-568-B series standards and section 8-8. The use of COMMERCIAL components may degrade system performance. Use of such products or components may result in unsatisfactory performance in industrial control applications.

8-7.1.2 Active Interfaces (PMD)

Copper and fiber based COMMERCIAL active products shall meet the minimum requirements of this chapter. Since these devices are not expected to be industrial hardened, they may require additional mitigation when installed in a harsh environment. The copper interfaces shall provide a non-sealed RJ-45 jack at the network interface. The fiber interface shall provide connectivity to one of the non-sealed connectors detailed in 8-9.5.2.1 (LC, SC or ST).

8-7.2 Industrial EtherNet/IP products

8-7.2.1 EtherNet/IP Copper and Fiber Cabling Components

These components are designed to better withstand high noise and harsh environments common to the industrial environment. Additional consideration is required for the selection of materials in special applications such as robotic and welding applications ect..

8-7.2.2 Industrial EtherNet/IP Active Interfaces

Copper and fiber-based Industrial EtherNet/IP products shall meet all applicable requirements of the EtherNet/IP specification, chapter 9. For a product to achieve the Industrial EtherNet/IP performance in harsh environments level, the physical layer shall conform to the requirements as outlined in section 8-9 of this chapter.

8-8 COMMERCIAL Based EtherNet/IP Products and Physical Layer

The use of COTS components may degrade system performance. Careful consideration should be given to the use of COMMERCIAL components in industrial control applications

8-8.1 Copper Media

8-8.1.1 Cables

The transmission performance of shielded or unshielded 4 pair twisted pair cables shall meet the requirements of ANSI/TIA/EIA-568-B.2 standards and the requirements of E1 columns of Table 8-9.4, Table 8-9.5 and Table 8-9.6.

8-8.1.2 Connectors

8-8.1.2.1 RJ-45 Connector Variant

The RJ-45 connectors are the de-facto standard for Ethernet systems. RJ-45 connectors shall meet the requirements stated in ANSI/TIA/EIA-568-B.2. In addition IEC 60603-7 series defines the mechanical and electrical requirements for the RJ-45 connectors.

8-8.1.3 Length Constraints

The total permanent link length for twisted pair systems is limited to 90m (295 ft). The permanent link shall conform to ANSI/TIA/EIA-568-B.1.

The total channel length for twisted pair systems is 100m (328 ft) including patch cables as defined in ANSI/TIA/EIA-568-B.1. Channel and patch cable design and testing shall be in accordance with ANSI/TIA/EIA-568-B.1 and 'B.2 respectively.

8-9 Industrial EtherNet/IP Media and Physical Layer

8-9.1 Environmental Requirements

Copper and fiber based Industrial EtherNet/IP products should meet the minimum environmental recommendations as defined in Table 8-9.1. Copper and Fiber cabling components shall support the requirements in of Table 8-9.2. Active devices shall meet the minimum EMI requirements of Table 8-9.2. The values found in Table 8-9.2 represent the minimum requirements for IEC light industrial.

Table 8-9.1 Minimum Environmental Recommendations

Environmental Test	Criteria	Industry Standard
Vibration (Unpackaged)		
Frequency Range	10-57Hz	IEC 60068-2-6
Displacement	0.3 mm	
	57-500Hz	
Acceleration	2g	
Shock (Unpackaged)		
Acceleration	15g (operational)	IEC 60068-2-27
	30g (non-operational)	
Temperature		
Operating range	0 °C min. to +60 °C min. *	IEC 60068-2-1 IEC 60068-2-2
Storage	-40 to +70 °C	IEC 60068-2-1 IEC 60068-2-2
Humidity operating		IEC 60068-2-30
	5 to 95% RH condensing	
Ingress protection		
	IP 20 minimum	IEC 60529
Voltage proof (connector only)		IEC 60512-1
Contact/contact	1000 Vd.c. or a.c. peak	
Contact/test panel	1500 Vd.c. or a.c. peak	

* There may be components or topology de-rating for temperatures below 0 degrees C, or above 60 degrees C.

Table 8-9.2 Minimum EMI Requirements for EtherNet/IP Components

Environmental Test	Criteria	Industry Standard
EMI		
ESD	4kv contact 8kv air	IEC 61000-6-2 IEC 61131-2 IEC 61326-1
Radiated RF	10V/m @ 80-1000MHz @ 1kHz 3V/m @ 1.4-2.0GHz @ 1kHz 1V/m @ 2.0-2.7GHz @ 1kHz	IEC 61000-4-3
Conducted RF	10V RMS @ 150kHz-80MHz @ 1kHz	IEC 61000-4-6
EFT Comms to ground	2kv	IEC 61000-4-4
Surge Comms to ground	2kv	IEC 61000-4-5
Magnetic field (50/60Hz)	30A/m, 1 min.	IEC 61000-4-8

8-9.2 Copper Media

8-9.2.1 Copper Media Attachment (Normative References)

A copper media attachment to an EtherNet/IP network shall support shielded and unshielded twisted pair technology. The specifications shall contain enhancements (where needed) based on ANSI/TIA/EIA-568-B.1 category 5e cabling performance levels minimum. The signaling and coupling of these variants shall comply with the requirements of IEEE 802.3, 2005 Ed/TP-PMD standard subject to the deviations listed in this section 8-9.2.4. Likewise, the cable's electrical mechanical and environmental performance shall be as defined in section 8-4. The IEEE 802.3 standard defines many internal interfaces within the physical layer. EtherNet/IP products need not directly implement each of these interfaces, but shall behave as if these interfaces exist. These interfaces may be internal to the node and possibly internal to a semiconductor device.

This standard supports 10BASE T and 100BASE TX copper variants as defined by IEEE Std 802.3, 2005 Ed. and the ANSI X.3.263 TP-PMD. Two pair cabling does not support 100BASE-T4 and is infrequently used, therefore 100BASE-T4 is not supported by this standard.

Active devices shall be fitted with one of the jacks defined by this chapter. Attached cables with flying leads or flying leads with jacks are not allowed.

8-9.2.2 Copper Cabling Commercial and Industrial

The cable is critical in influencing the performance of the network in the presence of high noise. To support industrial information and industrial control systems two basic cable types (Commercial and Industrial EtherNet/IP Cables) are recognized. Only cables adhering to this specification will be eligible for the appropriate conformance check mark.

Cables shall conform to the specifications table below.

Table 8-9.3 Minimum Cable Requirements for Commercial and Industrial cabling

Industrial EtherNet/IP Cable Specifications and Requirements		
Specification	**Type**	
Electrical	**Shielded**	**Unshielded**
Conductors	2 or 4 pairs + Shield	2 or 4 pairs
Attenuation Solid Conductors	ANSI/TIA-EIA 568-B.2 Cat 5e Horizontal	ANSI/TIA-EIA 568-B.2 Cat 5e Horizontal
Attenuation Stranded Conductors	ANSI/TIA-EIA 568-B.2 Cat 5e Patch [1]	ANSI/TIA-EIA 568-B.2 Cat 5e Patch [1]
Impedance (fitted) ASTM 4566	95-110 Ω 1-4 MHz 95 – 107 Ω 4-100 MHz	95-110 Ω 1-4 MHz 95 – 107 Ω 4-100 MHz
RL (dB)	1-10 MHz 20 + 6Log$_{10}$(f) 10-20 MHz 26 20-100 MHz 26-5*Log$_{10}$(f/20)	1-10 MHz 20 + 6Log$_{10}$(f) 10-20 MHz 26 20-100 MHz 26-5*Log$_{10}$(f/20)
NEXT Loss (dB)	ANSI/TIA-EIA 568-B.2 Cat 5e	ANSI/TIA-EIA 568-B.2 Cat 5e
Coupling Attenuation (dB)	Freq (MHz) E1 E2 E3 See Table 8-9.5	NA
Shielding Effectiveness	tbd	N/A
Capacitance unbalance	<= 150pf/100meter	< = 150pf /100meter
DCR	9.38 Ω/100 meters	9.38 Ω/100 meters
DCR Unbalance	3%	3%
TCL	NA	Frequency (MHz) E1 E2 E3 See Table 8-9.4
ELTCTL		Frequency (MHz) E1 E2 E3 See Table 8-9.5
Mechanical	**Shielded**	**Unshielded**
Pulling Tension	111 N	111 N
Breaking Strength	400 N	400 N
Bend Radius	1" at -20C	1" at –20C
Dimensional (Recommended for RJ 45 compatibility)	**Shielded**	**Unshielded**
Jacket OD	0.315" Max	0.315" Max
Insulated Conductor	0.048" Max	0.048" Max

[1] The insertion loss is based on COMMERCIAL cables. Other constructions, such as high flex, may have different performance. Consult the manufacturer for more information.

