



# **General Recommendations for EtherNet/IP Developers**

**Version 2.00  
January 28, 2005**

**Published by  
EtherNet/IP Implementors Workshop  
ODVA**

**Document Revision Log**

<b>Revision</b>	<b>Sections</b>	<b>Remarks</b>	<b>Date</b>	<b>Author</b>
0		Initial Release	May 10, 2001	Enabler JSIG
0.01		Revise entire document, change title and add interoperability recommendations	March 10, 2004	K. Knake
0.02		Revision before EtherNet/IP WS#13	Sept 13, 2004	P. Green
0.03		Revision after WS#13	Nov 3, 2004	P. Green
0.04		Revision after WS#14	Jan 17, 2005	P. Green
0.05		Updates from review comments	Jan 18, 2005	P. Green
2.00		Final updates after review period	Jan 28, 2005	P. Green

**Table of Contents**

1	Introduction .....	1
1.1	Document Organization .....	2
1.2	Companion Documents .....	3
2	Technology Overview .....	4
2.1	Overview of TCP/IP .....	4
2.2	Overview of EtherNet/IP .....	6
2.2.1	Introduction .....	6
2.2.2	Device Applications .....	6
2.2.3	Architectural Design .....	7
2.2.4	Network Topology .....	7
2.2.5	EtherNet/IP Transmission Types .....	8
2.2.6	EtherNet/IP Device Classes .....	9
2.3	Levels of Communication .....	10
2.3.1	Categories of Control Network Messages .....	10
2.3.2	Quality of Service .....	12
2.3.3	Quality of Service Relationships .....	14
2.3.4	How do we decide if Ethernet is the right network? .....	17
3	Recommendations .....	20
3.1	Recommendations for Physical Layer Components .....	20
3.1.1	Controller Chips .....	20
3.1.2	Magnetics .....	21
3.1.3	Protection Circuitry .....	21
3.1.4	Connectors (Jacks and Plugs) .....	23
3.1.5	Cables .....	23
3.1.6	Device Design (Physical Layer) .....	23
3.2	Recommendations for TCP/IP Stack .....	28
3.2.1	TCP/IP Stack Recommendations for All EtherNet/IP Devices .....	28
3.2.2	TCP/IP Stack Recommendations for EtherNet/IP Scanners .....	28
3.3	Recommendations for Operating System .....	29
3.4	Processor Recommendations .....	29
3.5	EtherNet/IP Protocol Stacks .....	29
4	Organizational Support .....	30
4.1	JSIGs .....	30
4.2	Enabler Technologies .....	31
4.3	EtherNet/IP Implementors Workshop .....	31
4.3.1	EtherNet/IP Recommendations Papers .....	31
4.3.2	EtherNet/IP Plug Fests .....	32
4.3.3	North American and European Tours .....	32
5	Terms and Definitions .....	33

### 1 Introduction

Interest in networking continues to grow rapidly as more companies and industries move their products and services to the Internet. Within the office environment, Ethernet is the most popular medium for connecting computers to each other and the Internet. This growing interest has not been lost on the industrial communications industry. There have been increasing inquiries from vendors, OEMs and end users about using Ethernet as the communication network for factory floor control.

For devices to converse and interoperate over Ethernet (or any network), a common application layer is needed. Although common protocols for file transfer (FTP), e-mail (SMTP), World Wide Web (HTTP) and others have been established for several applications, the situation is not so simple in the area of industrial automation. Each vendor of automation equipment that runs over Ethernet TCP/IP has implemented its own application layer. They each speak a different language. As a result, a standard application layer with common object models and universal device profiles, doesn't exist. Ethernet users are currently tied to proprietary solutions and aren't able to benefit from the best-in-class and best-in-value options offered by an open market. **EtherNet/IP**, which is based on TCP/IP and UDP/IP protocols, fills the void by delivering interoperable Ethernet products. This is possible because EtherNet/IP uses a proven and open application layer standard. An independent EtherNet/IP conformance test labs has been established to promote conformance to the specification and promote multi-vendor interoperability.

The benefits of bringing EtherNet/IP to the plant floor are reduced cost, ease of integration and the intriguing idea of building Internet features into factory automation equipment.

The ODVA (<http://www.ODVA.org>) and ControlNet International (<http://www.ControlNet.org>) recognize the interest and potential of implementing existing fieldbus protocols on top of Ethernet. They have taken the joint initiative to identify best practices and methods for implementing the DeviceNet/ControlNet application layer protocols over a TCP/IP network. One of the results of this effort has been the development of **EtherNet/IP**, an industrial automation protocol based on CIP (Common Industrial Protocol) which is used in both DeviceNet and ControlNet. EtherNet/IP is defined as CIP operating on top of the TCP/IP protocol suite on Ethernet. Another result of the initiative is the publication of papers to assist vendors and end users in the development and use of EtherNet/IP products.

This paper provides recommendations to vendors interested in developing EtherNet/IP products for the industrial control markets. The recommendations are intended to help vendors develop EtherNet/IP products that best meet the needs of end users. Many of these recommendations are based on experiences from early adopters and EtherNet/IP technology inventors.

### 1.1 Document Organization

This document provides an overview of the EtherNet/IP protocol technology, then looks at specific features that are important in the implementation of an EtherNet/IP device. Specifically, the document is organized into the following sections.

- **Overview of the Technology** - This section provides an overview of Ethernet, TCP/IP, and EtherNet/IP and discusses its use in industrial control applications. The following items are covered in this section:
  - **Overview of TCP/IP** - A brief discussion of the TCP/IP suite.
  - **Overview of EtherNet/IP** - A brief discussion of the EtherNet/IP protocol.
  - **Ethernet and TCP/IP in Industrial Control Applications** - A look at the applicability of Ethernet and TCP/IP to industrial control applications.
  - **Levels of Communication** - A discussion of the requirements of communication for industrial control and relates these to Ethernet, TCP/IP, and EtherNet/IP.
- **Recommendations** - This section comprises a set of recommendations for EtherNet/IP implementations, ranging from physical layer to the TCP/IP stack and operating system. The following items are covered in this section:
  - **Physical Layer Components** - Feature recommendations for physical layer components, including controllers, physical layer circuitry, connectors, and cabling.
  - **TCP/IP Stack** - Recommendations for features and functionality that are desirable in a TCP/IP protocol suite to be used for EtherNet/IP.
  - **Operating Systems** - Feature recommendations for operating systems to be used on an EtherNet/IP device.
  - **Processor** - Guidelines for processor selection for EtherNet/IP devices.
- **Organizational Support** - This section discusses the roles that the various organizations and SIGs play in the support of EtherNet/IP.
- **Terms and Definitions** - This section provides a glossary for terms used when discussing EtherNet/IP and CIP.

### 1.2 Companion Documents

The purpose of this document is to cover general recommendations. More specific recommendations, and greater details can be found in the following companion documents. These documents are a result of work generated by the ongoing series of EtherNet/IP Implementors Workshops.

Implementors Workshop documents can be found on the ODVA web site ([www.odva.org](http://www.odva.org)), under EtherNet/IP Papers and Presentations.

*Recommended Functionality for EtherNet/IP Devices, Version 1.0, June 10, 2004.*

This document recommends functionality related to the EtherNet/IP protocol implementation in EtherNet/IP devices. The recommendations are being made to help promote interoperability between devices and provide a minimum level of capability required for user applications.

*Recommended IP Addressing Methods for EtherNet/IP Devices, Version 1.0, June 10 2003.*

This document recommends common mechanisms related to IP address assignment and management for EtherNet/IP devices.

*IPv4 Address Conflict Detection for EtherNet/IP Devices, Version 1.0, July 27 2004.*

This document specifies a common mechanism by which EtherNet/IP devices can detect IP address conflicts (commonly referred to as “duplicate IP addresses”).

*Performance Terminology for EtherNet/IP Devices, Version 1.0, September 16, 2004.*

This document discusses and defines a number of terms that are used in describing performance tests and the results for EtherNet/IP devices.

*Recommended Operation for Switches Running Relay Agent and Option 82.*

This document details the DHCP Option 82 process and provides guidance for switch vendors to provide a design compatible with the EtherNet/IP protocol.

The following documents are in development in the workshop and will be released at a later date.

*Performance Methodology for EtherNet/IP Devices.*

This document discusses and defines a number of test methodologies that are used in describing the performance of EtherNet/IP devices.

*Recommended Functionality for DHCP Clients.*

This document recommends DHCP client functionality as it pertains to EtherNet/IP device IP addressing and configuration.

