4. INTRODUCTION TO ETHERNET/IP TECHNOLOGY

4.1. Introduction

Ethernet Industrial Protocol (EtherNet/IP) is an open industrial networking standard. It has been developed by Rockwell Automation, managed by ODVA (Open DeviceNet Vendors Association) and designed for use in process control, hard-time systems, and industrial automation applications. (EtherNet/IP) was introduced in 2001 and today is the most developed, proven and complete industrial Ethernet network solution available for manufacturing automation. EtherNet/IP emerged due to the high demand of the Ethernet network for control applications. Because there is wide acceptance of Ethernet technology and it is now well-established, the cost per node for Ethernet switches and other Ethernet physical media is low.

EtherNet/IP uses the tools and technologies of traditional Ethernet such as Transport Control Protocol (TCP), the Internet Protocol (IP) or User Datagram Protocol (UDP). Ethernet/IP uses standard Ethernet TCP/IP as it is, therefore it is classified as Class 1 Real Time Ethernet, according to the IEC 61 784-2. Class 1 is the class that conforms most to the Ethernet TCP/IP standard and can therefore use standard hardware and software components. With the CIPsync extensions it is possible to get isochronous communication that satisfies class 2 applications. These extensions use 100 MBit/s networks with the help of IEEE 1588 time synchronisation.

Ethernet/IP was developed from a very widely implemented standard used for transferring data between two devices in DeviceNet and ControlNet, called the **Common Industrial Protocol** (CIP).

It is useful to take a look at EtherNet/IP in terms of the seven-layer Open System Interconnection (OSI) Reference Model as presented in Figure 4.1. As with all CIP Networks, EtherNet/IP implements CIP at the Session layer and above and adapts CIP to the specific EtherNet/IP technology at the Transport layer. TCP/IP **encapsulation** allows a node on the network to embed a message as the data portion in an Ethernet message. The encapsulation technique uses both the TCP and UDP layers of the TCP/IP layers and provides the method that allows CIP to be implemented transparently on top of Ethernet and TCP/IP.

To obtain the desired level of **service quality**, EtherNet/IP also uses **standard mechanisms** defined in layer 3 (IP) and layer 2 for Ethernet (e.g. 802.1D/Q), supported by a **proper hardware infrastructure** (e.g. multiple queue switches).