8-9.2.2.1 Cabling Balance

8-9.2.2.1.1 Unshielded twisted pair transverse conversions loss (TCL) and equal level transverse conversion transfer loss (ELTCTL)

Each pair of unshielded twisted-pair channels shall meet the TCL requirements of Table 8-9.4 and ELTCTL requirements of Table 8-9.5 below. TCL and ELTCTL shall be measured in accordance with ANSI/TIA/EIA-568-B.2-9.

Table 8-9.4 TCL Limits for Unshielded Twisted Pair Cabling

Category	Frequency (MHz)	Minimum TCL (dB) ISO/IEC 24702		
		E_1	E_2	E_3
5e	$1 \le f < 30$ $30 \le f \le 100$	53-15log(f), (40 max) 60.4 -20log(f)	63-15log(f), (40 max) 70.4 -20log(f)	73-15log(f), (40 max) 80.4 -20log(f)

Table 8-9.5 ELTCTL Limits for Unshielded Twisted Pair Cabling

Category	Frequency (MHz)	Minimum ELTCTL (dB) ISO/IEC 24702		
		E_1	E_2	E_3
5e and 6	$1 \le f \le 30$	30-20log(f)	40-20log(f)	50-20log(f), (40 max)

8-9.2.2.1.2 Shielded Twisted Pair Coupling Attenuation

Each pair of screened twisted-pair channels shall meet the coupling attenuation requirements of Table 8-9.6.

Table 8-9.6 Coupling Attenuation for Screened Twisted Pair Cabling

Category	Frequency (MHz)	Minimum Coupling Attenuation (dB) ISO/IEC 24702		
		E_1	E_2	E_3
5e	$30 \le f \le 100$	40	50	60
6	$30 \le f \le 250$	80-20log(f) (Max 40 dB)	90-20log(f) (Max 50 dB)	100-20Log(f) (Max 60 dB)

Note for EMC purposes, coupling attenuation should be measured up to 1 GHz

Coupling attenuation shall be measured in accordance with IEC 61156-5

8-9.2.2.1.3 Two and four pair color codes

Two and four pair cable color codes shall be as defined Table 8-9.7 and Table 8-9.8 in respectively

Table 8-9.7 Two Pair Color Codes

Pair Assignment	Signal Name	2 Pair
Pair 1	TX+	White-orange
	TX-	Orange
Pair 2	RX+	White-green
	RX-	Green

Table 8-9.8 Four Pair Color Codes

TIA Pair Assignment	Signal Name	Color
Pair 2	TX+	White-orange
	TX-	Orange
Pair 3	RX+	White-green
Pair 1[1]	NA	Blue
	NA	White-blue
Pair 3	RX-	Green
Pair 4 [1]	NA	White-brown
	NA	Brown

[1] Not used for 10 Mbps and 100 Mbps TX networks

8-9.2.3 Connectors

8-9.2.3.1 Industrial EtherNet/IP Connector RJ-45 Variant

Attachment to the medium shall be via either of two types of Industrial grade RJ-45 connectors:

- Non-Sealed industrial RJ-45 EtherNet/IP connector – The RJ-45 EtherNet/IP connector shall meet the IEC 60603-7 standard and additional requirements of this chapter.
- Sealed Industrial EtherNet/IP RJ-45 connector housing – The IP67 sealed industrial EtherNet/IP connector housing shall conform to the specifications IEC 61076-3-106.

8-9.2.3.1.1 Sealed and Non-Sealed Industrial EtherNet/IP Connector

Standard industrial hardened RJ-45 connector shall meet the following specifications:

Industrial EtherNet/IP Connector Specifications and Requirements		
Specification	**Type**	
Electrical	**RJ-45-Shielded**	**RJ-45**
Conductors	8 + 1 Shield	8
Insertion Loss	ANSI/TIA/EIA-568-B.2 Category 5E	ANSI/TIA/EIA-568-B.2 Category 5E
RL	ANSI/TIA/EIA-568-B.2 Category 5E	ANSI/TIA/EIA-568-B.2 Category 5E
NEXT Loss	ANSI/TIA/EIA-568-B.2 Category 5E	ANSI/TIA/EIA-568-B.2 Category 5E
Shielding Effectiveness	ANSI/TIA/EIA-568-B.2 Category 5E	N/A

Mechanical	RJ-45-Shielded	RJ-45
Gender	Plug and Socket	Plug and Socket
Mating Specification	CEI IEC 60603-7	CEI IEC 60603-7
Contact plating	50u inches min. gold over 100u inches min. nickel or equivalent plating system	50u inches min. gold over 100u inches min. nickel or equivalent plating system
Contact LLCR over life	< 20 mΩ	< 20 mΩ
Initial Contact Low Level Contact Resistance	<=2.5 mΩ	<=2.5 mΩ
Minimum contact force	100 grams	100 grams
Minimum plug retention force [1]	133 N	133 N
Contact Life	750 insertions and extractions min.	750 insertions and extractions min.

[1] Required when the connector is used as a standalone connector (not in a protective shell)

The non-sealed connector shall be wired in accordance with the pin/wire assignments in Table 8-9.9.

Table 8-9.9 8-Way Modular Connector Pin/Pair Cable Assignment

PIN	Signal Name	Pin T568A	Pair Assignment	Pin T568B	Pair Assignment
1	TXD+	White Green	Pair 3	White Orange	Pair 2
2	TXD-	Green		Orange	
3	RXD+	White Orange	Pair 2	White Green	Pair 3
4	NA[1]	Blue	Pair 1	Blue	Pair 1
5	NA[1]	White Blue		White Blue	
6	RXD-	Orange	Pair 2	Green	Pair 3
7	NA[1]	White Brown	Pair 4	White Brown	Pair 4
8	NA[1]	Brown		Brown	

[1] Not used for 10 Mbps and 100 Mbps Networks

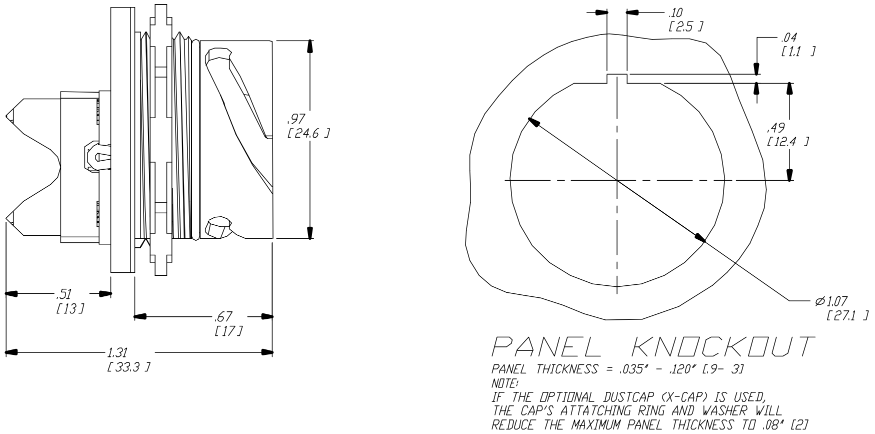
Both ends of the cable shall be wired the same unless constructing a crossover cable.

8-9.2.3.1.2 Sealed Industrial EtherNet/IP RJ-45 Housing

The sealing interface shall meet a minimum of IP67 sealing performance as defined in IEC 60529. The pin/pair wiring of section 8-9.2.3.1.1 applies to the Sealed Industrial EtherNet/IP 8-Way modular connector. Cross over cable are allowed within the same connector family.

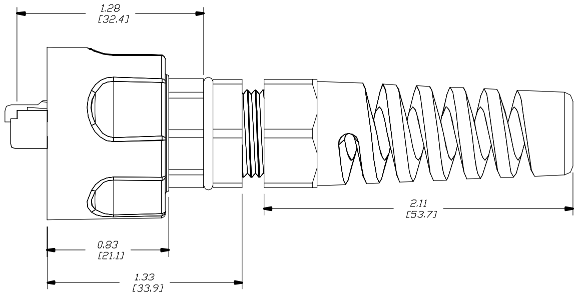
The Sealed RJ-45 variant 1 is based on the IEC 61076-3-106 specification. The following sealed jack drawing sufficiently defines the jack to maintain compatibility for mating and sealing amongst various vendors who may make one or both parts. The jack may be offered as a PCB mount, bulkhead connector and cable end either field installed or manufactured assembly. The jack is fully compatible with standard off-the-shelf plugs.

Figure 8-9.1 Typical Sealed Jack


.10
[2.5]
.04
[1.1]
.97
[24.6]
.49
[12.4]
.51
[13]
.67
[17]
1.31
[33.3]
⌀1.07
[27.1]
PANEL KNOCKOUT
PANEL THICKNESS = .035" - .120" [.9- 3]
NOTE:
IF THE OPTIONAL DUSTCAP (X-CAP) IS USED,
THE CAP'S ATTATCHING RING AND WASHER WILL
REDUCE THE MAXIMUM PANEL THICKNESS TO .08" [2]

The Sealed RJ 45 variant 1 is based on the IEC 61076-3-106 specification. The following sealed plug drawing sufficiently defines the plug to maintain compatibility for mating and sealing amongst various vendors who may make one or both parts. The plug may be offered as a field installable or manufactured cable assembly. The plug housing will accommodate a standard plug as defined by IEC 60603-7 standard with the exception of the locking mechanism, which is disabled.

