Industrial Ethernet - Challenges and Drawbacks

Comparison

of

MODBUS TCP/IP

and

EtherNet/IP

Bachelor Thesis

Fall semester 2010/2011

Markus Klein

0726101

markus.klein@tuwien.ac.at

Supervisor

Ao. Univ. Prof. Dr. Wolfgang Kastner

Statement

Hereby I declare that this work has been written autonomously, that all used sources and utilities are denoted accordingly and that these points of the work - including tables, maps and figures - which where taken from other creations or the Internet have been marked as borrowing by quoting the original sources. This document at hand will not be submitted to any other course.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Contents

List of Tables

List of Figures

Abstract

This paper provides a general overview of industrial communication on fieldbus level using state-of-the-art Ethernet-based protocols.

After having a look at the history of fieldbuses and the legal standards, this paper builds the bridge to modern Ethernet-based fieldbus protocols, which are designed to match today's requirements in industrial automation.

Special focus is on the protocols EtherNet/IP, based on the sophisticated Common Industrial Protocol, and MODBUS TCP/IP, the Ethernet-based version of the former de-facto industry standard MODBUS.

Finally, this report compares the two protocols, putting specific attention on some important aspects and challenges of these protocols.

1 Introduction

1.1 Industrial Communication

Machines, possibly consisting of several dedicated devices, have to talk to a controller or with each other. The underlying communication protocol builds the backbone of any automation system.

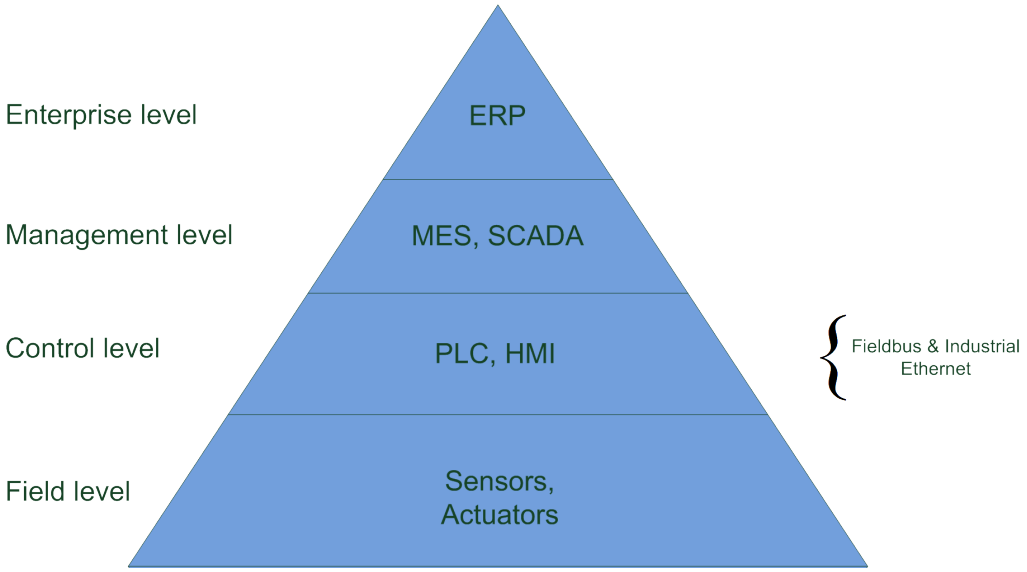

Figure 1.1: Automation pyramid [13, p. 536]

The automation pyramid in Figure 1.1 depicts the levels of automation within a factory. Each of these levels has its own requirements of communication, called industrial communication.

Industrial communication is used since the early 1980s. Until now, the technologies were undergoing massive changes as new demanding requirements sought for new ideas.

Today's protocols and devices break with the traditional view of the automation pyramid (Figure 1.1). As embedded CPUs are getting faster and smaller, sensors and actuators are getting more intelligent. These intelligent devices are subsumed under the term "Smart Instruments". Smart Instruments may include the necessary hardware to be directly in-

tegrated into the control level of the automation pyramid, mixing up the classic definition of the levels. Thus, it is suggested to distinguish between the functional architecture and the operational architecture. [14, p. 84]

Some advantages of this integration process are:

- Faster access to devices.
- Direct access to devices' data for immediate evaluation in MES or SCADA systems.
- Cost saving for hardware like cabling on the field level.

