Principles of EtherNet/IP Communication

Schneider Electric

Contents

Introduction

The Common Industrial Protocol, also known as CIP, is an application layer protocol that provides connection between industrial devices across multiple networks. CIP protocol is the common application layer across EtherNet/IP, CompoNet, DeviceNet, and ControlNet. CIP gives the user the ability to transport control oriented data associated with I/O devices and other information such as configuration parameters and diagnostics.

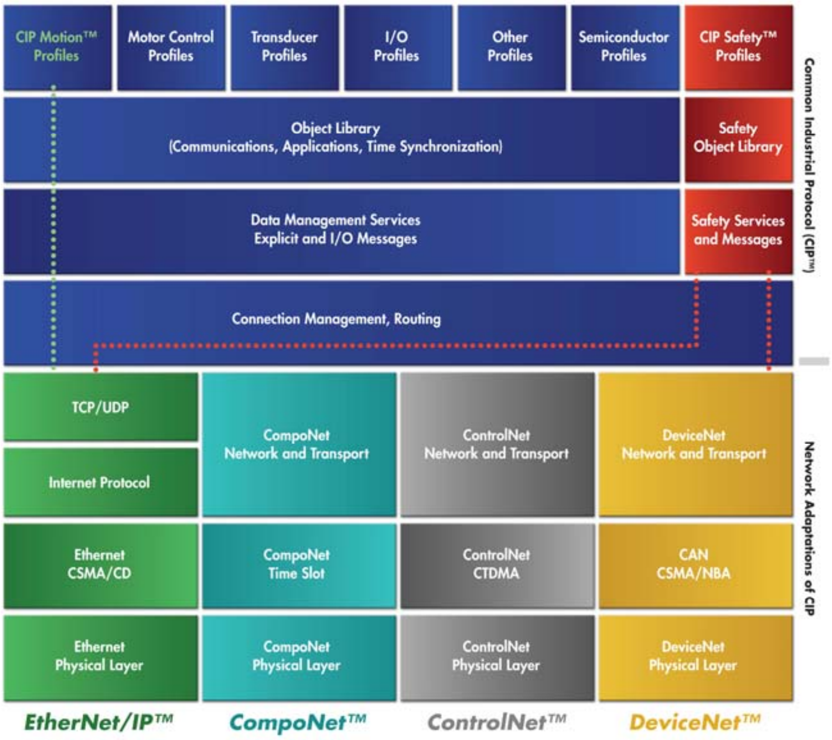

Figure 1: CIP Protocol Stack

CIP uses object modeling to describe the structure, operation and functionality of devices. A CIP node is described by a collection of objects and device profiles that define common interfaces and behaviors. Most objects are common across all supported networks. Objects divide the functionality of a device into logically related subsets. These objects interact to provide a basic product behavior.

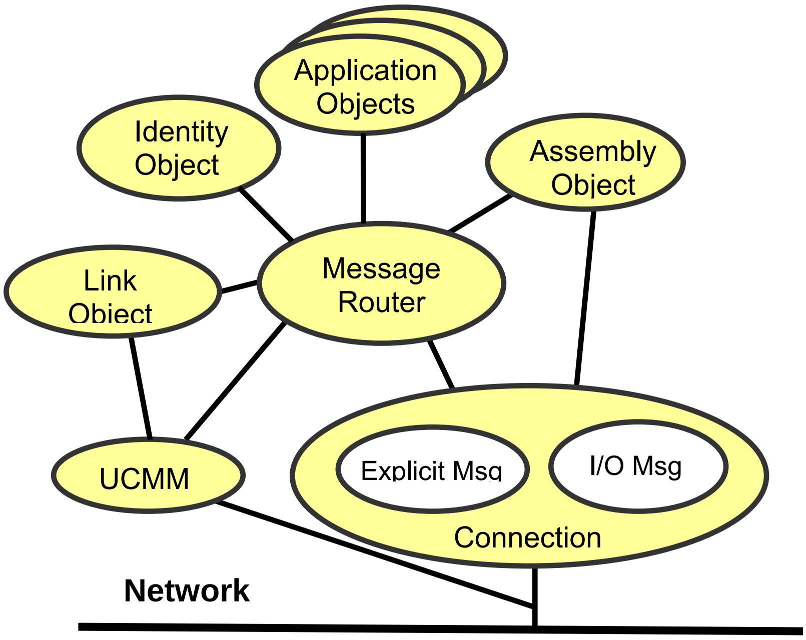

Figure 2: Object Oriented Data Model

EtherNet/IP is the implementation of CIP protocol over standard Ethernet. EtherNet/IP defines the encapsulation protocol used to structure the CIP data found in the TCP Data field. All encapsulated TCP or UDP messages are sent to port 0xAF12. The encapsulation message consists of two parts:

- Encapsulation Header
- Encapsulation Data

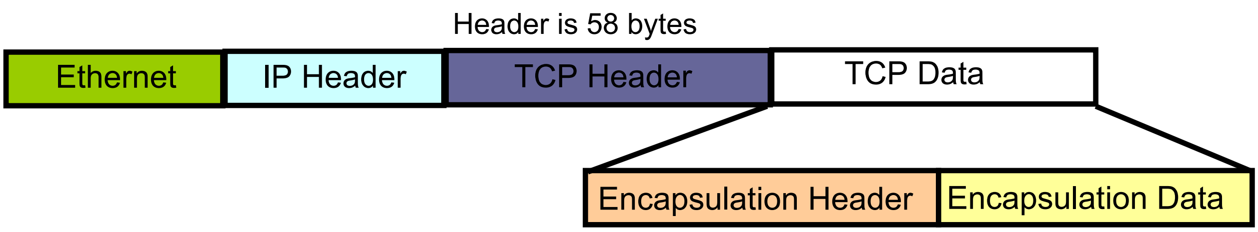

Figure 3: EtherNet IP Encapsulation Format

The encapsulation protocol defines the communications relationship between two nodes as a Session. A session is similar to a connection. It sets up the TCP resources, i.e. "opens a socket". Once a session is opened, it remains open until:

- The target or originator closes the TCP resources
- The TCP resources breaks down, i.e. reaching the limit for TCP connections
- Either one of nodes closes the session

Producer/Consumer Model

The EtherNet/IP protocol uses the producer/consumer architecture for I/O data transfer.

- Producers are the field devices that generate data onto the network. Generally, Producers generate data at a pre-established rate, without requiring a request to be issued each time. Producers are equivalent to Servers in a Modbus environment. This pre-established rate is referred to as RPI, Request Packet Interval. The RPI is a configurable parameter and part of the module's configuration.
- Consumers are the devices on the network that make use of the data generated by Producer devices. These devices establish the rules for the data to be generated by other devices. If necessary the Consumers will write any required output data to the Producers at the same RPI rate. Consumers are equivalent to Clients in a Modbus environment.
- Any device may act as a Producer, a Consumer or both, depending on the device's capabilities.

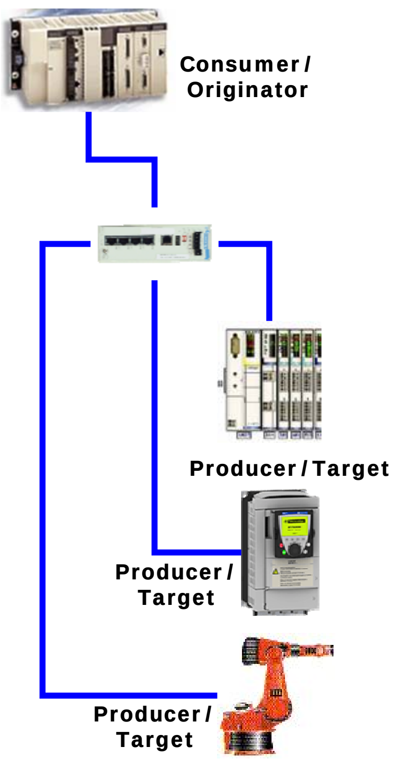

Figure 4: **Producer/Consumer Model**

When talking about I/O Data exchange you may also hear terminology such as originator or target devices.

- Originators are devices that initiate any data exchange with other devices on the network. This applies to both I/O communications and service messaging. This is the equivalent of the role of a Client in a Modbus network.
- Targets are the devices that address any data requests generated by Originators. This applies to both I/O communications and service messaging. This is the equivalent of the role of a Server in a Modbus network.

Some Consumer devices are capable of assuming the role of a producer. Schneider Electric's implementation of peer to peer communication makes use of this feature. In order to synchronize data from one PLC system to another, the EtherNet/IP module in the PLC rack will assume a role of a Target device simultaneously to their role as an Originator.