## 2 Technology Overview

### 2.1 Overview of TCP/IP

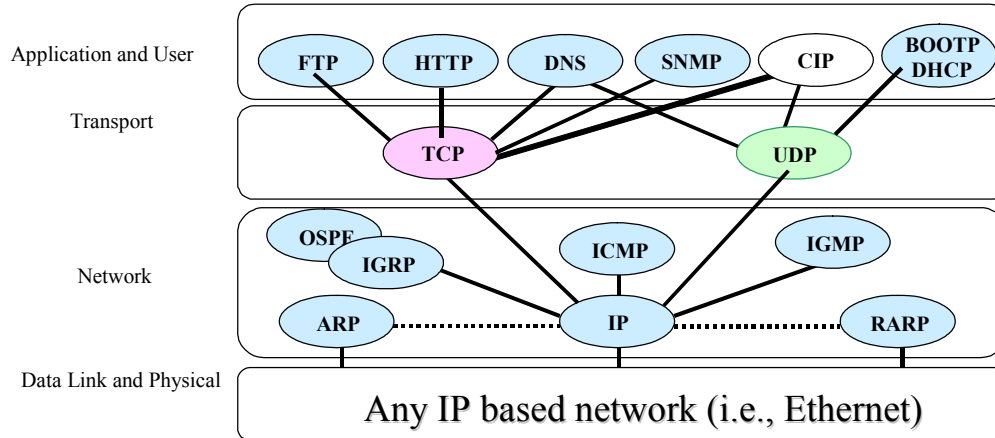
TCP/IP stands for “Transmission Control Protocol / Internet Protocol”. As is normal within the Internet community, we will use “TCP/IP” to refer to the entire suite of internetworking protocols that provide the networking between machines and sub-networks and is the common language of the Internet and its applications. TCP/IP can operate over many different physical networks. This paper is concerned with TCP/IP operating over Ethernet. The protocols providing the foundation of a TCP/IP network are:

- **IP – (Internet Protocol)** provides the means to transfer a packet of data from a source node to a destination node either within the same local network or to a distant network. IP also provides the means to transmit a packet of data to multiple destinations through the use of broadcast or multicast destination addresses.
- **ICMP – (Internet Control Message Protocol)** provides the means to send control and diagnostic information between nodes on a network.
- **IGMP – (Internet Group Multicast Protocol)** provides support for the management of multicast groups within a TCP/IP network. A single message transmitted to a multicast destination address may be received by any number of destinations. IGMP is used by destination to join that address. Any node may belong to zero, one or more multicast groups depending on the capability of its TCP/IP stack and Ethernet chip/driver.
- **UDP – (User Datagram Protocol)** provides best effort non-guaranteed delivery of data to an end point within a node. A UDP message requires less CPU overhead to process and route than would a TCP message.
- **TCP – (Transmission Control Protocol)** provides connection-oriented transfer of data between exactly two end points. Unlike UDP, data transfers are verified by TCP. This verification requires more CPU overhead.

There are many other protocols available within the TCP/IP suite. The ones listed here provide the foundation upon which EtherNet/IP is built. TCP/IP, like most network protocols, is referred to as a “stack”. You will read or hear references to “protocol stacks” or “layers” whenever network software is discussed. These words are used because the different protocols that comprise TCP/IP are organized as layers stacked on top of each other. IP is the foundation upon which ICMP, IGMP, UDP and TCP are stacked. Higher level protocols are stacked on top of UDP and TCP. A packet of data must pass through the intermediate layers of protocols in order to reach its final destination.

## General Recommendations for EtherNet/IP Developers

The following diagram illustrates the stack concept. Notice how the Common Industrial Protocol (CIP) straddles both UDP and TCP. Most high level protocols are stacked on top of either TCP or UDP. CIP uses both UDP and TCP for conveying implicit and explicit CIP messages. Explicit messages use TCP to pass one time messages and commands from one node to another. Implicit messages use UDP for control messages.



**Figure 1 TCP/IP Protocol Suite Stack**

Further information on TCP/IP can be found at the Internet Engineering Task Force web site:  
<http://www.ietf.org>

### 2.2 Overview of EtherNet/IP

#### 2.2.1 Introduction

Ethernet Industrial Protocol (EtherNet/IP) is an open industrial networking standard that supports real-time I/O messaging, explicit message exchange or both and uses commercial off-the-shelf Ethernet communication chips and physical media. Using Ethernet products is not only a common trend in technology today but provides users the ability to access device-level data all the way from the Internet. EtherNet/IP emerged due to the high demand for using the Ethernet network for control applications.

EtherNet/IP is an open network because it uses:

- IEEE 802.3 Physical and Data Link standard.
- Ethernet TCP/IP protocol suite (Transmission Control Protocol/Internet Protocol), the Ethernet industry standard.
- Common Industrial Protocol (CIP), the protocol that provides real-time I/O messaging and information / peer-to-peer messaging. ControlNet (IEC61158 Part 6) and DeviceNet (EN50325 part 2, IEC62026 part 3) networks also use CIP.

Because Ethernet technology and standard protocol suites such as TCP/IP have been published for public use, standardized software tools and physical media have been mass-produced and are readily available offering two benefits: known technology and accessibility. To make EtherNet/IP successful, CIP has been added on top of TCP/UDP/IP to provide a common application layer. Therefore, when selecting an EtherNet/IP product, you are choosing a product with CIP capabilities. Additionally, EtherNet/IP uses the producer/consumer network model, as do DeviceNet and ControlNet networks. With the introduction of Ethernet switch technology and full-duplex data transmission, the occurrence of data collisions is theoretically eliminated, and performance is drastically enhanced on an EtherNet/IP network.

#### 2.2.2 Device Applications

Typical devices communicating across an EtherNet/IP network include:

- Mainframe Computers
- PLC Processors
- Robots
- HMI
- I/O and I/O Adapters

Target applications include:

- Plant management system interaction with MES (Manufacturing Execution Systems), material handling, SCADA applications, ...
- Configuration, data collection, and control on a single high-speed network
- Time-critical applications with no established schedule (like ControlNet provides)

### 2.2.3 Architectural Design

The following diagram illustrates the relationship between CIP, EtherNet/IP, and TCP/IP.

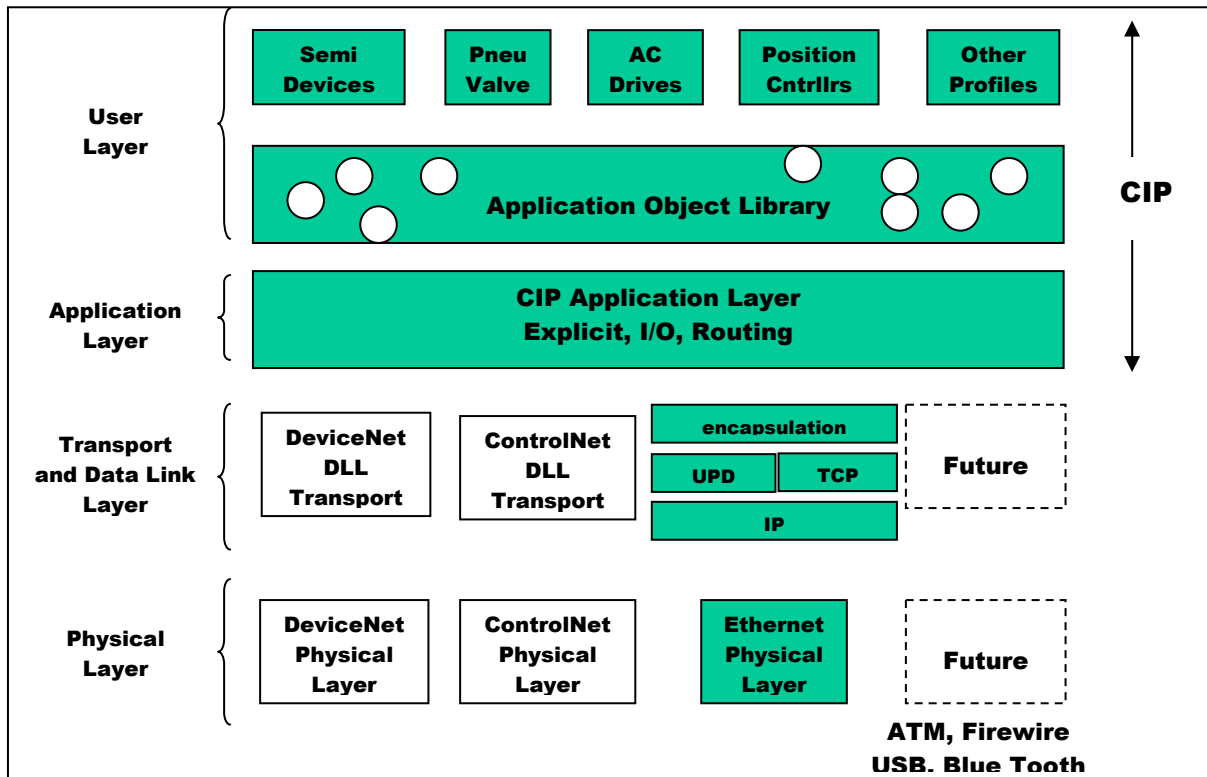


Figure 2 CIP, EtherNet/IP and Ethernet Architecture in OSI 7 Layer Model

### 2.2.4 Network Topology

Typically an Ethernet network uses an active star topology in which groups of devices are connected point-to-point to a switch. The benefit of a star topology is in its support of both 10 and 100M bit/s products. Mixing 10 and 100M bit/s is possible, and most Ethernet switches will negotiate the speed automatically. The star topology offers connections that are simple to wire, easy to debug and easy to maintain.