1.1.1 Fieldbuses

A fieldbus is a communication system that interconnects field level devices and builds the bridge to control level PLCs. The very first system, used in an experiment, was built in France, 1981 [14, p. 82]. Following this experiment a working group was established, creating the FIP (now WorldFIP) fieldbus. Other groups in different countries also developed new bus systems, leading to a variety of protocols during the 1980s.

Facing this wide range of fieldbuses leads to the question "What is a fieldbus?". J.P. Thomesse states two objectives, how a fieldbus may be seen by a designer: [14, p. 81]

- A fieldbus is only a network for simplifying the wiring between devices.
- A fieldbus is the spinal column of a distributed real-time system.

Thomesse furthermore describes the main requirements important for all fieldbuses from the end-user's point of view: [14, p. 83]

- Safety, availability and dependability
- Better maintainability
- Better modularity, and capacity for evolution

- Openness, interoperability, interchangeability and long life-times
- Better performance and lower costs

1.1.2 Standardization

In parallel to the development of over 50 fieldbuses, various committees of standardization organisations were working on fieldbus standards, paving the way for better interoperability between fieldbuses.

In Europe, FIP and PROFIBUS were popular fieldbuses. Both came from different countries and both tried to bring their approach into an international standard. This was one reason for the so called “Fieldbus War”, preventing a worldwide standard.

Finally, after more than ten years passing by, the war was put to an end. IEC 61158 (1999), the long awaited international standard for fieldbuses, was released. The standard is not based on an agreement of the competing companies, but simply structures all of the present approaches and combines them to a single document structure. The additional set of standards in IEC 61784 define profiles, telling how to combine the modules of IEC 61158 for the specific fieldbuses.

1.2 Ethernet in Automation

Ethernet has been used on the management level and enterprise level for a long time, connecting any kind of device using Internet technology. Bringing this technology one level down to the control level is the next step of the integration process. Considering the levels of the automation pyramid this means, that the fieldbus directly connects to a company’s LAN/intranet removing a complete (operational) level from the pyramid.

This once again helps to reduce costs and increases flexibility in today’s production processes.

Increasing needs for transportation of bigger data quantities and low cost circuits for Ethernet are reasons for manufacturers to create protocols on basis of Ethernet, IP and TCP/UDP. [1, p. 1102]

In [2, p. 1118], Max Felser defines two requirements for using Ethernet on fieldbus level:

- Support migration of the office Ethernet to real-time Ethernet
- Use of standard components: bridges, Ethernet controllers and protocol stacks as far as possible

Tackling the problem of non-determinism of the Ethernet protocol on one hand and the need for guaranteed real-time transmissions on the other hand led to the development of diverse modifications of the standard Ethernet and TCP/IP protocols.

In [1, p. 1114], Decotignie lists the following proposals for "Industrial Ethernet" supporting real-time communications: PROFINET, Ethernet Powerlink, JetSync, EtherNet/IP, SERCOS III, Modbus-TCP, EtherCAT, PowerDNA, Real-Time Publish-Subscribe and SynqNet. Some of today's big players are MODBUS TCP, PROFINET IO, EtherNet/IP, Ethernet Powerlink and EtherCAT.

[2, p. 1121] shows the possible structures used to realize real-time Ethernet. The following list briefly describes the different approaches:

- Top of TCP/IP: Places the real-time components on top of the standard TCP/IP protocol stack (e.g. EtherNet/IP).
- Top of Ethernet: Modifies or replaces the TCP/IP stack with the real-time protocol, but doesn't touch the Ethernet layer (e.g. PROFINET CBA).
- Modified Ethernet: Provides a dedicated real-time protocol, that includes a modified Ethernet layer (e.g. EtherCAT).