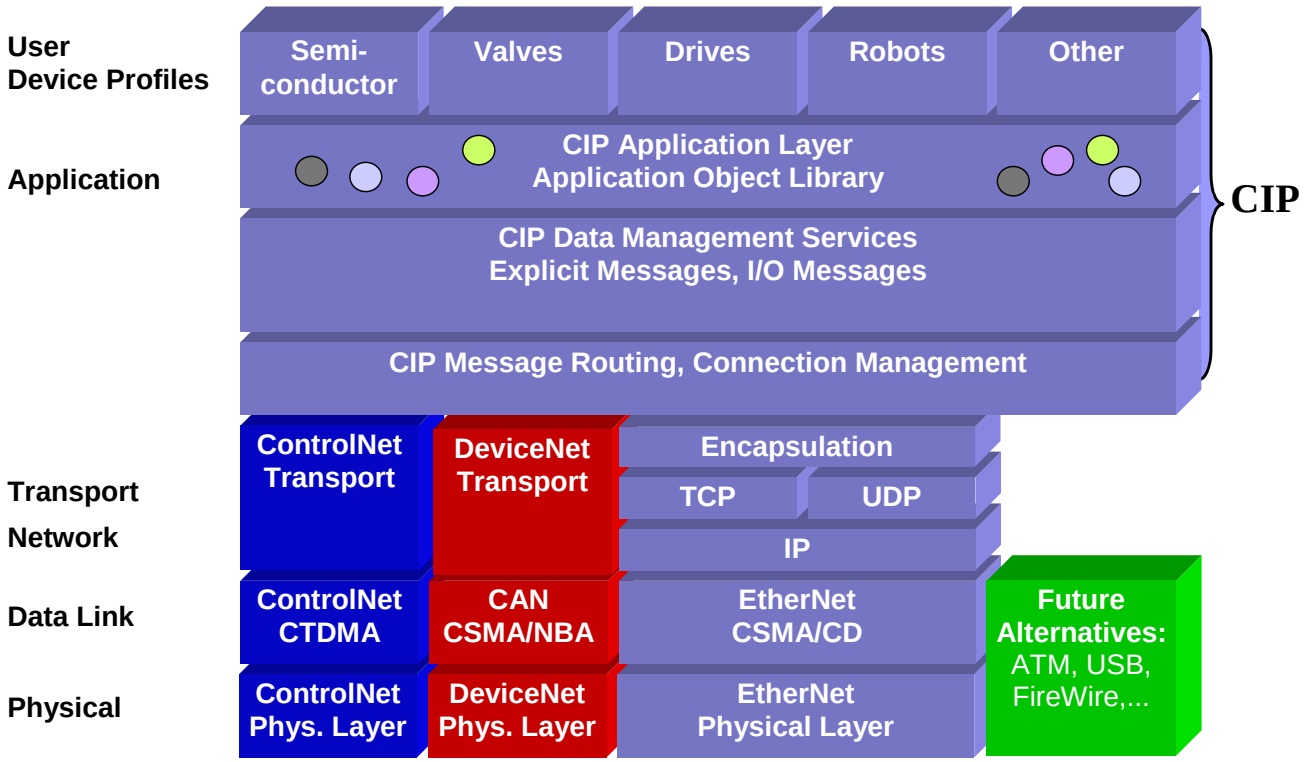

Fig. 4.1. (EtherNet/IP), Comparison of DeviceNet and ControlNet OSI [1]

The Physical Layer

EtherNet/IP uses the standard IEEE 802.3 model at the Physical and Data Link Layers. The Physical Layer is concerned primarily with the interaction of a single device with a medium. The Physical Layer is responsible for bit-level transmission between network nodes, and defines specifications for electrical signals (copper networks) or the characteristics of light signals (fiber optic networks). The Physical Layer also defines items such as: connector types, cable types, voltages, and pin-outs.

The Data Link Layer

The IEEE 802.3 specification is also used for transmitting packets of data from device to device on the EtherNet/IP Data Link Layer. The Data Link Layer is concerned more with the interactions of multiple devices (i.e., at least two) with a shared medium. (EtherNet/IP) uses a CSMA/CD Media Access Control (MAC) model that determines how networked devices share a common bus (i.e. cable), and how they detect and respond to collisions between packets.

The Network and Transport Layers

The Network and Transport Layers use TCP/IP Suite to send messages between one or more devices. At these layers, messages used by all **CIP networks are encapsulated**.

TCP/IP encapsulation allows a node on the network to embed a message as the data portion in an Ethernet message. The encapsulation technique uses both the TCP and UDP layers of the TCP/IP suite and provides the method that allows CIP to be implemented transparently on top of Ethernet and TCP/IP.

EtherNet/IP uses two forms of messaging and the appropriate resources at the nodes, as defined by CIP standard (Fig.4.3)

• **Unconnected messaging** is used in the connection establishment process and for infrequent, low-priority messages. The unconnected resources in a device are referred to as the Unconnected Message Manager, or UCMM. Unconnected messages on EtherNet/IP utilize TCP/IP resources to move messages across Ethernet, asking for connection resource each time from UCMM.

• **Connected messaging** EtherNet/IP utilizes resources within each node that are dedicated (reserved) in advance to a particular purpose, such as frequent explicit message transactions or real-time I/O data transfers. Connection resources are reserved and configured using communications services available via the CMM.

EtherNet/IP has two types of network connections: **Explicit** and **Implicit.** By using TCP/IP, EtherNet/IP is able to send Explicit messages, which are used to perform client-server (point-to-point) type transactions between nodes. For real-time messaging, EtherNet/IP uses the TCP/UDP model, which allows messages to be multicast (in the sense that its target is a number of nodes in a network, and it is directed to a group of hosts/destination addresses). With Implicit messaging connection, the data field contains no protocol information, only real-time I/O data. Since the meaning of the data is pre-defined at the time the connection is established, processing time is minimized during runtime. UDP is connectionless and makes no guarantee that data will get from one device to another, however UDP messages are smaller and can be processed more quickly than TCP/IP messages. As a result, EtherNet/IP uses UDP/IP to transport I/O messages that typically contain time-critical control data.

There are three transmission types used in an (EtherNet/IP) network (Table 4.1):

> **Information.** Non-time critical data transfers — typically large packet size. Information data exchanges are short-lived explicit connections between one origin 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 origin and any number of target devices. I/O data packets use the UDP/IP protocols and take advantage of the high-speed throughput capability of UDP.
>
> **Real-time Interlocking.** Cyclic data synchronization between one producer processor and any number of consumer processors. Interlocking data packets use the faster UDP/IP protocols and take advantage of the high-speed throughput capability of UDP.