Figure 8-9.2 Typical Sealed Plug



8-9.2.3.1.3 Sealed M12-4 "D" Coding

The M12-4 "Type D" coding connector is well known and accepted in industrial ethernet applications – for more than 20 years it has been the standard for connection of sensors in the industry. The connector is defined in Amendment 1 to IEC 61076-2-101, 4-pin "Type D" Coding. The 4-pin M12 connector is suitable for use with 2-pair shielded or unshielded Ethernet cables only.

The M12-4 "D" coding connector shall be wired in accordance with the pin/wire assignments in Table 8-9.10.

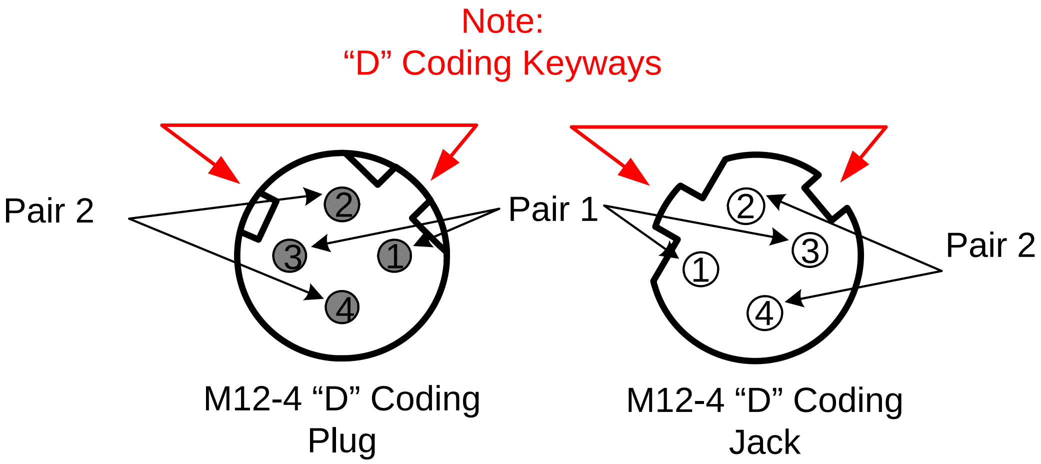
Table 8-9.10 M12-4 "D" Coding Cable Pin/Pair Cable Assignment

PIN	Signal Name	Color Code	Pair Assignment
1	TXD +	White Orange	Pair 1
3	TXD -	Orange	
2	RXD +	White Green	Pair 2
4	RXD -	Green	

Construction of crossover cables and conversion cables is permissible. Conversion cables constructed with 4-circuit M12-4 “D” coding connectors to 8-Way modular connectors shall be constructed from 2 pair cables containing the wire color codes defined in Table 8-9.10. The use of a 4 pair cable with 4 position M12-4 “D” coding connector is not permissible.

The 4-Pin M12 connector is suitable for use with 2 pair shielded or unshielded Ethernet cables only.

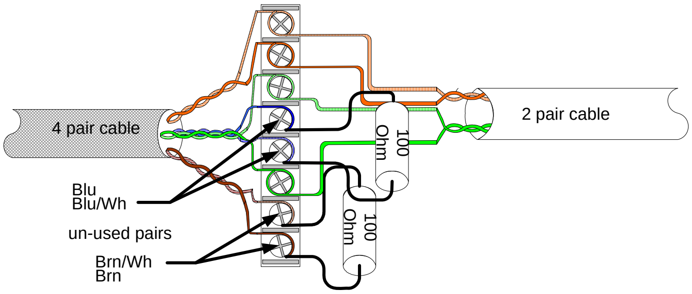
Figure 8-9.3 Plug Side and Jack Side Mating View



8-9.2.3.2 Mixing 2 and 4 Pair Cabling Components in a Channel

Cords using multi family connectors are permissible provided the number of conductors in the cable is equal to the minimum contact number of connector of the least contact assignment. For example, 4 pair cables shall not be used in the same channel with M12-4 “D” coding connectors. An exception to this requirement is where the unused pairs of active channel are terminated at their characteristic impedance. If terminated, a differential termination without reference to ground is preferred. Figure 8-9.4 shows the concept of terminating the un-used pairs of active channels in a 4 pair cable. The use of terminal strips is not recommended and is only here for illustration. Termination is required at both ends of the active cable where the pairs are not used.

Figure 8-9.4 Example of termination of un-used pair (for reference only)



8-9.2.3.3 Coupler

A coupler consists of two closely spaced (less than 10cm) electrically connected interfaces. Both interfaces are of the same physical mating interface.

A mated coupler shall conform to the transmission requirements of one connection of the appropriate media and category.

If the interfaces are not electrically close or the coupler does not meet the transmission of the appropriate media and category, then the coupler shall be counted as two mated connections.

8-9.2.3.4 Adapter

An adapter consists of two closely spaced electrically connected interfaces. They may be of different circuit counts. Both interfaces may be of the same or different physical mating interface; for example an M12-4 “D” coding connector to a RJ-45.

A mated adapter shall conform to the transmission requirements of one connection of the appropriate media and category.

If they not electrically close or the adapter dose not meet the transmission of the appropriate media and category, then the adapter shall be counted as two mated connections.

8-9.2.3.5 Bulkhead Connectors

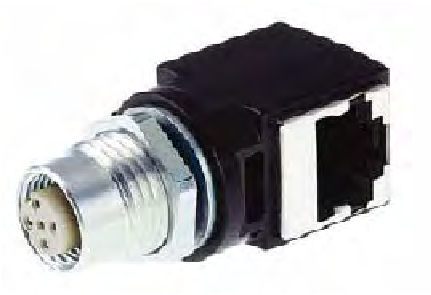
Bulkhead connectors are typically used at environmental or enclosure boundaries to facilitate connection and disconnection to the enclosure. A term used to define a mounting style of connectors. Bulkhead connectors are designed to be inserted into a panel cut-out from the rear (component side) or front side of the panel. The connector should be used where cables enter or exit the cabinet to maintain enclosure seal integrity. In addition they may be used to construct modular systems whereby providing modular connectivity.

Bulkhead connectors allow systems to be designed and built in modular configurations. This method should be considered based on user design and service preferences. Modularity provides quick deployment and ease of serviceability.

The designer shall be aware of metallic bulkhead feed troughs that connect the cabling at the enclosure wall. This may form a ground loop that could disrupt communications. Where a ground loop may be formed, a separate grounding conductor should be installed to provide an equal potential between the two points. An alternative method would be to isolate the bulkhead feed through using an insulator between the bulkhead feed through and the enclosure wall.

The transmission performance requirements for a bulkhead connector are defined in section 8-9.2.3.5. Figure 8-9.5 is an example of M12-4 D-coding EtherNet/IP bulkhead feed through connectors.

Figure 8-9.5 M12-4 to 8-way Modular Bulkhead



Consult the manufacturer’s data sheet for mounting hole cut out dimensions. Consider the panel minimum and maximum wall thickness of the enclosure when selecting a bulkhead.

8-9.2.3.6 Industrial Channel Length

8-9.2.3.6.1 Patch Cord Length

EtherNet/IP specifications limit the channel to 100 meters or up to 90 meters horizontal wiring with two 5-meter patch cords. Some applications will require longer patch cords. In these applications the total length of horizontal wiring must be adjusted to compensate for the added loss of each connector pair and additional patch cord length beyond 10m.

$$C = \frac{(102 - H)}{(1 + D)} \qquad (1)$$

Where:

C is the maximum combined length (m) of the work area cable, equipment cable, and patch cord.

H is the length (m) of the horizontal cable (H + C </= 100 m).

D is a de-rating factor for the patch cord type (0.2 for 24 AWG UTP/24 AWG ScTP and 0.5 for 26 AWG ScTP). The de-rating factors are based on COMMERCIAL cables. Other constructions, such as high flex, may have different performance. Consult the manufacturer for more information.

W is the maximum length (m) of the work area cable

T is the total length of horizontal, patch and equipment cords.

The maximum stranded cable length is limited to 85m for the channel with the standard 20% derating for standard stranded cables.