No question, Ethernet will pave its way in industrial automation. As a single versatile protocol Ethernet will be the best solution for the majority of cases. Unfortunately, current evolution shows a similar development of industrial Ethernet standards as it has been the case with fieldbus standards. In 2005, IEC 61784 already contained ten real-time Ethernet

profiles. In the latest revision of the standard, this has been expanded with two additional profiles, namely: RAPIEnet (CFP17) and SafetyNET p (CFP18). [3]

Max Felser ends his paper [2] with the question: "*Is it up to the end user and the market to decide which one of the proposed solutions fulfils the requirement of the automation applications and will end up in real applications?*" In 2011, the answer to this questions seems to be "Yes", but to say it with the words of J.P. Thomesse: "*The world is still not stable.*" [14, p. 91]

2 EtherNet/IP

The full name of EtherNet/IP is Ethernet **Industrial Protocol** derived from the underlying CIP (Common **Industrial Protocol**). EtherNet/IP is often confused with a combination of Ethernet and Internet Protocol, well known from the TCP/IP model, used in a wide range of network applications. What is even more confusing is the fact, that EtherNet/IP as an application layer protocol (refer to OSI model [4]) actually operates over Ethernet (physical layer) and the Internet Protocol stack (network and transport layer).

2.1 Description

EtherNet/IP follows the Open Systems Interconnection (OSI) model with its seven layers [4]. As depicted in Figure 2.1 and like all other CIP protocols, this protocol implements CIP at the top three layers (session, presentation and application). The remaining four layers are adapted to the specific EtherNet/IP technology using an **unmodified** TCP/UDP/IP stack.

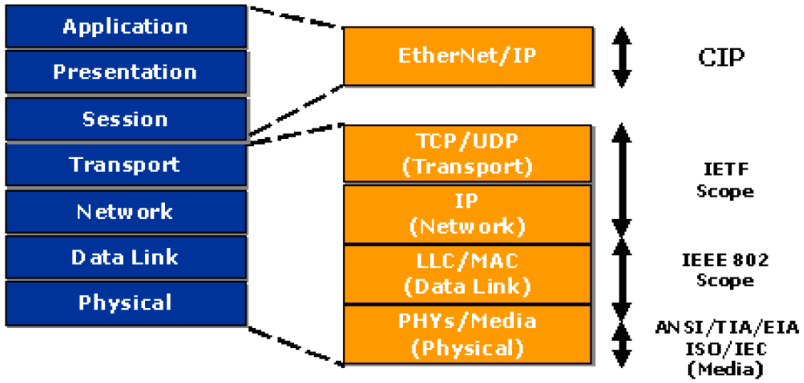

Figure 2.1: EtherNet/IP Protocol Stack [12, p. 6]

2.2 History

The EtherNet/IP protocol was initially developed by Allen-Bradley (Rockwell Automation, USA) in the late 1990s.

Since 2001, EtherNet/IP is maintained by the Open DeviceNet Vendors Association (short ODVA[1]), which is an international organization supporting technologies based on CIP.

Currently, ODVA has over 500 registered vendors, including leading automation companies, and is responsible for ensuring multi-vendor interoperability.

2.3 Protocol Specification

The EtherNet/IP application layer protocol is based on the Common Industrial Protocol (CIP) standard, which provides compatibility to DeviceNet, CompoNet and ControlNet. [9, p. 2]

CIP, formerly known as Control and Information Protocol, is a collection of messages and services with the focus on manufacturing automation applications providing a unified communication architecture. Due to the generic definition of messages and services, CIP is truly media-independent (refer to Figures 2.2 and 2.3).

CIP uses an object model approach containing data addressing methods and message exchange rules.

This approach is strictly object-oriented at the upper layers of the protocol. An object is a collection of attributes, services, connections and behaviours. For standard purpose tasks like analog/digital I/O, file transfers or position feedback, CIP provides an object library. The communication model is based on the producer-consumer model, enabling a flexible and efficient usage of limited network resources.

Device interoperability is assured by device profiles defining groups of available objects together with configuration options and I/O data formats.