Table 4.1. Ethernet/IP message types

ETHERNET/IP Transmission Type	Message Type	Description	Example
Information	Explicit	Non-time-critical Information Data	Read/Write data by message instruction
I/O Data	Implicit	Real-time I/O Data	Control real time data from remote I/O device
Real-time Interlocking	Implicit	Real-time Device Interlocking	Exchange real-time data between two processors

Session, Presentation and Application Layers

EtherNet/IP uses the Common Industrial Protocol (CIP), a strictly object-oriented protocol, at the upper layers. Each CIP object has attributes (data), services (commands) and behaviors (reactions to events).

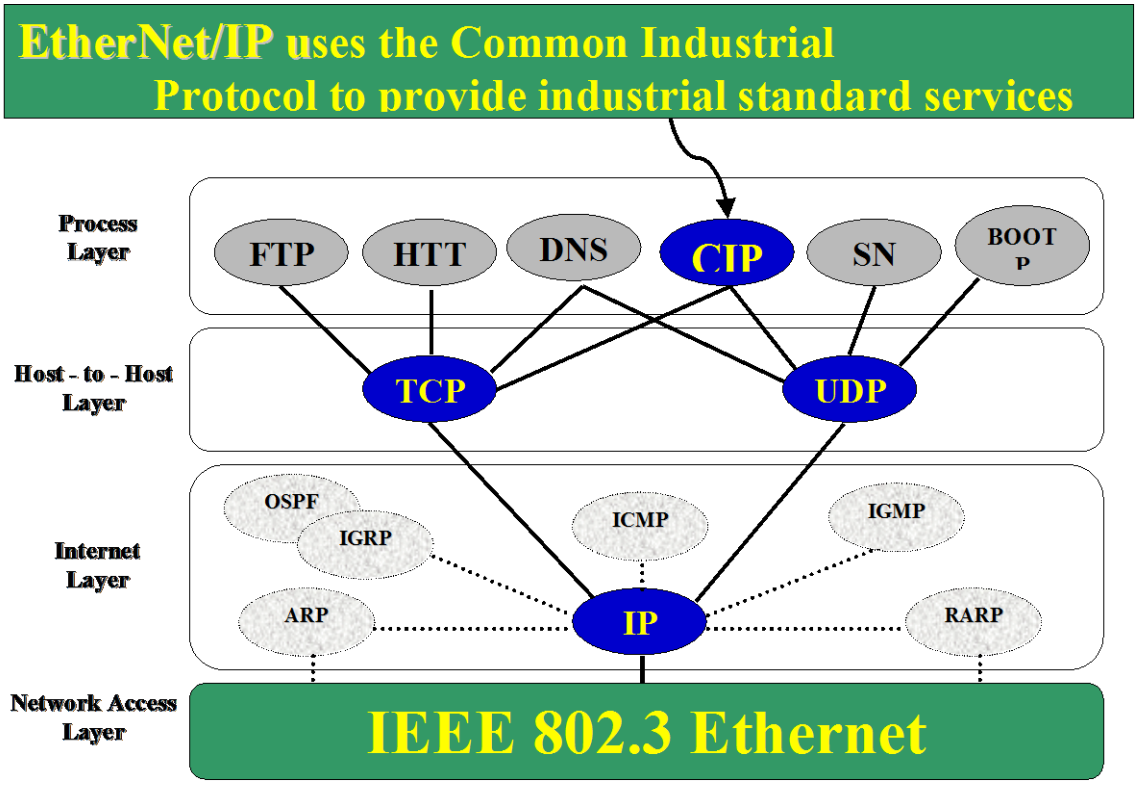

Fig. 4.2 Upper Layers with CIP protocol [1]

4.2. CIP and data exchange model

In the CIP Protocol, every network device represents itself as a series of objects. Each object is simply a grouping of the related data values in a device. There are three types

of objects defined by the CIP protocol and used by Ethernet/IP for data representation.

Required Objects

Objects required by the specification of every CIP device. For example:

- Identity object: contains identity data called **attributes** (ex. Vendor ID, Date of manufacturer, Device serial number and other identity data)
- Message Router object: this object routes explict request messages between objects in a device
- Network object: contains the physical connection data for the object such as IP address and other data describing the interface to the Ethernet port on the device

Application Objects

Application Objects allow the user to organize the data that are specific to a particular kind of device. These objects define the data encapsulated by the device. They are specific to the device type and function. For example an analog I/O device can be described in object terms by attributes such as: type, resolution, values of input and output.