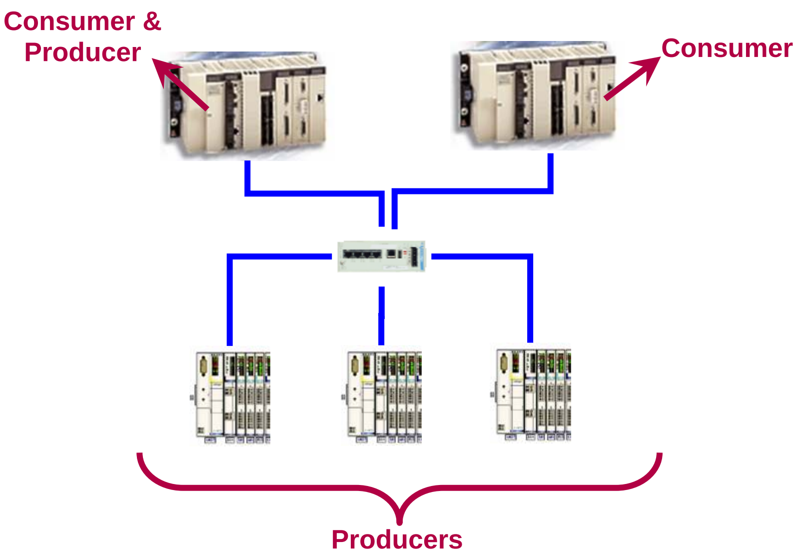

Figure 5: Dual Role Producer/Consumer Model

EtherNet/IP Messaging Types

EtherNet/IP defines two types of messaging types, Implicit, and Explicit Messaging.

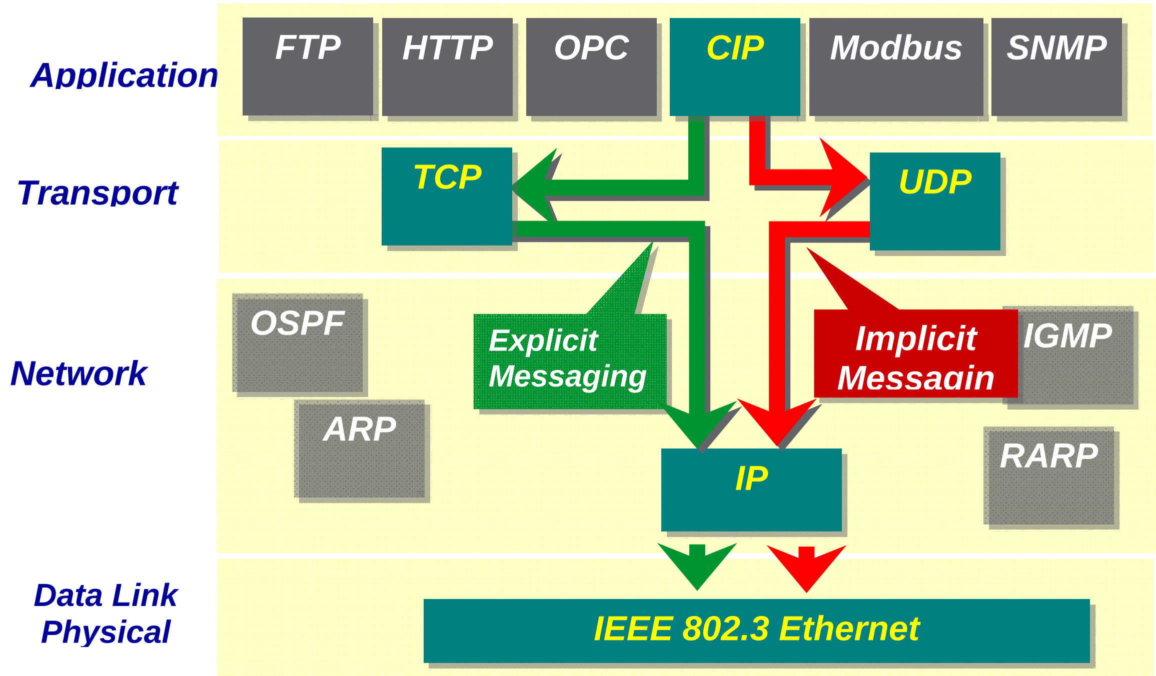

Figure 6: Implicit and Explicit Messaging

Implicit messaging is used for I/O data exchange. It is based on UDP protocol for faster delivery and optimized performance, due to its low overhead and resource consumption. The EtherNet/IP communication module on the PLC is the Originator and requests data from the Target devices. The Originator dictates the RPI, rate at which it wants to exchange the data with each Target device in its configuration; using a specific EtherNet/IP command called Forward Open.