Ethernet is designed to handle large amounts of messaging data, 1500 bytes maximum per packet. In addition to handling large amounts of data, the Ethernet speed (10/100M bit/s) makes that data transmission even more appealing. Because of the wide acceptance of Ethernet technology throughout the years, the cost per node for Ethernet switches and other Ethernet physical media is rapidly decreasing. With these characteristics, Ethernet is becoming a viable choice for many control applications.

The Ethernet cabling components provide flexibility in two areas: overall cost and manufacturer. Due to the large number of third-party vendors, there is a wide selection of media components and cost considerations. When building a network, people may use many of the following components: cabling, transceivers, hubs, repeaters, routers, and switches.

Standard twisted-pair and fiber-optic cables are fully functional with Ethernet. Depending on the environment, people should consider products that have been proven for industrial applications. Depending upon the network configuration, an Ethernet hub or switch is appropriate. A **hub** is an inexpensive connectivity method that provides an easy method of connecting devices on information networks (shared Ethernet). A **switch** reduces collisions and is recommended for real-time control installations (switched Ethernet). **Routers** are used to isolate control data traffic from other types of office data traffic, to isolate information traffic on the plant floor from control traffic on the plant floor, and for security purposes, i.e., firewalls. **Repeaters** extend the overall network cable length. They can also connect networks with different media types.

### 2.2.5 EtherNet/IP Transmission Types

The EtherNet/IP communication protocol above TCP and UDP is called "Common Industrial Protocol" (CIP), which was introduced in 1999 for interoperability. CIP provides **implicit** messaging to be used for real-time I/O, or control messages. **Explicit messaging** is used for message exchange, or informational messages. These definitions explain the different transmission types used in the table below.

- **Information.** Non-time critical data transfers — typically large packet size. Information data exchanges are short-lived explicit connections between one originator and one target device. Information data packets use the TCP/IP protocol and take advantage of the TCP data handling features.
- **I/O Data.** Time-critical data transfers — typically smaller packet size. I/O data exchanges are long-term implicit connections between one originator and any number of target devices. I/O data packets use the UDP/IP protocols and take advantage of high-speed throughput capability of UDP.

ETHERNET/IP TRANSMISSION TYPES	MESSAGE TYPE	DESCRIPTION	EXAMPLE
Information	Explicit	Non-time-critical Information Data	Read / Write data via message instruction
I/O Data	Implicit	Real-time I/O Data	Control real-time data from a remote I/O device

Table 1 EtherNet/IP Transmission Types

### 2.2.6 EtherNet/IP Device Classes

EtherNet/IP devices are divided into the following classifications. Note that each succeeding class is a superset of the previous class.

<b>Explicit Message Server</b>	Capable of answering explicit requests only.
<b>Explicit Message Client</b>	Capable of both answering and sending explicit requests.
<b>Adapter</b>	Explicit message client and server capability plus I/O connection target.
<b>Scanner</b>	Explicit message client and server, and both target and originator of I/O connections.

### 2.3 Levels of Communication

Communications functionality required in the industrial control environment overlaps that of the office environment and adds some unique demands. Within a control system, the communication requirements vary depending upon the content of messages passed between nodes as well as the application being controlled. This section shall briefly describe three categories of messages conveyed over industrial control networks in attempt to identify the unique requirements of each.

1. Control (I/O) messages
2. Diagnostic and configuration messages
3. Information and identification messages

Control messages include the reading of inputs and writing of outputs. Diagnostic messages perform functions such as reading a detected fault value and silencing alarms. Configuration messages often occur while machines are running, to change gains and so forth, but are not performed as often as control messages. Lastly, information and identification messages occur on demand at intervals much greater than the other two message types.

#### 2.3.1 Categories of Control Network Messages

This section shall describe the network message categories on a control network. Details are provided to identify the typical content of each type of message that may be used in a control application.

##### 2.3.1.1 Control Messages

Control messages involve the majority of nodes connected to a control network, where each node communicates control data at relatively short intervals of time, require relatively short response times, require relatively rapid indication of detected faults and results in relatively high levels of control determinism. These messages provide real time status about the machine or process being controlled and whose message values have direct impact on the quality of the end product being manufactured. If a typical control network were statistically monitored, the majority of message traffic would be considered control messages. The content of control messages is usually fixed; data values are often directly applied to output points and actuators, and interpretation of message contents via typical “interpreted message protocols” generally does not exist. In effect, the meaning of each data bit at a given position within the control messages are usually predefined, only change if the system is reconfigured by a user, and therefore have significantly less CPU overhead to service than typical “interpreted messages”. EtherNet/IP shall use the UDP/IP protocols to meet control message requirements since the “time to process the messages” is minimized due to the fact that the software layers that require interpretation are significantly less.

### 2.3.1.2 Information and Identification Messages

Information and identification messages reside at the other Quality of Service extreme. At any point in time, these messages may involve a few nodes connected to a subnet, communicating over relatively long intervals of time, do not require short response times, do not require a rapid indication of detected faults and do not require high levels of determinism. Application programs can generally be written so that delayed delivery of these types of messages shall not negatively impact control message determinism.

For example, a recipe that is conveyed to a batch metering system need only be delivered prior to its usage. Once the recipe is used, the actual amount of ingredients used in the current batch must be saved to an inventory control system. As long as the actual amounts are conveyed before the next batch is started the requirements are met. The intervals between these types of messages are generally measured in minutes. In fact, loss of this information during delivery, although highly undesirable, does not directly affect the quality of the product just manufactured. Additionally, the interval between a retry does not matter, as long as the information is not lost.

Due to the non-deterministic and intermittent nature of these messages, as well as the widely varying content of these messages, existing TCP/IP protocols easily convey this category of messages.

### 2.3.1.3 Diagnostic & Configuration Messages

The diagnostic and configuration messages fill the void between control messages and information messages. These messages only occur when faults are detected or configurations are being altered during machine operation. When they occur, they generally don't involve all the nodes on the subnet, the control actions have generally already occurred, and these messages are attempting to identify why a fault occurred to ideally prevent it in the future. The determinism required here, assuming fault actions are performed during the control messages, are of higher priority than information messages and lower priority than control messages. Here again, existing Ethernet TCP/IP protocols and technology may be sufficient for the majority of these messages.

### 2.3.2 Quality of Service

Each of the message categories that have been identified also have characteristics pertaining to the Quality of Service (QoS) required of the messages. QoS characteristics can be divided into the following:

- Determinism
- Percentage of nodes communicating every second
- Response time
- Fault detection to fault action time

Later discussions will illustrate the *relative importance* of the Quality of Service criteria based upon the type of information being conveyed.

All networks and communicating devices have a finite amount of bandwidth, where the amount of messages presented will eventually saturate the media or the CPU within the devices. When system tradeoffs are required, the relative importance of the various Quality of Service attributes shall be made using the following criteria.

#### 2.3.2.1 Determinism

In control networks, determinism is *the ability to guarantee delivery of messages to the application within a specified period of time or else a fault action shall occur*. The application being controlled will ideally determine the fault duration, although default values are generally used. Given a limited amount of network and CPU bandwidth, delivery of control messages (I/O) takes precedence over all other message types since they affect the quality of the process being controlled. Determinism is less important for diagnostics and the least important for information messages.

#### 2.3.2.2 Percentage of nodes communicating within an interval of time

Consider a control network containing 50 nodes. In general, almost all of the nodes shall be producing and consuming control messages at a relatively short interval. Over a defined interval of time, the number of nodes producing diagnostic or configuration messages is generally less than five percent and may have bursts of up to ninety percent. The number of nodes producing information messages over the same interval will generally seldom exceed ten percent. When viewed on a network monitor, control messages will generally dominate the network traffic.

#### 2.3.2.3 Fault Response Time

The time to detect a control fault and act on it is much shorter than the acceptable duration to detect a configuration message error, which is less duration than an information message error. For example, for control messages, detecting a faulted input node and taking a control action generally occurs within hundreds of milliseconds. The absolute value of the required duration is application and machine dependent, but this value is a good rule of thumb. However, if an information upload fails, a machine or process failure does not generally occur. Bottom line, detecting the lack of the delivery of a control message and taking a control action must be done within a very short interval relative to the other message types.