Table 8-9.11 Wire Type versus Length

	D	H	W	C	T
Patch Cable Gauge	**Patch Derating**	**Horizontal Length, (H+C<=100 m)**	**Patch Length**	**Total Length Patch and Equipment**	**Total length of patch, equipment and horizontal**
#24	0.2	100	0	0	100
#24	0.2	0	80	85	85
#24	0.2	25	59	64	89
#24	0.2	50	38	43	93
#26	0.5	0	63	68	68
#26	0.5	25	46	51	76
#26	0.5	50	30	35	85
#26	0.5	100	0	0	100

8-9.2.3.6.2 Channel Length Based on Temperature

Elevated temperatures cause higher signal loss in copper cables due to increased resistance. This added loss must be considered in addition to the type of copper cable (solid conductor horizontal or stranded conductor patch) to determine the maximum channel length. Shielded (STP) copper cable typically exhibit 0.2% attenuation increase for every 1° C temperature rise above 20° C to 60° C. Unshielded (UTP) Category 5e cables typically exhibit 0.4% attenuation increase for every 1° C temperature rise from 20° C to 60° C. Unshielded (UTP) Category 6 cable exhibit 0.4% attenuation increase for every 1° C temperature rise from 20° C to 40° C and 0.6% attenuation increase for every 1° C temperature rise from 40° C to 60° C, due to more copper and plastic content. The elevated temperature insertion loss is based on COMMERCIAL cables. Other constructions, such as high flex, may have different performance. The change in attenuation with temperatures beyond 60° C is product specific. Consult your supplier for more information.

The channel length and attenuation are linearly related, that is a 12% increase in attenuation reduces the channel length 12%. The following examples show how to calculate the maximum channel length for a given configuration and temperature.

AElev.Temp.=AIncrease Coefficient * Δ T

LElev.Temp.=AIncrease Coefficient * Δ T

Where: AElev.Temp = elevated temperature attenuation
AIncrease Coefficient = attenuation temperature coefficient
Δ T = change in temperature
LElev.Temp = elevated temperature maximum length

Assume you want to use solid conductor, Category 5e, horizontal cable at 60° C.

Note: The entire length should be treated as if the temperature is the worst-case temperature to ensure a conservative, simplified calculation.

You are limited to 100 meters based on the cable type. This distance must be de-rated to accommodate the elevated temperature. 60° C is 40° C above 20° C. 40° C times 0.4% equals 16% length reduction. The length reduction is calculated by taking the percent reduction times the cable type length limit: 16% x 100 meters = 16 meters.

The maximum channel length is calculated by subtracting the elevated temperature length reduction from the cable type channel limit: 100 meters – 16 meters = 84 meters. The maximum channel length for all solid, horizontal Cat 5e cable at 60° C is 84 meters.

For all stranded conductor patch Cat 5e at 60° C we have the following:

Cable type channel limit= 85 meters

Temperature change = 40C

Temperature coefficient = 0.4%

Total change = 16%

Length reduction = 13.6 meters

Maximum channel length for all stranded, patch Cat 5 at 60° C is 68.7 meters.

For 25 meters solid, horizontal Cat 5e cable with some length of #24 AWG, stranded conductor, Cat 5e patch at 40° C we have the following:

- 25 meters of solid, horizontal cable at 40° C has the loss of 8% more length of cable, 25 x 1.08 = 27 meters effective length
- Based on 27 meters we can have the effective length of patch as, (102-27)/(1+0.2)=62.5
- Total effective maximum stranded, patch length = 62.5 meters
- 62.5 meters of stranded, Cat 5e patch has 8% more loss then the actual length at 20° C, 62.5/1.08 = 57.9 meters actual length.
- The actual maximum stranded length = 57.9 meters
- The total channel length limit is the sum of the actual solid, horizontal cable maximum length limit plus the actual stranded, patch cable maximum length limit, 25 + 57.9 = 82.9 meters
- The maximum channel length limit for 25 meters of solid conductor, horizontal Cat 5e cable is 82.9 meters at 40° C with a maximum of 57.9 meters of stranded conductor, Cat 5e patch cable.

8-9.2.3.7 Industrial Permanent Link

The length of a industrial permanent link is limited to that of 8-9.2.3.6 less the equivalent length of 10 meters of patch cords.

8-9.2.3.8 Number of connections in a channel:

The number of mated connections allowed in a channel is determined by the desired channel performance (Category) and the performance level of the components selected. A Mated Connection is defined as an electrically conductive communications path comprised of a mated jack and plug. Back to back jack bulkheads may be counted as one connection provided they meet the requirements of this chapter. Cable lengths between connecting hardware greater than 10cm shall be counted in the total channel/link appropriate cable length budget.

Alternate configurations should be field tested to ensure adequate performance. Table 8-9.12 provides guidance for connector cable performance levels to achieve a given category channel for more than 4 connections.

Table 8-9.12 Number of allowable Connections in a Channel

Desired Channel performance	Number of Mated connections	Category connector (required)	Category cable (required)
5e	6	6A	5e

Current studies show that:

a) A Category 5e channel topology can include up to 6-mated connections, where each mated connection meets minimum Category 6A performance.

b) Maximum distance between jack and jack of the bulkhead connection is 10 cm. If the distance is greater than 10 cm each plug/jack interface shall be considered as a separate mated connection.

In order to maintain Category 5e performance in the channel for more than 4 mated connections, Category 6A connections shall be used. See Table 8-9.13 for return loss and NEXT, transmission requirements for construction of higher count channels.

Table 8-9.13 Transmission Requirements for More Than 4-connections in a Channel

Desired Channel Class	Number of Connections	Required Minimum Connecting Hardware Return Loss (dB)	Required Minimum Connecting Hardware NEXT (dB)	Cable Category
5e	5 or 6	26-20 log(f/100)	54-20log(f/100)	CAT 5e

8-9.2.3.9 Bulkhead Feed Through and Cable Glands

8-9.2.3.9.1 Bulkhead Cable Glands

Bulkhead cable glands provide entry/exit passages for permanently installed cables. Bulkhead feed troughs and/or bulkhead connectors allow systems to be designed and built in modular configurations. This method should be considered based on user design and service preferences. Modularity provides quick deployment and ease of serviceability.

8-9.2.3.9.2 Channels Using Balanced Cabling Bulkhead Connections

Figure 8-9.6 shows an intermediate cabling channel and a floor distribution channel created using a fixed cable terminated at a closure bulkhead.

The length of the fixed cable used within a channel shall be determined by the equations shown in section 8-9.2.3.6.

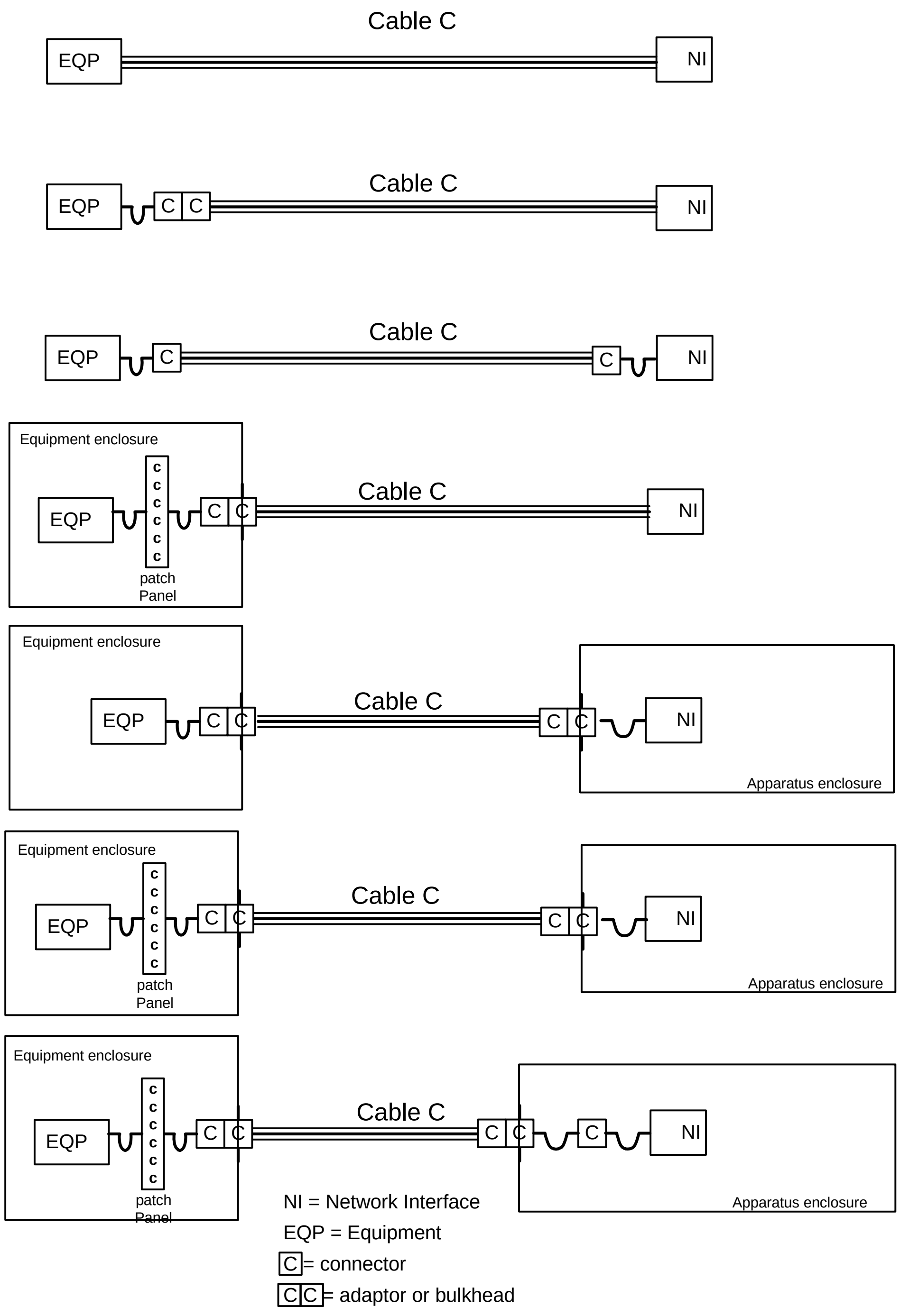
In section 8-9.2.3.6, it is assumed that;