[1] http://www.odva.org/

Figure 2.2: Services provided by CIP

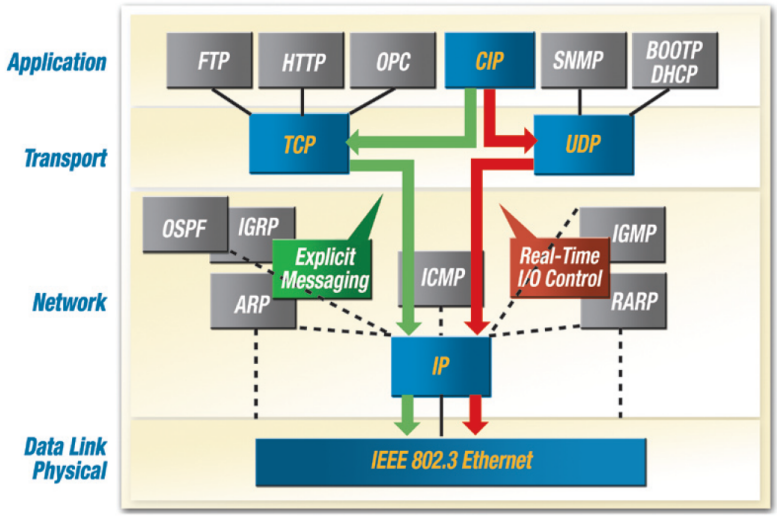

Figure 2.3: CIP multi-protocol support [12, p. 6]

2.3.1 Data model and addressing

As described, CIP uses an object oriented approach for defining a device's capabilities. Therefore, CIP specifies three types of objects a device may contain: [12, p. 8]

- Required Objects: These objects are mandatory to any CIP device and include the Identity Object, the Message Router Object and network-specific objects.
- Application Objects: Depending on the device type and its function, these objects define, how data is encapsulated by a device.
- Vendor-specific Objects: Objects describing vendor-specific services, which are not listed in a device profile.

The **Identity Object** (class ID 0x01) is a read-only object (except for one attribute) that defines basic information about a device. Mandatory attributes of this class are: Vendor ID, Device Type, Product Code, Revision, Status, Serial Number and Product Name.

The **Message Router Object** of a node takes care of distributing explicit message requests to the appropriate application objects. [10]

As multiple instances of an object can co-exist in one device, this set of instances is referred as Class. This may be confusing when compared to object oriented approaches in software engineering (SE), as an object in the SE world is commonly known to be an instance of a class. So to build the bridge:

- A class in SE world is an object in CIP world.
- A class in CIP world is a set of objects instantiated from the same class in SE world.

Furthermore, CIP also specifies the methods used to access the data. Assembly Objects allow vendors to define a message (I/O or configuration) containing a variety of data from different other objects. E.g. an

Assembly Object can be a collection of several attributes of diverse application objects.

An interesting aspect of EtherNet/IP (CIP) is the very fine grained addressing schema for objects. [12, p. 9]

An object address (called CIP Segment) is well structured and is assembled by the the following parts:

- Device network address: Can be a node address or a medium access control identifier.
- Class ID: This holds the ID of the class this instance belongs to.
- Instance ID: The instance itself.
- Attribute ID: The attribute that is of interest.
- Service code: Describes the action/service required by this request.

Each part of a CIP Segment can have various formats: 1 byte, 2 byte or 4 byte. [10, p. 25]

2.3.2 Services

"*CIP is a connection-based protocol. A CIP connection provides a path between multiple application objects. When a connection is established, the transmissions associated with that connection are assigned a Connection ID.*" [10, p. 13]

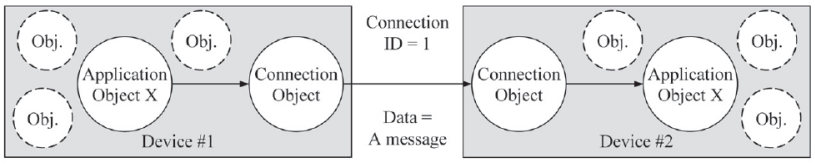

Figure 2.4: CIP connection and Connection ID. [10, p. 13]

To establish such a connection, the Unconnected Message Manager (UCMM) function is used, which processes unconnected explicit requests

and responses. The UCMM Forward_Open service request message contains all required information to initiate a connection between the originator and the target device.