### 2.3.2.4 Control Response Time

This criterion is listed last, since it is necessary, but not nearly sufficient to perform control over a network. Control response time is defined as the time it takes a sensor to detect a change in a machine or process state, to convey the new sensor value(s) to the control algorithm, to determine the desired action, and to effect a change in the related actuator(s). The sensor detection time, control algorithm time and actuation time are network-independent values. The network contributes the following delays:

1. The time it takes from process detection to send the first bit of the input message on the wire (media access).
2. The time it takes to convey the input message on the wire (wire time).
3. The time it takes to deliver the new sensor value(s) to the control algorithm (stack execution).
4. The time it takes to send the first bit of the output message on the wire (media access).
5. The time it takes to convey the output message on the wire (wire time).
6. The time it takes to deliver the output value(s) to the actuator (stack execution).

Failure to respond within the required interval may either damage the machine and/or process or result in less than desirable product quality. Generally, diagnostic and configuration messages are less critical, and information messages are the lowest priority.

Although the control loop can be broken into infinitely finer granularity, this document shall focus on the prior *general message categories* and *Quality of Service* message behaviors.

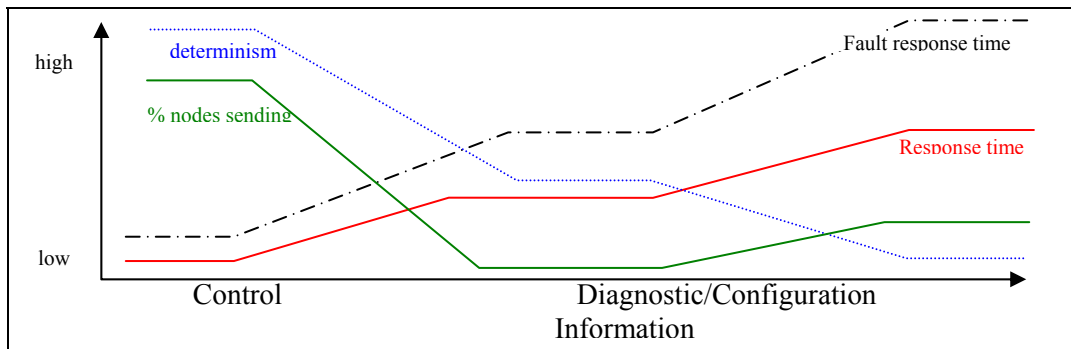
### 2.3.3 Quality of Service Relationships

When building a control network, it is important to understand the relative importance that the various quality of service measurements have on the various message categories. Summaries of the prior discussions are provided both textually in table form and graphically below.

QUALITY OF SERVICE	MESSAGE CATEGORIES		
	Control Messages	Diagnostic & Configuration Messages	Information Messages
<b>Determinism</b>	Highly important	Medium importance	Low importance
<b>Percent of Nodes Sending Messages every second</b>	Majority (90%) nodes send these messages every second	Few (<5%) nodes send these messages every second	Few, (<10%) nodes send these messages every second
<b>Fault Response Time</b>	Short interval (100's msec) from faulted control message to fault action	Medium interval (second) from faulted diagnostic message or retry	Long interval (seconds) from faulted information message to retry
<b>Response Time</b>	Short interval (10's msec) from message receipt to action /response.	Medium interval from receipt of message to response message (100's msec)	Long interval from receipt of message to response message (seconds)

**Table 2 Message Categories Versus Quality of Service**

The meaning of the vertical axis changes with each category of message and was described in the prior sections. The various plots are general relationships, the values of which are application specific, and are solely intended to provide a relative measurement in a control network.



**Figure 3 Message Categories Versus Quality of Service**

It should be apparent that control networks historically were optimized to support the far-left side of the messages shown in Figure 3. Information networks historically were optimized to support the far right side. The middle portion of the figure was historically achieved with “point to point” protocols. Today the improved network and CPU implementation technologies are allowing the convergence of all the features onto the same network. DeviceNet, ControlNet, and EtherNet/IP are individual examples of this convergence. In effect, the requirements have changed little over the past twenty years. The only change has been the availability of affordable technologies to meet these requirements without custom implementations.

As history shows, networks can be easily constructed by limiting the requirements to those that are easily achieved given a specified technology. EtherNet/IP has specified an available set of open technologies and attempts to meet all existing requirements. As is always the case, attempts to achieve the far-left side of the requirements shall have consequences on system configurations.

### 2.3.3.1 Determinism, Control & Fault Response Time Requirements

Determinism does NOT mean delivering control data at a repetitive interval. Determinism refers to the ability to deliver data within a known interval, as well the ability to detect when it is NOT delivered within a known interval. Subnet determinism is the ability to maintain a minimum update time from a sensor change to the associated actuator action(s) sufficient to maintain acceptable control of the machine or process. Fault response time requires that if a node faults, or the control data is NOT delivered within the required control loop interval, a fault action occurs prior to the occurrence of a machine or process failure. Combining these two requirements into a constant update interval is a brute force, easily understood, and easily implemented solution to meet three requirements. This is the approach taken for various competitive Ethernet networks, and is an allowed configuration using EtherNet/IP.

An ideal change of state control system would only convey data when it has changed or prior to a fault action occurring at a consuming node. When the implementation technology has a limited “wire bandwidth”, health indications may be provided by sending null (zero data byte) messages if control data has not changed, or sending control data messages immediately upon detection of a relevant change. Receipt of a null control data message may not result in execution of the associated control algorithm and production of the resulting output message. Since Ethernet's minimum packet sizes are so large, the only reason a null message would be sent would be to eliminate the CPU control algorithm execution time and associated control output message production when control data has not changed. Existing network standards, like the TCP/IP suite do not include this functionality within their scope of requirements. The common methods of achieving these requirements are among some of the items specified within EtherNet/IP and CIP.

From an implementation perspective, within many control devices, the processing of a control message shall preempt the processing of any of the other message categories. The application layer for control messages is often analogous to directly reading and writing memory locations, where output actions occur immediately upon writing the received value to memory. Bottom line, interpretation of message protocols is reduced to its lowest common denominator to minimize delays. Most commercially available TCP/IP stacks do not provide expedient action of selected UDP/IP messages versus TCP/IP messages. This is a key differentiator of commercially available TCP/IP stacks.

The existing TCP/IP stacks rely entirely on the Ethernet access mechanisms to detect collisions, back off, and retransmit messages. This often becomes a discussion topic relative to determinism. As discussed earlier, this only becomes an issue when fault actions occur due to lack of control message delivery after a defined interval of time. In many applications, like general process control, the fault actions' duration is so large that this will rarely, if ever, become an issue.

A low rate of Ethernet packet collisions will allow a higher level of determinism. Some Ethernet protocols rely on low traffic density to minimize collisions, but they sacrifice determinism in their worst cases. A simple collision avoidance technique is used in many traditional master/slave “scanners”. Programmable controllers, present in many manufacturing plants today, employ a master/slave scanner to write and read I/O data. A single client controller sends output data to a node. The node replies with its input data. Upon receipt of the input response, the client controller sends output data to the next node in its scan list. This continues until all I/O nodes have been updated. The “polling” process repeats continuously while the controller is running. (Other collision avoidance techniques exist.) In a master/slave system, polled I/O nodes only speak when they are spoken to, only 1 master can speak on the network at a given time. The result is that collisions NEVER occur! Using this simple system configuration, subnet utilization is solely limited by the time it takes the I/O node to service the output message produced by the client controller and start producing its input response message.

This type of system is totally deterministic, especially if the maximum delay that each I/O node may take to start its response can be determined. The “scan interval” is merely the sum of the update intervals of all the I/O nodes and generally includes some additional time for information and diagnostic messages after the I/O scan. Bottom line, techniques are available to solve all of the message delivery issues previously discussed as long as a customer limits the types of nodes connected to a subnet. The extremely simple master/slave solution does NOT allow more than one client to transmit on the control subnet, nor does it allow office type communications like surfing the web! A robust solution may provide methods of: identifying a new node, identifying when a node may participate on the physical medium, and enabling it's a node's communications assuming it meets the criteria established for a given machine application. EtherNet/IP may specify various system behaviors and associated mechanisms as time goes on.

### 2.3.3.2 Determinism & Fault Response Time Requirement

When control messages cease, immediate local fault actions shall occur. This has nothing to do with the delivery of messages, but the timeliness under which *non-delivery of a control message is detected and a fault action taken*. Module health is actually a separate requirement from control data delivery, but both are very important in a control system. The fault intervals are generally two to three times the expected packet rate and the fault action almost always involves the objects consuming the control data. Since the production of a separate health message would only consume valuable subnet and CPU bandwidth, traditional implementations generally combine these requirements by monitoring the control data connection. Control message consumption intervals provided by TCP/IP stacks are much greater than intervals provided by UDP/IP stacks. CIP utilizes the efficiency of the UDP/IP protocol for control communication. CIP technologies define *common interfaces* to monitor module health and control data delivery.