a) The flexible cable within these cords has a higher insertion loss specification than that used in the fixed cable,

b) The cables within these cords in the channel have a common insertion loss specification.

The maximum length of the fixed cable will depend on the total length of cords to be supported within a channel. During the operation of the installed cabling, a management system should be implemented to ensure that the cords used to create the channel conform to the design rules for the floor, building or installation.

Figure 8-9.6 Channel Configurations

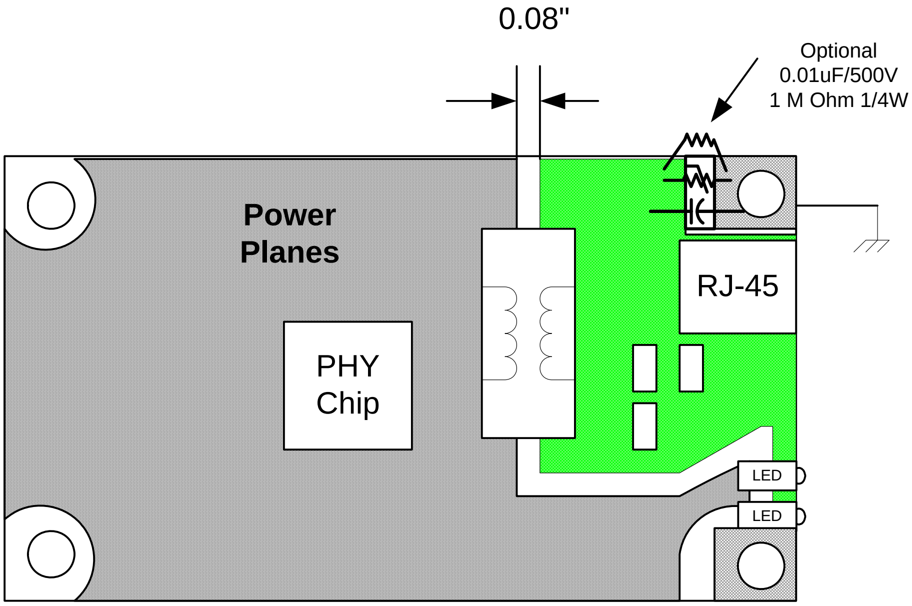


8-9.2.4 Industrial EtherNet/IP TP-PMD (Normative References)

A device that connects to the Industrial EtherNet/IP copper media shall conform to IEC 802.3 and ANSI X3.263 TP-PMD standard unless noted in this subclause.

The impedance at the media interface shall conform to ISO/IEC 802.3 (ANSI/IEEE Std 802.3) and IEEE Std 802.3u-1995 supplement with the exception of impedance tolerance. The temperature range and vibration shall be consistent with the targeted environment. In some cases it may be necessary to add components to protect the PMD from surge, ESD, EFT and conducted noises. Figure 8-9.10 is an example of how protection devices may be used to protect the EtherNet/IP device. In order to maximize the performance in noise, it is critical that the components selected for the PMD provide key characteristics. The transformer should (highly recommended) provide a minimum of 59dB common mode rejection (CMR) at 30 MHz. In addition special care in the circuit board trace parameters is needed to maintain impedance and noise immunity. Figure 8-9.7 is an example layout showing ground planes and isolation areas to help maintain noise immunity.

Figure 8-9.7 Example Reference Circuit Board Layout (informative)



A copper media attachment to an EtherNet/IP network shall support shielded and unshielded twisted pair technology. Active interfaces shall be compatible with ANSI/TIA/EIA-568-B.1 category 5e cabling system and cabling/component enhancements specified by this chapter. The signaling, encoding and coupling of these variants shall comply with the requirements of IEEE 802.3/TP-PMD and ANSI X3.263 TP-PMD standard subject to the deviations listed in this chapter. Likewise, the cable's electrical mechanical and environmental performance shall be as defined in section 8-9.1. The environmental classifications that support this requirement is defined the MICE table as defined by IEC 24702. The IEEE 802.3 standard defines many internal interfaces within the physical layer. EtherNet/IP products need not directly implement each of these interfaces, but shall behave as if these interfaces exist. These interfaces may be internal to the node and possibly internal to a semiconductor device.

At a minimum active interfaces shall support 10BASE T and 100BASE TX as defined by IEEE Std 802.3, 2000 Ed. and the ANSI X.3.263 TP-PMD. Two pair cabling will not support 100BASE-T4 therefore 100BASE-T4 interfaces are not supported by this standard.

8-9.2.4.1 Network Jacks for Active Devices

Active devices are end devices such as computers, sensors and HMIs. These devices generally support single network connections unless redundant. An active device with an embedded switch shall support AutoMDIX on its ports and be wired in accordance with this sub chapter. Repeaters should default to MDIX mode when AutoMDIX or Auto Negotiation is disabled. Active devices shall be fitted with one of the jacks defined in this chapter. Attached cables with flying leads or flying leads with jacks are not allowed. The jacks for active devices shall be wired in accordance with the pin definition described in Table 8-9.9 and Table 8-9.10.

8-9.2.4.2 Network Jacks for Connectivity Devices (repeaters)

Connectivity devices are active devices used to control the flow of data throughout the infrastructure. For example connectivity devices are classified as repeaters, switches, routers and bridges. These devices may have one or more of the following ports, LAN, WAN, Uplink. Figure 8-9.8 shows the relationship between LAN and WAN/Uplink ports for connectivity and active devices. Connectivity devices such as Switches, routers and bridges shall be fitted with jacks. The LAN side of the connectivity devices shall be wired in accordance with Table 8-9.14 and Table 8-9.15 or provide AutoMDIX. If the connectivity device supports AutoMDIX, then it shall default to MDIX state when AutoMDIX is disabled. WAN ports including uplink ports shall be wired in accordance with Table 8-9.9 and Table 8-9.10.

Figure 8-9.8 Port Identification

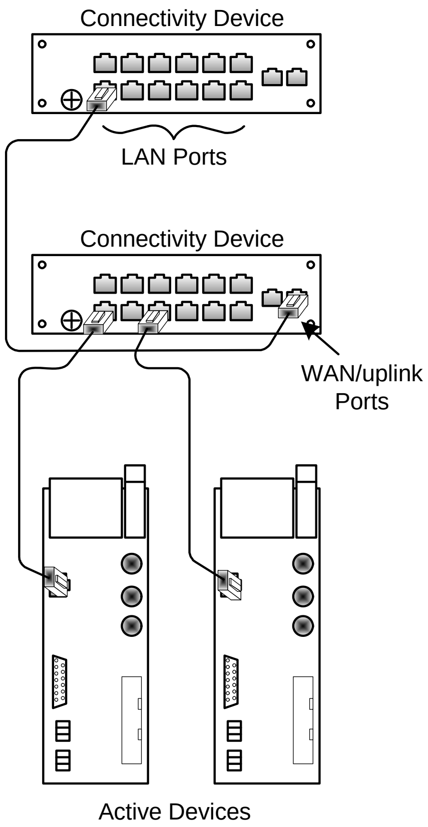


Table 8-9.14 8-Way Modular Jack Pin Assignment for LAN Ports

PIN	Signal Name
1	RXD+
2	RXD-
3	TXD+
4	NA[1]
5	NA[1]
6	TXD-
7	NA[1]
8	NA[1]

Table 8-9.15 M12-4 "D" Coding Jack Pin Assignment for LAN Ports

PIN	Signal Name
1	RXD +
3	RXD -
2	TXD +
4	TXD -

8-9.3 Termination for a 10/100 Mbps Interface with 4 Pair Support

Active devices shall use an appropriate termination technique such as found in Figure 8-9.9 for both the used and unused pairs. The unused pairs shall be terminated into their characteristic impedance at the device to prevent reflections of coupled energy. A common mode termination shall be used to terminate the TXD and RXD pairs. The resistor values may be adjusted between 50Ω and 75Ω to obtain 100Ω differential impedance and the appropriate common mode impedance.