These application layer objects are predefined for a large number of common device types. The same type of CIP devices must contain the same series of application objects. The series of application objects for a particular device type is known as the device profile.

Vendor Specific Objects

Objects not found in the profile for a device class are termed Vendor Specific. These objects are included by the vendor as additional features of the device. The CIP protocol provides access to these vendor extension objects in exactly the same way as either application or required objects.

Accessing regular pieces of data from the network to the device requires:

- Object Number
- Instance Number (Instances are the way of organizing the same kind of data , e.g., sharing same attributes)
- Attribute Number

A typical CIP device is shown in Fig. 4.3:

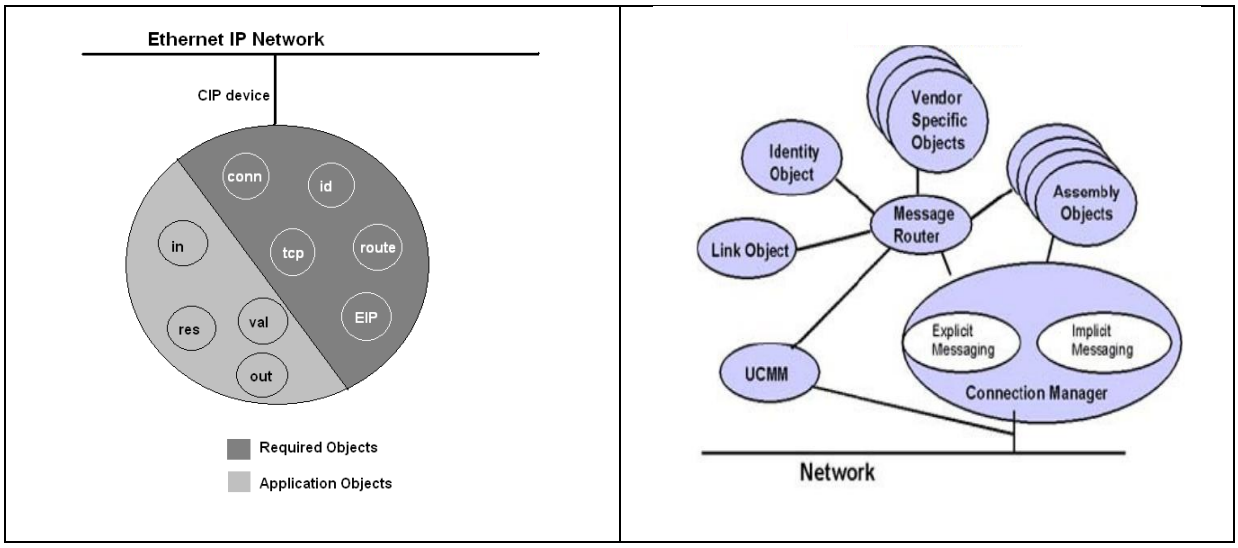

Fig. 4.3. A typical CIP device representation [1]

Ethernet/IP uses **the producer-consumer** data exchange model which describes rules for how data is exchanged between application programs running in devices.

The CIP producer/consumer networking model replaces the old source/destination (master/slave) model. In traditional I/O systems, controllers **poll** input modules to obtain their input status. In the CIP system, input modules are not polled by a controller. Instead, they produce (multicast) their data either upon a change of state or periodically. With implicit connections, messages are sent cyclically.The frequency of update depends upon the options chosen during configuration.

The input module, therefore, is a producer of input data, and the controller is a consumer of the data. The controller can also produce data for other controllers to consume. Information generated by one device can be consumed by a group of devices over the EtherNet/IP network.

When a message is introduced into the network it is identified by its connection ID not by its destination address. Multiple nodes may then consume the data to which the connection ID refers. As a result, when a node wants to receive data it only needs to ask for it once in order to consume the data each time it is produced. If subsequent nodes want to receive the same data simultaneously all they need to know is the connection ID. As a result we get much more efficient use of bandwidth.

When one adds a module to the I/O configuration of a controller, the **Requested Packet Interval (RPI)** must be entered as a parameter. This value specifies how often to produce the data for that device. For example, if one specifies an RPI of 50 ms, it means that every 50 ms the device should send its data to the controller or the controller should send its data to the device.

The following table (4.2) categorizes the message Transport Classes s`upported by EtherNet/IP.