Once the Target devices acknowledge the connection they will start producing the requested data at the rate indicated by the Originator. If required, the Originator will also start writing any data onto the Target devices at the same rate.

- Message construction:
  - The initial communication from the Originator to the Target is a Unicast* TCP message.
  - The data produced by the Target devices are Multicast** UDP messages, unless the user configures them differently.
  - The "write" commands from the Originator to the Target are Unicast TCP messages.

* Unicast message is communication between a single sender and a single receiver
** Multicast message is communication between a single sender and multiple receivers.

As mentioned Implicit messaging is used when I/O data must be exchanged at a constant rate. Such as:

- Scanning various IO modules
- Updating a variable speed drive
- Reading input data on sensors

As an example, a Consumer is requesting data to be provided to it every 20ms. After a successful Forward Open message the Producer will publish its data to a multicast address every 20ms. The Consumer is a subscriber to that multicast address and it is able to obtain the required information. At the same time the Consumer will update the Producer's output data at the same rate of 20ms but using a TCP Unicast message.

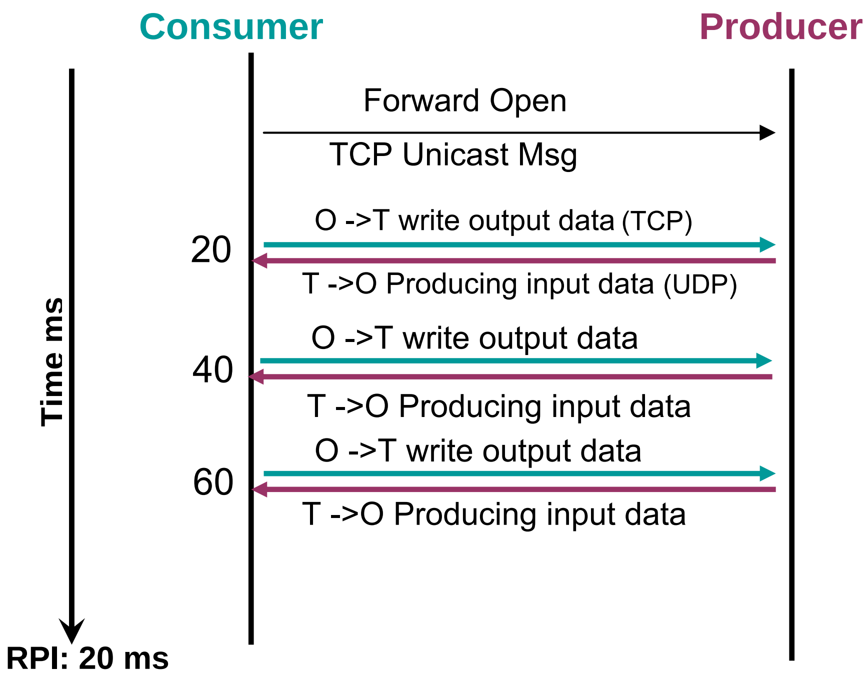

Figure 7: Sample Message Flow

Explicit messaging is used for exchange of non-cyclic and non-time critical data between devices in the EtherNet/IP network. Explicit messages are TCP/IP based messages where the EtherNet/IP module sends a request to the Target device and the Target device responds to that request. Explicit messages are TCP base messages.

- Message construction:
    - The initial request from the Originator to the Target is a Unicast TCP message.
    - The responses from the Target to the Originator are also Unicast TCP messages

Explicit messages are typically used to obtain information from the target devices. Some of the services include:

- Get_attribute_single
- Reset
- Start
- Stop

Real Time Data Exchanges

The following are the most typical forms of data exchanges found on an EtherNet/IP network.
Note: T = Target; O = Originator

a) PLC to I/O Target Device

Function	Packet type	Direction	Comm. Type
Configuration i.e. Parameter, RPI	TCP	O → T	Unicast
Implicit Messaging	UDP	T→ O	Multicast
	TCP	O → T	Unicast
Explicit Messaging	TCP	T←→ O	Unicast

b) PLC to PLC

Function	Packet type	Direction	Comm. Type
Implicit Messaging	UDP	T→ O	Multicast
	TCP	O → T	Unicast
Explicit Messaging	TCP	T←→ O	Unicast

c) HMI

Function	Packet type	Direction	Comm. Type
Explicit Messaging	TCP	T←→ O	Unicast

c) SCADA

Function	Packet type	Direction	Comm. Type
Explicit Messaging	TCP	T←→ O	Unicast
Configuration	Custom protocols or drivers required		

Typical Network Implementation

The following is an example of a typical EtherNet/IP network.