Figure 8-9.9 PHY of Termination Example

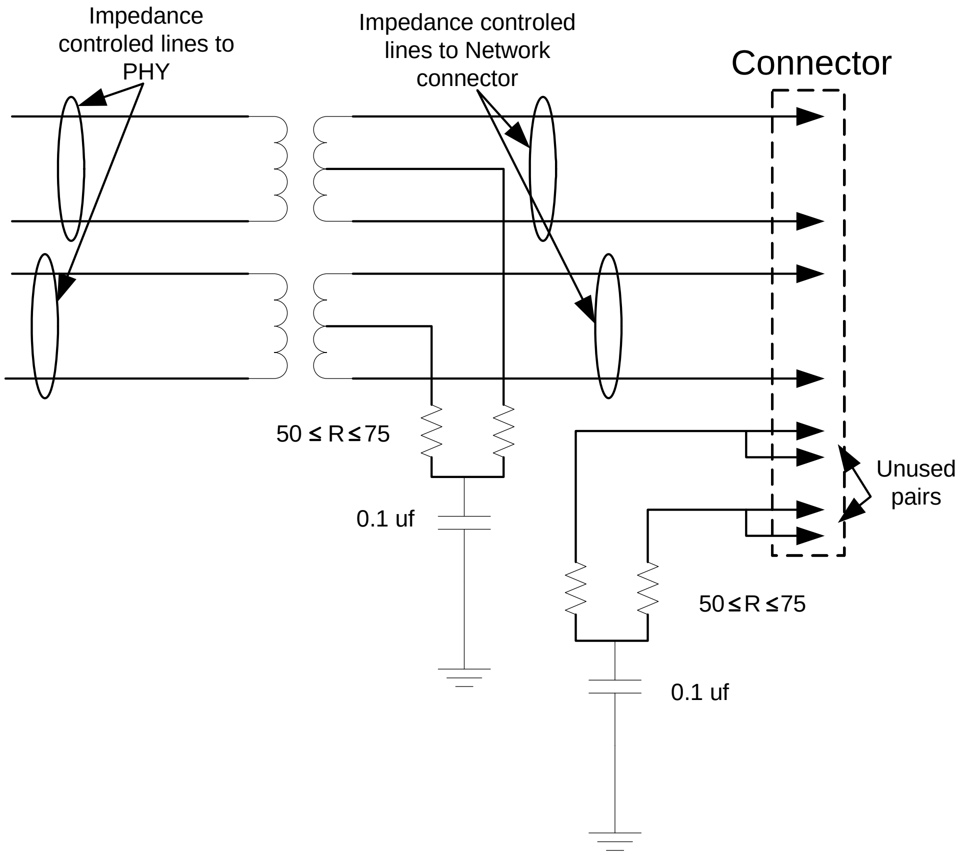
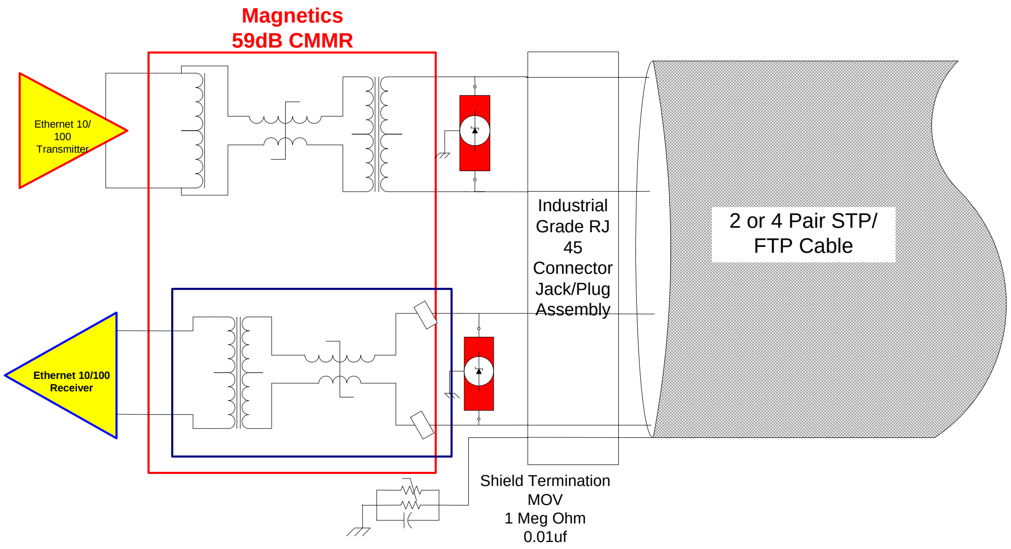

Impedance controled lines to PHY
Impedance controled lines to Network connector
Connector
50 ≤ R ≤ 75
0.1 uf
Unused pairs
50 ≤ R ≤ 75
0.1 uf

Figure 8-9.10 Example Physical Layer Block Diagram (informative)


Magnetics
59dB CMMR
Ethernet 10/ 100 Transmitter
Ethernet 10/100 Receiver
Industrial Grade RJ 45 Connector Jack/Plug Assembly
2 or 4 Pair STP/ FTP Cable
Shield Termination
MOV
1 Meg Ohm
0.01uf

8-9.4 Shield Grounding

8-9.4.1 Connectivity Device (Switch, Hub, Bridges, Routers, etc.)

The communications shield shall be terminated directly to earth ground in accordance with IEEE 802.3.

8-9.4.2 Two Port Devices

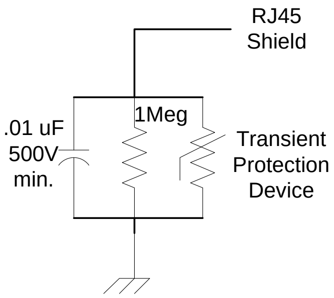
Two port devices that provide two active ports should not use ganged RJ 45 jacks with common shields; doing so will propagate the grounds and potential cause ground loops in the system. Termination of the shield shall be in accordance with Active Devices.

8-9.4.3 Active Devices (sensor, PLC etc.)

To prevent ground loops caused by shielded cables, devices shall not connect the shield directly to ground. Industrial EtherNet/IP devices shall provide shield terminated as detailed in Figure 8-9.11. For Commercial active devices where the shielded RJ 45 connector provides direct ground, the shield should be disconnected at the active device end of the channel as shown in Figure 8-9.12 and Figure 8-9.13.

The shield termination for Industrial EtherNet/IP active devices, using a parallel resistor and capacitor is shown in Figure 8-9.11.

Figure 8-9.11 Shield Termination for Devices



If the active device provides direct connection to ground through the RJ-45 connector, then the shield shall not be connected at the RJ45 plug. Figure 8-9.12 and Figure 8-9.13 are examples of how to break the shield at a device that is directly grounded.

Figure 8-9.12 Example Shield Termination

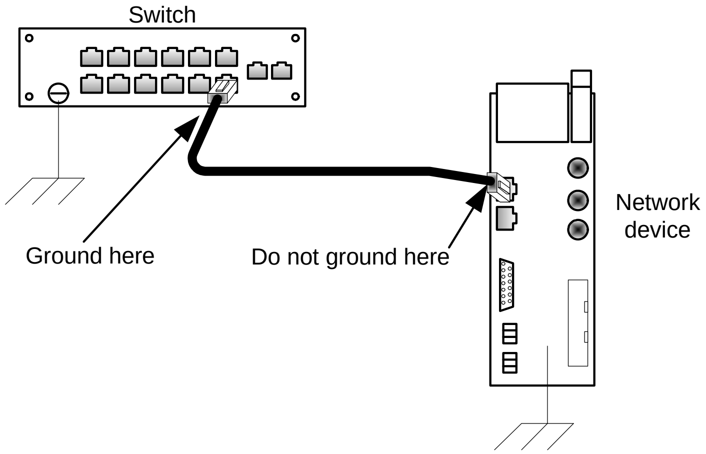
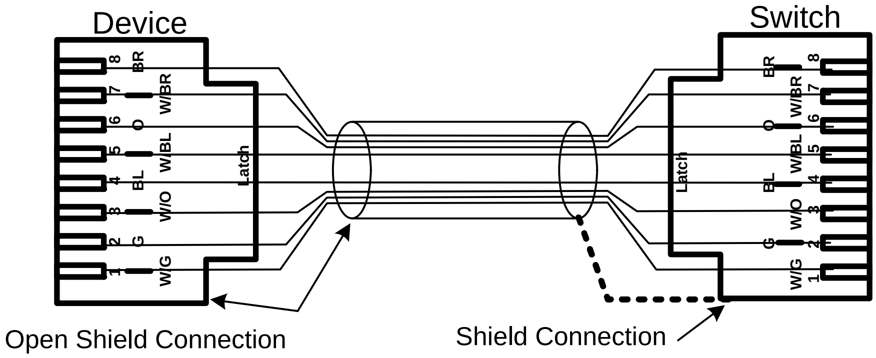


Figure 8-9.13 Shield Termination for Commercial Devices



8-9.5 Fiber Media Variant

8-9.5.1 Cables

The following fiber optic cables are supported by this standard

8-9.5.1.1 Multi Mode Fiber Optic Cables

The following multimode fiber optic cables are in accordance with ANSI/TIA/EIA 568-B.3.