### 2.3.3.3 Control versus Information Messages

As indicated, within a control system, the majority of nodes perform control messages on a relatively regular basis. Information messages occur even less often and with only a few nodes at any point in time. However, configuration messages generally only occur during initial system commissioning or during upgrades, and diagnostic messages only occur with faulted nodes and only when faults are detected. Stated differently, if nodes on a control subnet were allowed to speak at any time and convey very large amounts of data whenever they wanted, control messages may be preempted or even destroyed by collisions and fault actions shall occur. A users system configuration and associated application software will impact this behavior and is beyond the scope of this document.

### 2.3.4 How do we decide if Ethernet is the right network?

Where can Ethernet be applied today when the application contains the message categories described above? Referring to Table 2, it appears that existing Ethernet protocols and hardware may be suitable within applications requiring all but control messages. The major contributing factors of interest in assessing any Ethernet based solution, including EtherNet/IP are:

- The number of nodes per segment.
- The size of messages.
- The frequency of messages.
- Message traffic patterns.
- The events that trigger production of messages.
- The tolerance of the application to delayed message delivery due to a congested network.

#### 2.3.4.1 Message Throughput

For message throughput, we have to consider theoretical scenarios as well as real-life applications. The theoretical side is governed by what the physical media can carry under full load conditions. Considering the minimum message size of 64 bytes, the actual Ethernet frame has a length of  $(64 + 8) * 8 = 576$  bit times including the sync preamble of 8 bytes. This translates into 57.6  $\mu$ s for a 10 Mbps network and 5.76  $\mu$ s for a 100 Mbps network. Adding the minimum interframe gap of 9.6  $\mu$ s (for 10 Mbps) or 0.96  $\mu$ s (for 100 Mbps) yields a time of 67.2  $\mu$ s (10 Mbps) and 6.72  $\mu$ s (100 Mbps) that is required to transmit 64 byte frames in a back-to-back fashion. Adding a few  $\mu$ s for wire transmission delay brings us in line with the approximate 70  $\mu$ s per message found in an investigation on this subject<sup>1</sup>. This translates into a message rate of approximately 14,200 messages per second. At the time this investigation was done (1988), the conclusion “No known existing networks operate under these conditions and no known existing nodes are capable of accepting and/or rejecting one Ethernet message every 70 microseconds!” was correct, but today’s world is different. Translating these findings to a 100 Mbps network introduces a factor of ten, i.e. we would get a theoretical maximum rate of 142,000 messages per second and thus an Ethernet frame approximately every 7  $\mu$ s! Another detail to consider is network congestion due to collisions. Within a hub-based Ethernet network a “rule of thumb” states that an Ethernet segment is congested and approaching excessive load when the load exceeds 20% of the network utilization. Network utilization refers to the percentage of time in which the network is busy carrying data<sup>2</sup>. Beyond this load, the number of collisions increases and control message determinism degrades. Thus, in a hub-based network running at 10 Mbps, the *maximum theoretical* message rate is reduced to 2,800 frames per second, which is still quite good for the amount of data delivered in a UDP/IP packet. A 100 Mbps system would allow a *maximum theoretical* message rate of 28,000 frames per second under the same conditions.

---

<sup>1</sup> Source: Measured Capacity of an Ethernet: Myths and Reality, David R. Boggs, Jeffrey C. Mogul and Christopher A. Kent, September 1988, <http://www.hpl.hp.com/techreports/Compaq-DEC/WRL-88-4.pdf>

<sup>2</sup> Source: Technical Report – Issues in LAN Switching and Migration from a shared LAN Environment, Rich Seifert, November 1995, <http://www.ethermanage.com/ethernet/pdf/techrept14.pdf>

However, as already mention, in many cases, the situation has changed since the two quoted investigation were done. Three aspects have to be considered:

1. The processing power that can be available in Ethernet hosts has increased dramatically since then. There are products available by now that can operate at message rates of 20,000 messages per second and beyond. However, most devices have less processing power with some of them being able to support only a few hundred Ethernet frames per second. If message throughput is to be increased, increased processing power should be considered or other measures are needed to improve the situation.
2. Many applications now use switches instead of hubs. This change improves the situation dramatically since this means a node will only see those messages directed at its MAC address, not all the messages in the segment. Once this is done, a rate of a few hundred messages per second may already be good enough for many applications. However, once I/O multicast traffic is used, the situation changes once again: Simple switches translate multicast frames into broadcasts and all frames will again appear on all ports of the switch. It is therefore mandatory to use more intelligent switches (with IGMP snooping) to avoid overload conditions when multicast traffic is present.
3. While 10 Mbps systems are still very common today, the general trend goes toward 100 Mbps systems. When looking at the theoretical considerations above, this seems to make things worse, but in reality it improves the situation by reducing the influence of the transmission system. A 10 Mbps hub-based system that is regarded as congested with a 20% utilization would only have a 2% utilization if run at 100 Mbps and would not be regarded as congested any more. Therefore, in a 100 Mbps system using switches, it is the typically the throughput of the scanner devices (this node sees all the I/O traffic) that limits system performance.

### 2.3.4.2 Ethernet for Control Messages

*How suitable is Ethernet for control messages?* Answering this question is when control engineers get nervous. If the guaranteed delivery of control messages is very frequent and extremely critical and the loss or delay of a single control message could result in damaged product, equipment or injured personnel, commercial Ethernet solutions should NOT be used. The open CIP solution, once complete, shall provide the technology and conformance testing necessary to achieve traditional control system requirements in various system configurations.

### 2.3.4.3 What CIP Brings to the Table

In addition to the physical and data link layer technologies, EtherNet/IP provides a proven industrial application layer within CIP. CIP is based on an open, object oriented Producer/Consumer Communication Model, that is optimized for automation applications. Additionally, it contains a broad range of standardized application objects, enabling improved multi-vendor system interoperability. An open network protocol, without a common set of application interfaces and behaviors, can never approach the ease of integration of this type of approach.

As should be apparent from the prior network loading discussion, when it comes to a network's speed, baud rate is only one of the factors to consider. The network communication model used to exchange control data and information between devices actually has a significant impact on network functionality, which is especially important on an Ethernet network. All networks fall into one of two categories: Source/Destination or Producer/Consumer. The difference between the two can be best conveyed by a human analogy. Assume one needs to tell a room full of people the time of day. With source/destination, one person reads a clock and then proceeds to individually tell each person in the room the time. In producer/consumer mode, the same person announces the time to everyone at once and everyone can consume it at the same time. In a Producer/Consumer environment, "identifiers" embedded into each message are used by the devices to determine which messages they should "consume." Although the network model does not impact the rate at which data is transmitted, it does affect how efficiently the available bandwidth is used. It uses less bandwidth because a producer/consumer network transmits a piece of information only once. Less bandwidth equates to greater efficiency and overall speed.

EtherNet/IP, ControlNet, DeviceNet, and Foundation Fieldbus are examples of networks based on the Producer/Consumer technology.

### 3 Recommendations

This section provides recommendations to vendors interested in developing EtherNet/IP products for the industrial control markets. The recommendations are intended to help vendors develop EtherNet/IP products that best meet the needs of end users. Many of these recommendations are based on experiences from early adopters and EtherNet/IP technology inventors.

Note that this paper will not identify specific brands or vendors in its recommendations. The purpose of the document is to provide a list of recommended features and functions that a developer should be aware of when selecting products to be used with their EtherNet/IP implementation.

#### 3.1 Recommendations for Physical Layer Components

EtherNet/IP and other Ethernet based industrial protocols require physical layer features that enhance the determinism and reliability of communications. The heart of an Ethernet physical layer is the controller chip and its related transceiver and magnetic components.

##### 3.1.1 Controller Chips

Some controllers contain the transceiver on-chip while others require external transceivers. The sponsor of this paper, ODVA, have identified the following core list of features that would make controllers very attractive to developers of EtherNet/IP products:

- Resistant to industrial environments – temperature, humidity, vibration, etc.
- Support full duplex communication.
- Support 10 Mbps and 100 Mbps operation.
- Support IEEE 802.1p packet prioritization with different transmit and receive queues for each priority level. During a receive operation, the controller hardware would inspect the packet priority field and transfer the received packet to the appropriate queue. Driver software would check the queues in descending order of priority, ensuring that important messages are not queued up behind less important messages. During a transmit operation, the controller hardware would take the next waiting packet in the highest priority non-empty queue and transmit it.
- Ability to change the transmission characteristics on a packet by packet basis. Driver software would set a control field at the head of each packet before placing the packet into its transmit queue. The control field would contain information about how the controller should handle that packet during transmission. The control information would be used to instruct the controller to limit the number of retries to attempt before aborting transmission. This would be used for high priority packets that cannot afford to be delayed by numerous retries. The industrial control application may need to know immediately if the packet was not transmitted in a timely fashion. The transmission control field could also be used to enable or disable the automatic appending of the CRC to the end of the packet.
- Strong support for multicast address filtering. Minimize CPU cycles needed to accept and reject multicast packets.
- Support for receiving all packets (promiscuous mode). Useful for diagnostic and monitoring applications. Normal applications would disable this feature.