- PLC: Schneider Electric Quantum
- Network modules: Schneider Electric 140 NOC 771 00 EtherNet/IP network modules
  - Module Capacity: 7500 packets per second maximum (for Implicit and Explicit messaging)
- 60 Target EtherNet/IP Devices being scanned by the network module
- Network transmission rate: 100 Megabits per second (Mb/sec)

Network parameters

- I/O communications should not exceed 90% of the EtherNet/IP module's I/O capacity
- On average, each packet contains 58 bytes of I/O module specific data, leading to a total packet size of 100 bytes

Network load

- I/O impact - On average, a typical Ethernet implementation will consume less than 6% of bandwidth to support I/O communications.

Module capacity	7 500	Packets per second
Allocated capacity	90%	
	6 750	Packets per second
I/O data size	58	Bytes per packet
Ethernet header* size	42	Bytes per packet
Total packet size	100	Bytes per packet
Transmission total	675 000	Bytes per second
	5 400 000	Bits per second
Line capacity	100×10^6	Bits per second
Line utilization	5.4%	

*Ethernet header includes the following: Ethernet, IP and UDP header
Load calculation: Load = (number of packets per connection) x (number of connections)/ RPI

In this case, the maximum load of the module is known and the RPI must be determined: (number of packets per connection) x (number of connections) /Load = RPI
(2 x 60)/6750 = 0.018 seconds or RPI = 18 ms

The above load impact is on a link with most traffic, which is the connection between the switch and the Originator. The traffic on the link to the Target devices will be fraction of that load. Approximately, the load on each of the Target's link will be around 0.09%.

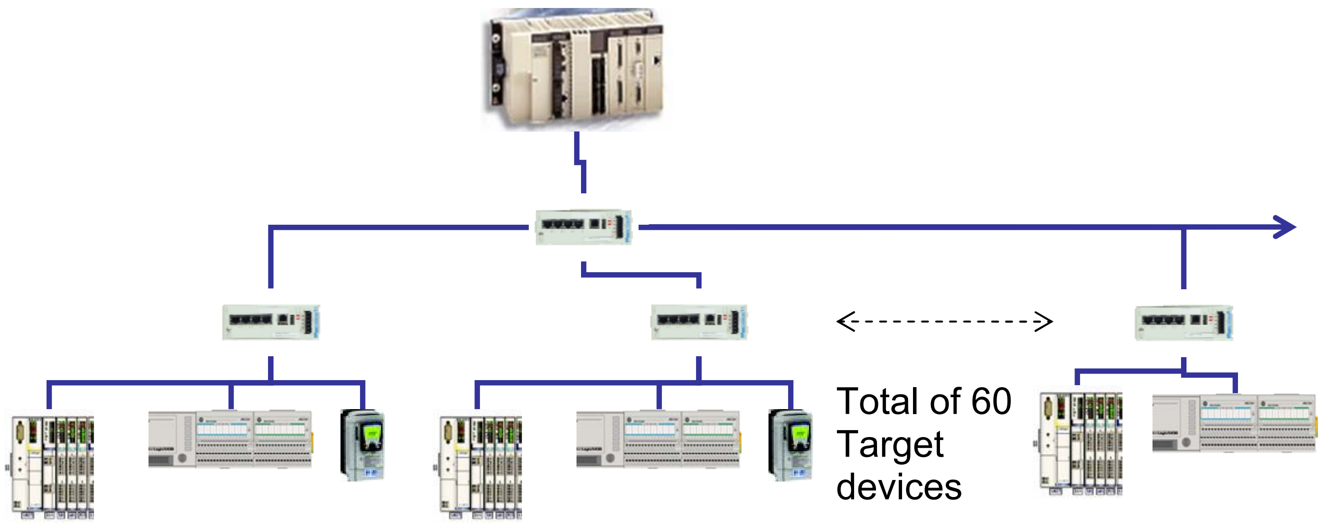

Figure 8: Example of EtherNet/IP Network

> Make the most of your energy

Schneider Electric Industries SAS

35, rue Joseph Monier
F-92500 Rueil-Malmaison
FRANCE
Tél. : +33 (0) 1 41 29 70 00

www.schneider-electric.com

Whitepaper by Nazie Champey
Design: Schneider Electric