- 62.5/125µm
- 50/125µm

8-9.5.1.2 Single mode fiber optic 9/125µm

The single mode fiber shall conform to ANSI/EIA/TIA 568-B.3 standard.

8-9.5.1.3 1mm Plastic Optical Fiber (POF)

TBD

Table 8-9.16 Recognized Fiber Cables

Fiber Type	Supported Fiber	Wavelength (typical)
Multimode	50/125µm, 62.5/125µ	1310 nm
Singlemode	9/125 µm	1310 nm
POF	TBD	650 nm

8-9.5.2 Connectors

The fiber media attachment to an EtherNet/IP network shall be limited to the LC, SC and ST variants. The signaling and coupling for the fiber types shall be as specified in the IEEE 802.3 standard subject to the deviations listed in this section (section 8-9.5). The SC and ST connectors are allowable, however are not recommended for new designs. SC and ST connectors are legacy connectors with limited interfaces available. EtherNet/IP devices utilizing the LC transceivers shall have duplex jacks with center spacing compatible with the FOCIS standard of 0.246 inches (6.25mm). Permanently attached fiber pigtails shall not be used.

8-9.5.2.1 Non-Sealed Connectors

Table 8-9.17 Non-sealed Connector Types and Reference Standards

Non-sealed connector type	Reference Standards
LC, ST, SC	ANSI/TIA/EIA-568-B.3, FOCIS

Table 8-9.18 LC, SC and ST Connector Insertion Loss

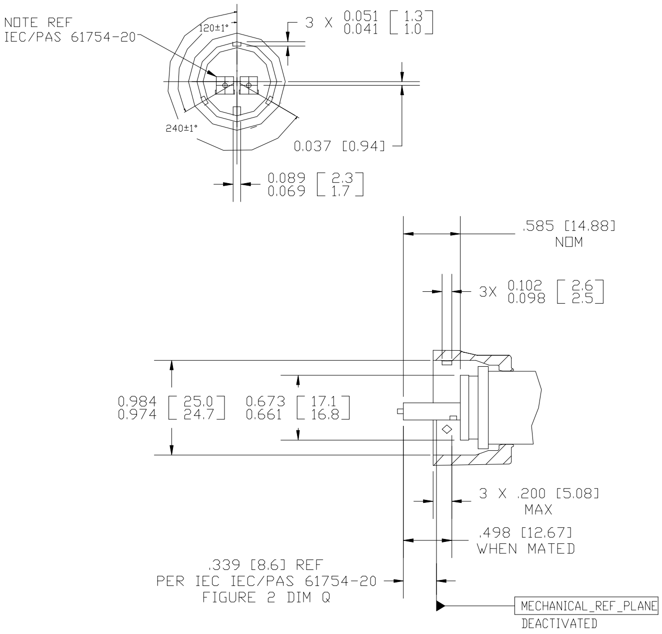
Fiber Medium/Wave length	650nm	1310nm
9/125µm	Not supported	0.75 dB max.
50/125µm	Not supported	0.75 dB max.
62.5/125µm	Not supported	0.75 dB max.
1mm POF	ffs	Ffs

8-9.5.2.2 Sealed Industrial LC Connectors

Fiber optic connector designs shall meet the requirements of the corresponding ANSI/TIA/EIA (Fiber Optic Connector Intermateability Standard (FOCIS) documents). In the case where the LC fiber optic connector is placed into the IP65/67 shell or enclosure whereby the latch is defeated, the FOCIS requirements may not be applicable. See Table 8-9.18 LC, SC and ST Connector Insertion Loss.

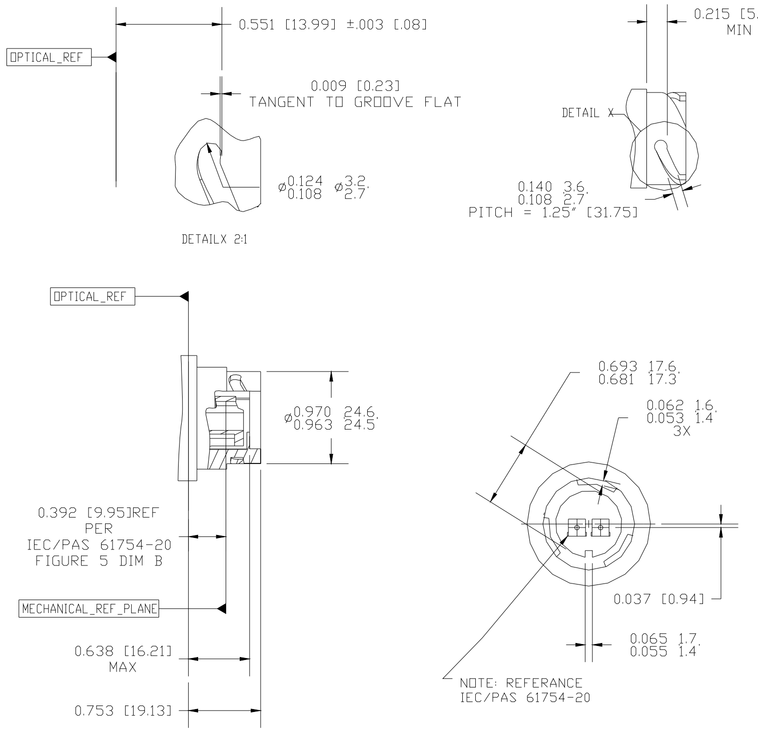
The following sealed plug drawing in Figure 8-9.14 defines the plug. The plug is fully compatible with standard off-the-shelf plugs with the exception of the defeated locking mechanism when placed in the Variant 1 housing. The dimensions are expressed in inches [mm].

Figure 8-9.14 Sealed Plug



The following sealed outlet/jack drawing in Figure 8-9.15 defines the outlet to maintain compatibility for mating and sealing amongst various vendors who make one or both parts. The outlet is fully compatible with off-the-shelf fiber optic LC plugs and jacks. The dimensions are expressed in inches [mm].

Figure 8-9.15 Sealed Outlet



8-9.5.2.3 Sealed M12 Circular Fiber Optic Connector

TBD

8-9.5.3 Fiber PMD

The IEEE 802.3 standard defines many internal interfaces within the Physical Layer. EtherNet/IP products need not directly implement each of these interfaces, but shall behave “as if” these interfaces existed. These interfaces may be internal to the node and possibly internal to a semiconductor device. There are three media variants supported:

- 100BASE-LX10 using Single mode silica fibers;
- 100BASE-FX using multi mode silica fibers;
- 100 Mbps using Multi mode graded index plastic optical fiber, compatible with signaling of 100BASE-FX.

Other data rates are possible; however they are outside the scope of this standard and will not be compatible with the 100 Mbps fiber optic systems.

8-9.5.4 Fiber Optic Transceivers

8-9.5.4.1 Single mode

Fiber transceivers shall conform to IEEE 802.3 for 100BASE-LX10 (Ethernet in the First Mile) using SM Silica fibers. Additional requirements can be found in ISO/IEC 9314-3 Information processing systems-Fiber distributed Data Interface (FDDI)- part 3 Physical Layer Medium Dependant (PMD) standards with the exception of the transceiver which provides single mode coupling using the same wavelength of 1310nm as the multimode variant. The data rate shall be 100 Mbps.

The optical cabling power budget shall be a minimum of 10 dB.

Connectors supported (new designs):

- LC defined in 8-9.5.2.1
- Sealed LC defined in 8-9.5.2.2

The SC and ST connectors are allowable, however are not recommended for new designs.

8-9.5.4.2 Multimode

Fiber transceivers shall conform to IEEE 802.3 for 100BASE-FX when using multimode fibers. Additional requirements can be found in ISO/IEC 9314-3 Information processing systems-Fiber distributed Data Interface (FDDI)- part 3 Physical Layer Medium Dependant (PMD) standards.

The optical cabling power budget shall be a minimum of 11dB.

Connectors supported (new designs);

- LC defined in 8-9.5.2.1
- Sealed LC defined in 8-9.5.2.2

The SC and ST connectors are allowable, however are not recommended for new designs.