- Ability to perform deeper packet filtering in hardware – allow packets to be accepted/reject based on network protocol, UDP destination port, TCP destination port, etc.

Some of the above features, such as 10/100 Mbps and Full/Half Duplex have already been implemented by some manufacturers. The other features will probably evolve over time as industrial uses for Ethernet continue to grow and a substantial market develops for these features.

### 3.1.2 Magnetics

Magnetics play an important role in providing isolation for the device from the cable plant. Any one of the components in the network will set the entire network performance. In addition to isolation, the transformer must provide as high common mode rejection (CMR) as possible. Most transformers only supply 30 dB to 40 dB at 0-30 MHz. Developers are strongly encouraged to use transformers with a minimum of 59 dB at 30 MHz. Pulse Engineering makes an example of this type of transformer (H1112 and H1126.) One known drawback is the common mode chokes may cause problems for transceivers that perform auto-crossover wiring detection. Some transceivers will produce distorted signals when the common mode choke is on the transceiver side of the transformer.

### 3.1.3 Protection Circuitry

Protection circuitry provides *Electrostatic Discharge* (ESD), Surge and *Electrical Fast Transient* (EFT) protection. The circuit must be designed in such a way to not cause an imbalance in the transmit and receive pairs. Once again, any imbalance in the network will directly impact the CMR noise performance of that network. The purpose of the protection circuitry is to clamp the differential voltages to within safe levels for the transceiver. In addition, there are some devices that can clamp the common mode noise within safe limits. It is recommended that these devices be placed in different places depending on the desired level of protection.

- If there is a need to clamp common mode and differential mode noises, the clamp circuitry should be placed between the transformer and the transceiver. Reference Figure 4.
- If there is a need to only clamp the differential voltages on the pairs, then the clamp circuitry can be placed between the connector and the transformer. Reference Figure 5.

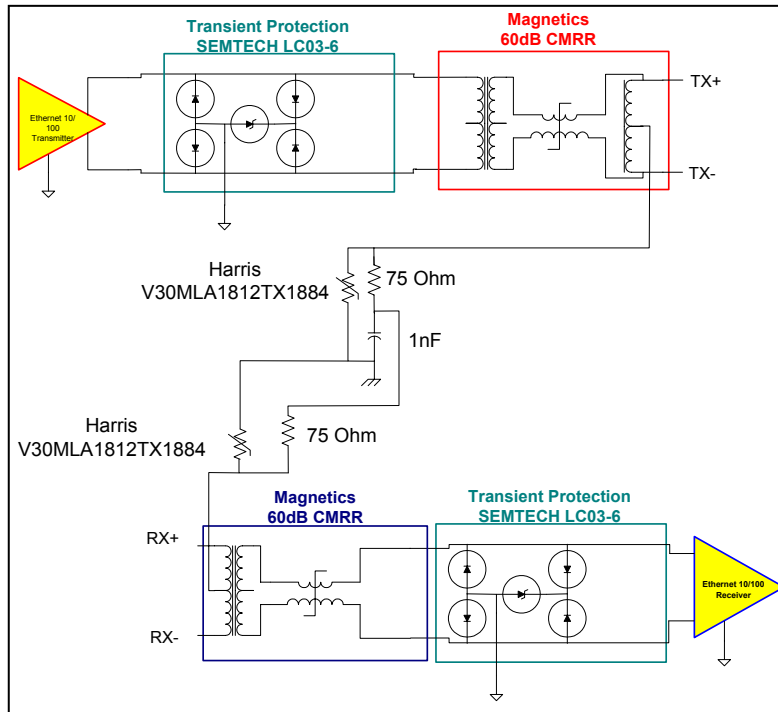


Figure 4 Example Differential and Common Mode Clamp Circuit

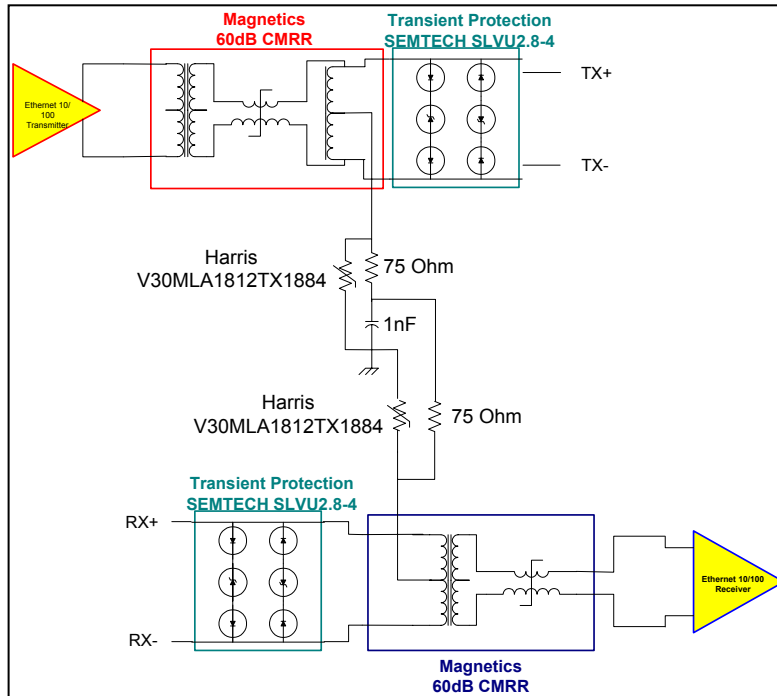


Figure 5 Example Differential Mode Clamp Circuit

### 3.1.4 Connectors (Jacks and Plugs)

RJ style connectors are found on virtually all shielded twisted pair (STP) and unshielded twisted pair (UTP) Medium Attachment Units. Generally these connectors perform well in office environments. However, industrial environments present some challenges. The following performance parameters must be investigated before designing in an RJ-45 connector.

- Contacts must be designed to withstand the high vibration and shock found on the factory floor. This vibration is typically 2 to 10 Gs.
- Temperature ranges from –10 to +85 degrees C. The plastics must be selected to withstand the high temperatures.
- Gold contacts are required to prevent corrosion in caustic environments.
- The gold and under-plating must be robust enough to survive vibration and repeated insertion and removals of the connectors.
- Sealing for the connector. There may be several connectors introduced in the next few months that attempt to solve the sealing issues. Selecting connectors other than the RJ-45 (sealed or unsealed) puts the responsibility on the designer to make sure the connector meets the stringent fast Ethernet specifications from an electrical performance perspective.

Approved industrial connectors include a sealed RJ45 IEC61076-106-3 Variant 01 and a M12-4 IEC 61076-101. Approved connector descriptions are in the EtherNet/IP specification, Volume 2, Chapter 8, Sections 8-5.1.4, 8-6.2.3, and 8-6.3.1.

### 3.1.5 Cables

Cable performance is the most important contributor to the overall performance the industrial Ethernet network. Placement of the cable in the system is equally as important. System and/or cable designers are strongly encouraged to consult the EtherNet/IP specification for selection and design of cables to be used in an industrial control application. The use of standard off the shelf cables may cause system malfunction or degradation. For proper installation and grounding methods are described in the EtherNet/IP Installation Guide.

Use either CAT6 or shielded twisted pair (STP) cables in high noise environments.

### 3.1.6 Device Design (Physical Layer)

The industrial environment requires changes in the design and selection of components used in an industrial communications module. The following are critical details to consider when designing an industrial communications module for Industrial EtherNet/IP:

#### 3.1.6.1 Temperature/Humidity

Select components that will survive the temperature and humidity of the environment. For example, the magnetics must meet the attenuation specification over the extended temperature range.

#### 3.1.6.2 Immunity to Radiated and Conducted Noise

High CMR is the key to maximizing the systems immunity to noise. Hardware designs must assure that Network CMR is not degraded by poor layout or improper selection of components. Trace lengths between the connector, magnetics and transceiver should be equal in length.

### 3.1.6.3 Radiated Emissions

Layouts should be optimized for isolation. All digital traces should be kept away from the physical layer and never should be placed over earth grounds. Likewise, traces from the network side of the transformer should never be routed into the digital areas of the design. Earth ground should never be placed under the digital or analog area of a design.