Table 4.2 Traffic classes

Forms of messaging	**Explicit Messaging Unscheduled TCP/IP**	**Implicit Data Transfer Scheduled UDP/IP**
Unconnected	UCMM (	n/a
Connected	Class 3 (T3)	Class 1 (T1)

4.3. Network Topology

When building a network, users 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/IP. Depending on the network configuration, an Ethernet hub or switch is appropriate.

Hub is an inexpensive connectivity method that provides an easy method of connecting devices on information networks (shared Ethernet).

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.

To successfully apply EtherNet/IP in automation, the issue of determinism has to be considered. The inherent principle of the Ethernet bus access mechanism – whereby collisions are detected and resolved using CSMA/CD protocol – cannot guarantee determinism.

First of all, the typical hubs used in an office environment have to be replaced by intelligent switches that will forward only those Ethernet frames that are intended for nodes connected to this switch. With the use of switch technology, collisions are largely avoided except for those cases where two or more messages are sent to the same node at the same time. The situation can be further improved by switching to a higher baud rate without increasing the number of messages. This will result in lower bandwidth utilization and will thus reduce the chance of collision.

Typically an EtherNet/IP network uses an active star topology where 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. The use of Ethernet controller chips, wiring and switches that support **full duplex** operation eliminates collisions on the network and permits a node to simultaneously transmit and receive messages effectively (Table 4.3). When implemented, full duplex increases the level of determinism.

Table 4.3 Half-duplex v. full duplex ETHERNET

Half-duplex Ethernet	Full-duplex Ethernet
• Can work in broadcast mode • Mandatory to support • When two devices try to transmit at the same time a collision occurs • Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol resolves collisions, but generates delays	• Restricted to point-to - point links connecting exactly two stations • Requires switching • Each side sends with no blocking on anything • Receiving is independent of sending • Quality of service better than half-duplex • The preferred real-time mode

4.4. Quality of Service (QOS)

When building a control network, it is important to understand the relative importance the various quality of service (QoS) measurements have on the various types of messages.

A summary is provided in Table 4.4. The definition of these rough classes is based on experiences with existing classes of applications in bus technologies.

Tab. 4.4 Message Categories vs QoS

QoS Class	Application	QoS Latency	Jitter
1	Controller-to-controller	100 ms	-
2	Distributed I/O devices	< 25% packet interval	Up to a tolerable maximum
3	Motion control	*≥1ms*	≥1µs

The following QoS principles are employed in the networks:

- Distinguishing one traffic stream from another (classification)
- Assigning a label to each kind of traffic
- Providing different treatment to different traffic classes

EtherNet/IP specification ed. 1.6 defines the behavior of EtherNet/IP networks with respect to QoS. The overall approach calls for devices to mark their packets with a priority value, using IEEE 802.1D/Q standard or DiffServ Code Points. Switches and routes must be able to differentiate real-time traffic from non-critical traffic streams (e.g. implicit vs. explicit messages).

Several different QoS mechanisms have been defined for network protocols. For EtherNet/IP one of the examples is the IEEE 802.1D/Q standard. Using priorities established in IEEE 802.1D/Q creates an effective method for isolating time-critical and best-effort data.

IEEE 802.1D defines the use of priority in the IEEE 802.1Q format, while the IEEE 802.1Q standard was developed to address the problem of how to break large networks into smaller parts so that broadcast and multicast traffic wouldn't grab more bandwidth than necessary. The standard also helps provide a higher level of security between segments of internal networks. The 802.1Q specification establishes a standard method for inserting virtual LAN (VLAN) membership information into Ethernet frames.

The general QoS approach to EtherNet/IP calls for devices to mark their packets with a priority value. From this marking, switches are able to differentiate EtherNet/IP traffic from non-real time traffic as well as differentiate I/O Data from Explicit Messages.

For example, the QoS behavior of EtherNet/IP is supported by the following solutions:

- For CIP transport class 0 and 1 UDP connections, there is a defined mapping of CIP priorities to 802.1D priorities (Layer 2) or other standard codes of differentiated services (Layers 3:RFC 2475)
- For CIP transport class 3 connections (TCP), there is a single defined 802.1D priority value

4.5. Further reading and references

Further information can be found at the sources indicated below.

http://www.odva.org

http://www.rtaautomation.com

http://en.wikipedia.org/wiki/EtherNet/IP

http://www.cisco.com

Figure and table sources:

[1] - http://www.ethernetip.de Industrial Ethernet Symposium, Amsterdam 2005

[2] http://www.ethernetip.de/ Developer Guide

Authors: Wojciech Modzelewski, W. Grega