Volume 2: EtherNet/IP Adaptation of CIP

Chapter 9: Indicators & Middle Layers

Contents

9-1 Introduction

Chapter 9 specifies the standard appearance and behavior of EtherNet/IP diagnostic LEDs. This chapter also specifies TCP/IP requirements of EtherNet/IP devices.

9-2 Data Link Layers

Though this specification is called “EtherNet/IP”, Ethernet is technically not required. The EtherNet/IP protocol may be used on any media that supports the transmission of the Internet Protocol.

NOTE: For example, the EtherNet/IP protocol could be used over FDDI, modem lines (SLIP or PPP), ATM, etc.

When any particular medium is used, it shall be used in accordance to commonly accepted standards. In particular, when Ethernet is used, it shall be used as defined by the IEEE 802.3 specification.

9-3 Requirements for TCP/IP Support

In addition to the various requirements set forth in this specification, all EtherNet/IP hosts are required to have a minimally functional TCP/IP protocol suite and transport mechanism. The minimum host requirements for EtherNet/IP hosts shall be those covered in RFC-1122, RFC-1123, and RFC-1127 and the subsequent documents that may supersede them. Whenever a feature or protocol is implemented by an EtherNet/IP host, that feature shall be implemented in accordance to the appropriate RFC documents, regardless of whether the feature or protocol is considered required or optional by this specification. The Internet and the RFCs are dynamic. There will be changes to the RFCs and to the requirements included in this section as the Internet and this specification evolves and these changes will not always provide for backward compatibility.

All EtherNet/IP devices shall at a minimum support:

- Internet Protocol (IP version 4) (RFC 791)
- User Datagram Protocol (UDP) (RFC 768)
- Transmission Control Protocol (TCP) (RFC 793)
- Address Resolution Protocol (ARP) (RFC 826)
- Internet Control Messaging Protocol (ICMP) (RFC 792)
- Internet Group Management Protocol (IGMP) (RFC 1112 & 2236)
- IEEE 802.3 (Ethernet) as defined in RFC 894

NOTE: Although the encapsulation protocol is suitable for use on other networks besides Ethernet that support TCP/IP and products may be implemented on these other networks, conformance testing of EtherNet/IP products is limited to those products on Ethernet. Other suitable networks include:

- Point to Point Protocol (PPP) (RFC 1171)
- ARCNET (RFC 1201)
- FDDI (RFC 1103)

NOTE: EtherNet/IP devices are encouraged but not required to support other Internet protocols and applications not specified here. For example, may support HTTP, Telnet, FTP, etc. This specification makes no requirements with regards to these protocols and applications.

9-4 Indicators

9-4.1 Required Indicators

A product need not have indicators to be compliant with this specification. However, to be compliant with the Industrial Performance Level described in Chapter 8, a product shall support both the module status and network status indicators as defined by sections 9-4.2, 9-4.3 and 9-4.4.

If a product does support any of the indicators described here, they must adhere to the specifications described in this section (section 9-4).

Two types of status indicators may be provided:

- One module status indicator;
- One network status indicator;

Additional indicators may be present; however, the naming and symbol conventions of the standard indicators shall not be employed for other indicators.

NOTE: Indicators, typically implemented as LEDs, help maintenance personnel to quickly identify a faulty unit or media. As such, red indicators are used to indicate a fault condition.

NOTE: Products are encouraged to have an indicator that displays the state of link (for example, link status, tx/rx, collision, etc.) following generally accepted industry practices (as used in devices such as switches).

9-4.2 Common Indicator Requirements

9-4.2.1 Applicability of Common Requirements

The common indicator requirement shall only apply to indicators for which requirements are specified in this standard.

9-4.2.2 Visibility of Indicators

Indicators shall be viewable without removing covers or parts from the equipment. Indicators shall be easily seen in normal lighting. Any labels and icons shall be visible whether or not the indicator is illuminated.

9-4.2.3 Indicator Flash Rate

Unless otherwise indicated, the flash rate of all indicators is approximately 1 flash per second. The indicator should be on for approximately 0.5 second and off for approximately 0.5 second. This flash rate specification only applies to the indicators specified in this chapter.

9-4.2.4 Indicators at Power Up

An indicator test is to be performed at power-up. To allow a visual inspection, the following sequence shall be performed:

- Turn first indicator Green, all other indicators off

- Leave first indicator on Green for approximately 0.25 second
- Turn first indicator on Red for approximately 0.25 second
- Turn first indicator on Green
- Turn second indicator (if present) on Green for approximately 0.25 second
- Turn second indicator (if present) on Red for approximately 0.25 second
- Turn second indicator (if present) Off

If other indicators are present, test each indicator in sequence as prescribed by the second indicator above. If a Module Status indicator is present, it shall be the first indicator in the sequence, followed by any Network Status indicators present. After completion of this power up test, the indicator(s) shall turn to a normal operational state.

9-4.3 Module Status Indicator

9-4.3.1 Description

The indication of module status shall require a single bicolor (red/green) indicator that represents the state of the entire product.

NOTE: A product with more than one communication port would have only one module status indicator, but more than one network status indicator (one per port).

9-4.3.2 Labeling

The module status indicator shall be labeled with one of the following:

- “MS”;
- "Mod";
- “Mod Status”;
- “Module Status”.

9-4.3.3 States

The module status indicator shall be in one of the following states:

Table 9-4.1 Module Status Indicator

Indicator state	Summary	Requirement
Steady Off	No power	If no power is supplied to the device, the module status indicator shall be steady off.
Steady Green	Device operational	If the device is operating correctly, the module status indicator shall be steady green.
Flashing Green	Standby	If the device has not been configured, the module status indicator shall be flashing green.
Flashing Red	Minor fault	If the device has detected a recoverable minor fault, the module status indicator shall be flashing red. **NOTE**: An incorrect or inconsistent configuration would be considered a minor fault.
Steady Red	Major fault	If the device has detected a non-recoverable major fault, the module status indicator shall be steady red.
Flashing Green / Red	Self-test	While the device is performing its power up testing, the module status indicator shall be flashing green / red.

9-4.4 Network Status Indicator

9-4.4.1 Description

The indication of network status shall require a single bicolor (red/green) indicator that represents the state of a single communication port.

NOTE: A product with more than one communication port would have only one module status indicator, but more than one network status indicator (one per port).

9-4.4.2 Labeling

The network status indicator shall be labeled with one of the following:

- “NS”;
- “Net”;
- “Net Status”;
- "Network Status".

9-4.4.3 States

The network status indicator states shall be as follows:

Table 9-4.2 Network Status Indicator

Indicator state	Summary	Requirement
Steady Off	Not powered, no IP address	If the device does not have an IP address (or is powered off), the network status indicator shall be steady off.
Flashing Green	No connections	If the device has no established connections, but has obtained an IP address, the network status indicator shall be flashing green.
Steady Green	Connected	If the device has at least one established connection (even to the Message Router), the network status indicator shall be steady green.
Flashing Red	Connection timeout	If one or more of the connections in which this device is the target has timed out, the network status indicator shall be flashing red. This shall be left only if all timed out connections are reestablished or if the device is reset.
Steady Red	Duplicate IP	If the device has detected that its IP address is already in use, the network status indicator shall be steady red.
Flashing Green / Red	Self-test	While the device is performing its power up testing, the network status indicator shall be flashing green / red.

Volume 2: EtherNet/IP Adaptation of CIP

Chapter 10: Bridging & Routing

Contents

10-1 Introduction

This chapter of the EtherNet/IP specification contains additions to the definition of CIP bridging and routing that are EtherNet/IP specific. At this time, no such additions exist.

This page is intentionally left blank

Volume 2: EtherNet/IP Adaptation of CIP

Appendix A: Explicit Messaging Services

Contents

A-1 Introduction

This chapter of the EtherNet/IP specification contains additions to the definition of CIP explicit messaging services that are EtherNet/IP specific. At this time there are no such additions.

This page is intentionally left blank

Volume 2: EtherNet/IP Adaptation of CIP

Appendix B: Status Codes

Contents

B-1 Introduction

This chapter of the EtherNet/IP specification contains additions to the definition of CIP error codes that are EtherNet/IP specific. At this time there are no such additions.

This page is intentionally left blank

Volume 2: EtherNet/IP Adaptation of CIP

Appendix C: Data Management

Contents

C-1 Introduction

This chapter of the EtherNet/IP specification contains additions to the CIP Data Management specification that are EtherNet/IP specific. At this time there are no such additions.

This page is intentionally left blank

Volume 2: EtherNet/IP Adaptation of CIP

Appendix D: Engineering Units

Contents

D-1 Introduction

This chapter of the EtherNet/IP specification contains additions to the list of CIP engineering units that are EtherNet/IP specific. At this time, there are no such additions.

This page is intentionally left blank