### 3.1.6.4 Grounding

Grounding of the shield and protective devices is critical in surviving the high noise/voltage environmental tests. Further grounding of the shield in STP cables is important from a system perspective. Ground loops in the shield will cause noise to be coupled into the communications pairs. The shield termination recommendations are covered later in this section. Shield termination practices are an integral part of the EtherNet/IP specification.

### 3.1.6.5 Isolation

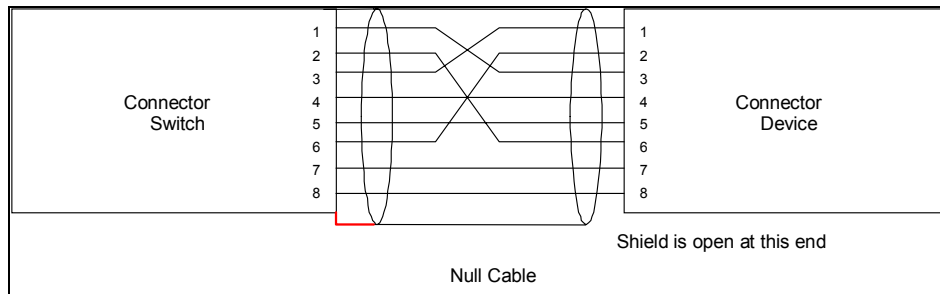
From a safety and reliability perspective, isolation from the network and device power supply is very important. Virtually all Ethernet transformers are only rated at 1500V. Most industrial testing requires a minimum of 2KV. There can be enough energy that crosses from the network side into the transceiver side of the transformer to cause failure and present a safety hazard. The routing of conductors and bypass capacitor placement in the design can impact the isolation. Placing a bypass capacitor between Earth ground and digital ground can couple noise to and from the logic section of your design and is not advisable.

### 3.1.6.6 Shield Termination

There are two methods of terminating the shield. Both result in a single DC point of ground.

#### *Option 1*

Option 1 should only be used when connecting two commercially off the shelf devices (Switch and NIC) together in a network. Any connectivity device such as Hub, Switch or Router that uses shielded RJ-45 jacks, will have a direct connection to earth either through the power plug or separate ground lug. This ground connection provides the single point of ground and should not be defeated. The device end should be isolated by not terminating the shield in the RJ-45 plug. The figure below is an example of how the single point termination can be implemented in the patch cord.



**Figure 6 Cross-Over Cable Single Ground Connection**

### Option 2

Option 2 should be used in the design of all EtherNet/IP devices. This provides a single point DC ground at the connectivity device and an AC ground at the EtherNet/IP device. The shield is terminated at the EtherNet/IP device to earth ground through a parallel RC. The values of the termination circuit are .01 $\mu$ F/500V (min) capacitor and 1 meg Ohm  $\frac{1}{4}$  watt resistor. To protect the capacitor, a MOV should be placed in parallel with the RC. The part number of the MOV is provided in the figures below.

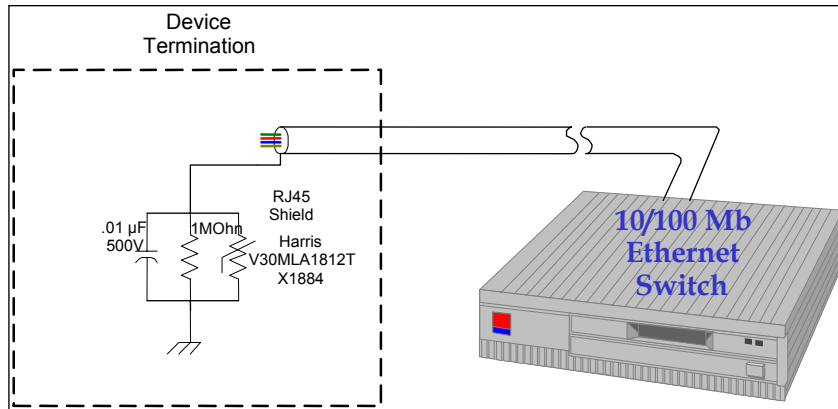


Figure 7 Example System Grounding Circuit

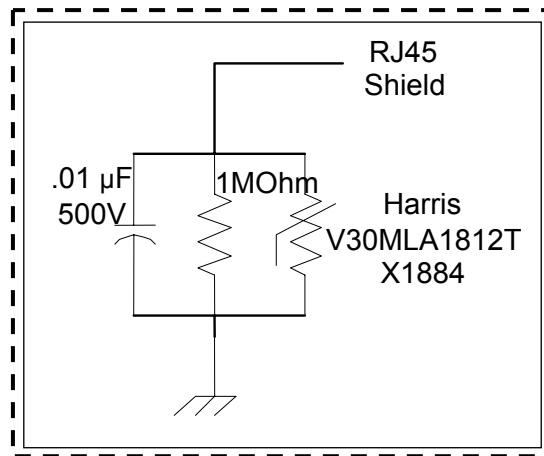


Figure 8 Example R-C Circuit with Protection

### 3.1.6.7 Layout

Layout is critical for noise hardening. High frequency devices should be isolated from the physical layer area. Clock lines should be terminated and routed away from the physical layer area. Likewise, never route earth ground or physical layer traces into or over digital areas. The digital ground plane and power planes should not extend into the physical layer area. The figure below provides an example layout. Circuit traces connecting the RJ-45 and transformers should be treated as transmission lines. This requires that the characteristic impedance be matched and constant.

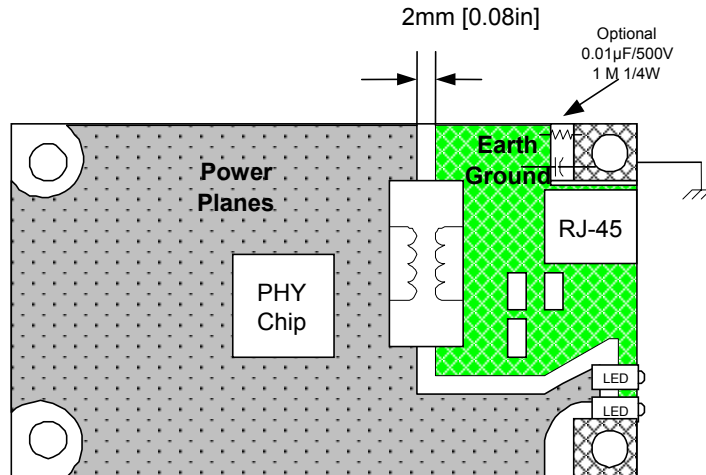


Figure 9 Typical Layout

### **3.2 Recommendations for TCP/IP Stack**

The TCP/IP stack shall be compliant with all relevant IETF (Internet Engineering Task Force) standards. Some stack features are required while others are recommended. EtherNet/IP Specification, Volume 2, Chapter 9, Section 4 (“Requirements for TCP/IP support”) contains minimum requirements for a TCP/IP stack in all EtherNet/IP devices. Stack requirements help ensure compatibility with TCP/IP stacks in other products. In addition to the EtherNet/IP specification, the stack recommendations outlined below, promote message processing speed and portability to other operating environments.

#### **3.2.1 TCP/IP Stack Recommendations for All EtherNet/IP Devices**

- Stack should minimize the copying of data when passing a packet up or down through the layers of the stack. This feature is often called “zero copy”.
- Stack should not be “tweaked” or modified so as to preclude any IP and/or any TCP header options
- Stack should include the option of enabling TCP & UDP Echo Servers (Port 7). The stack should allow port 7 to be disabled.
- Driver and stack should support prioritization of messages. This will allow EtherNet/IP messages to be prioritized above other TCP traffic. Note that the prioritization described below is rare in current stacks. If it is desired it will, most likely, have to be added and managed by the developer.
  - Driver for controller chip should support multiple priority message queues for both transmission and receptions.
  - Stack should support priority when processing received messages up through the layers of the stack to the application.
  - Stack should support priority when processing messages to be transmitted down through layers of the stack to the driver and the controller chip.
- Application Program Interface (API) should support standard socket oriented interface.
- Stack should provide options that allow it to operate in a mode that is optimized for EtherNet/IP. When operating in this mode, the stack will minimize CPU cycles required to determine if a message is EtherNet/IP. For example, the stack may filter by protocol ID or UDP/TCP port number before performing checksums.
- It is recommended that the stack be provided with source code to allow for EtherNet/IP and application specific performance and timing modifications. If the developer does not have source code for the stack it may be difficult or impossible to alter the stack to be suitable for Industrial Ethernet applications. For example, the Address Conflict Detection protocol requires the TCP/IP stack to send ARP probes with fairly fast timing requirements.

#### **3.2.2 TCP/IP Stack Recommendations for EtherNet/IP Scanners**

- Stack should support IP Multicast reception using IGMP v2. Note that IGMP is only required for consuming connection data, not producing.

### **3.3 Recommendations for Operating System**

The operating system that will be selected for an EtherNet/IP project should be considered in concert with the TCP/IP stack and processor selection. Many operating systems now include TCP/IP stacks and both the stack and the OS work best when used together as designed.

The operating system selection also depends on the application. Most low-end Adapter class devices will not require a high level OS, and may get by with a small kernel. Higher end devices which perform control and scanner functionality will require correspondingly higher level operating systems.

It is helpful if the OS provides the means to monitor run-time operation during development so that developers may analyze their application. Developers will want to be able to see CPU idle time, distribution of CPU time over the different application tasks, memory usage and interrupt performance. The monitor should be capable of being removed or disabled for the final application build.

### **3.4 Processor Recommendations**

As noted in the previous section, processor selection should be done in concert with TCP/IP stack and operating system selection.

Many processors are available with on-chip Ethernet controllers. If selecting a processor with an on-chip Ethernet controller, the controller features outlined in the Controller Chips section above should be considered when making the selection.

Many TCP/IP stacks have suggested processor and resource sizing. Follow these recommendations and add for the level of application.

### **3.5 EtherNet/IP Protocol Stacks**

There are some commercially available EtherNet/IP protocol stacks. A list of commercially available stacks is available on the ODVA web site ([www.odva.org](http://www.odva.org)).

ODVA ([www.odva.org](http://www.odva.org)) provides a freeware EtherNet/IP stack that may be used as a reference for a developer implementing EtherNet/IP. The freeware provides source code for an Adapter class device with minimal functionality.

## **4 Organizational Support**

ODVA and ControlNet International will co-manage the EtherNet/IP specification including all enhancements.

ODVA is an independent organization of users, vendors, and distributors to support the worldwide growth of DeviceNet by assisting with tools, training, compliance testing and marketing activities. ODVA operates globally with offices in Europe, North America and Australia and affiliates in Japan, Korea, New Zealand and the United Kingdom. DeviceNet is an open communications network designed to connect factory floor devices, such as sensors, push buttons, motor starters and drives, to control systems.

ControlNet International is an independent organization for users and vendors of ControlNet products. ControlNet is a real-time, control-layer network providing for high-speed transport of both time-critical I/O and messaging data, including upload/download of programming and configuration data and peer-to-peer messaging on a single or redundant, intrinsically safe physical media link. Deterministic and repeatable, ControlNet's high-speed control and data capabilities significantly enhance I/O performance and peer-to-peer communications.

### **4.1 JSIGs**

Joint Special Interest Groups (JSIG) comprised of ODVA and CI members have been formed to develop and manage other aspects of EtherNet/IP technology including EtherNet/IP conformance tests, enabler products and end-user educational materials. The following JSIGs are currently active.

<b>CIP System Architecture</b>	Develops and maintains technology common to EtherNet/IP, DeviceNet, and ControlNet.
<b>EtherNet/IP Conformance</b>	Promotes uniform interpretation of specifications and verify product coexistence in evolving multi-vendor information and/or control systems by developing and deploying conformance test methods and test labs.
<b>EtherNet/IP Enablers</b>	Promote the acceptance of the EtherNet/IP technology by providing assistance (in the form of enablers) to vendors developing EtherNet/IP products. The EtherNet/IP Implementors Workshop is operated by this JSIG.
<b>EtherNet/IP Physical Layer</b>	Develops and maintain specifications for Physical Layer components for use in Industrial EtherNet/IP applications.
<b>EtherNet/IP System Architecture</b>	Develops and maintains technology specific to EtherNet/IP.

### **EtherNet/IP Marketing and User Education**

Develops and publishes information to assist the automation community in the evaluation, selection, specification and installation of Ethernet-based control systems.

### **EtherNet/IP Infrastructure Recommendations Task Group**

Develops and publishes information to assist in infrastructure design for EtherNet/IP network architecture.

## **4.2 Enabler Technologies**

The ODVA and CI have agreed to make the EtherNet/IP specification and Enabler technologies available free of charge. The Enabler Technologies consists of:

- EtherNet/IP Adapter Example Code (C source code and user manual)
- Sample target (server) application with a GUI (runs on Windows NT, 2000, and XP)
- EtherNet/IP Specification
- White Papers and Technical Documentation

The Enabler technologies are posted on the ODVA web site at ([www.odva.org](http://www.odva.org)). They are available for developers to download free of charge.

ODVA and ControlNet International will support the adoption of EtherNet/IP by providing training classes, trade show presence, technical support, developer tools, speakers bureaus, white papers, and a conformance test suite.

## **4.3 EtherNet/IP Implementors Workshop**

The EtherNet/IP Implementors Workshop was created by the Enablers JSIG. The workshop is intended to provide a forum for EtherNet/IP product manufacturers and users to share ideas and information to further promote the use of EtherNet/IP. Workshop meetings are held approximately four times a year. Workshop presentation topics range from development recommendations to new technology overviews.

### **4.3.1 EtherNet/IP Recommendations Papers**

The main goal of the Implementors Workshop is to promote interoperability between EtherNet/IP devices. To meet this goal, the Workshop has produced several papers containing recommendations for product developers and users (including this document). These papers are listed at the front of this document in the Companion Documents section.

### **4.3.2 EtherNet/IP Plug Fests**

In keeping with the goal of device interoperability, the Implementors Workshop organizes an EtherNet/IP Plug Fest approximately 1-2 times a year. The Plug Fest provides an opportunity for EtherNet/IP product vendors to test their product against a suite of interoperability tests with a number of other product and infrastructure vendors. The goal of the Plug Fest is to provide feedback to the developer in regards to any possible interoperability problems before they are discovered in the field.

### **4.3.3 North American and European Tours**

The EtherNet/IP Implementors Workshop holds sessions in the United States and Europe. See the ODVA web site ([www.odva.org](http://www.odva.org)) for meeting schedules and agendas for both tours.

## 5 Terms and Definitions

Adapter Class	An Adapter Class product emulates functions provided by traditional rack-adapter products. This type of node exchanges real-time I/O data with a Scanner Class product. It does not initiate connections on its own.
Connected Messaging	A connection is a relationship between two or more application objects on different nodes. The connection establishes a virtual circuit between end points for transfer of data. Node resources are reserved in advance of data transfer and are dedicated and always available. Connected messaging reduces data handling of messages in the node. Connected messages can be Implicit or Explicit.
Connection Originator	Source for I/O connection or message requests. Initiates an I/O connection or explicit message connection.
Explicit Messaging	Explicit Messages can be sent as a connected or unconnected message. CIP defines an Explicit Messaging protocol that states the meaning of the message. This messaging protocol is contained in the message data. Explicit Messages provide a one-time transport of a data item. Explicit Messaging provide the means by which typical request/response oriented functions are performed (e.g. module configuration). These messages are typically point-to-point.
Implicit Messaging	Implicit Messages are exchanged across I/O Connections with an associated Connection ID. The Connection ID defines the meaning of the data and establishes the regular/repeated transport rate and the transport class. No messaging protocol is contained within the message data as with Explicit Messaging. Implicit Messages can be point-to-point or multicast and are used to transmit application-specific I/O data. This term is used interchangeably with the term I/O Messaging.
I/O Client	Function that uses the I/O messaging services of another (I/O Server) device to perform a task. Initiates a request for an I/O message to the server module. The I/O Client is a Connection Originator
I/O Messaging	Used interchangeably with the term Implicit Messaging.

## General Recommendations for EtherNet/IP Developers

---

I/O Server	Function that provides I/O messaging services to another (I/O Client) device. Responds to a request from the I/O Client. I/O Server is the target of the connection request.
Master	When used in a CIP context, Master is specific to DeviceNet with regard to the pre-defined Master/Slave Connection Set. EtherNet/IP does not use Master/Slave terminology.
Message Client	Function that uses the Explicit messaging services of another (Message Server) device to perform a task. Initiates an Explicit message request to the server device.
Message Server	Function that provides Explicit messaging services to another (Message Client) device. Responds to a Explicit message request from the Message Client.
Scanner Class	A Scanner Class product exchanges real-time I/O data with Adapter Class and Scanner Class products. This type of node can respond to connection requests and can also initiate connections on its own.
Slave	When used in a CIP context, Slave is specific to DeviceNet with regard to the predefined Master/Slave Connection Set. EtherNet/IP does not use Master/Slave terminology.
Connection Target	Destination for I/O connection or message requests. Can only respond to a request, cannot initiate an I/O connection or message.
Unconnected Messaging	Provides a means for a node to send message requests without establishing a connection prior to data transfer. More overhead is contained within each message and the message is not guaranteed destination node resources. Unconnected Messaging is used for non-periodic requests (i.e. network “Who” function). Explicit messages only.