Depending on the typical form of communication, EtherNet/IP defines two types of messages. Table 2.1 provides an overview which types are available and which protocols (on lower OSI layers) are used. [12, p. 11]

CIP Message Type	Explicit	Implicit
CIP Communication Relationship	Connected or Unconnected	Connected[1]
Transport Protocol	TCP/IP	UDP/IP
Communication Type	Request/reply transactions	I/O data transfers
Typical Use	Non time-critical information data	Real-time I/O data

[1] "*With Implicit Messaging you establish an association (a 'CIP connection') between two devices and produce the Implicit Messages according to a predetermined trigger mechanism,...*" [12, p. 11]

Table 2.1: EtherNet/IP message types. [12, p. 11]

Implicit messaging has the big advantage, that it can make use of the producer/consumer principle. This means that it is only necessary to send I/O data once and all interested receivers can read the data, helping to massively reduce load on the bus in cases where many devices send lots of data on high rates. Additionally, there is no request/response pattern for implicit messages once the connection has been established, reducing bus load even more. The sending of implicit messages can either be triggered cyclically or at change of state or due to an application specific reason.

3 MODBUS

3.1 Description

MODBUS is a powerful application layer [4] messaging protocol. It is solely based on the client/server principle and can be operated over several buses and networks.

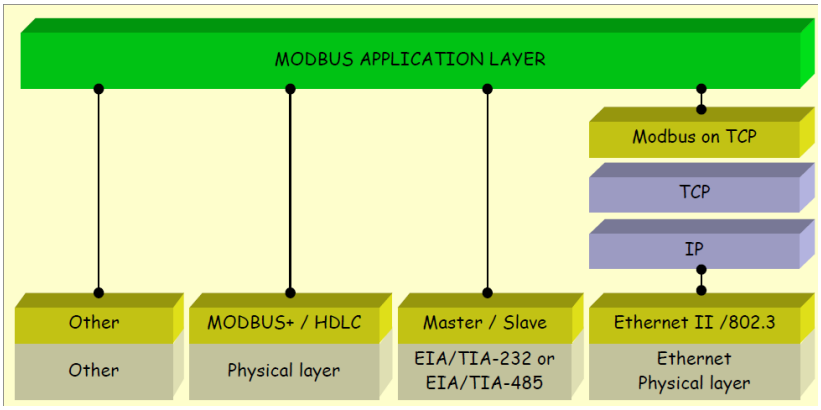

Figure 3.1: MODBUS communication stack [6, p. 2]

Figure 3.1 shows the main three buses that may be utilized by MODBUS. Communication between these buses in heterogeneous networks is easily possible, since the simple structure of the protocol allows building of inexpensive gateways.

Modbus-IDA focuses on driving the evolution towards the MODBUS TCP/IP protocol.

Although MODBUS is only a tiny protocol compared to other Ethernet based protocols, there is no doubt that it will not lose its position in industry. Integration of MODBUS devices will even be more comfortable, since big players, like EtherNet/IP in 2007, introduced additional specifications for mapping MODBUS based devices directly into their architecture. (Compare also “CIP network specification” - Volume seven, [11] and “*CIP-Modbus Integration*” [15])

CIP-to-Modbus integration supports all kinds of messaging and performs necessary translations to make MODBUS device data consistent

with the CIP communications model. Figures 3.2 and 3.3 show how the integration is achieved.

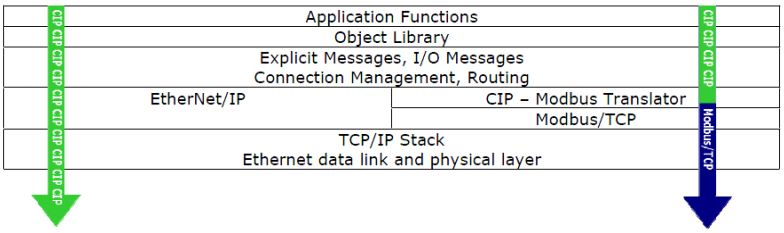

Figure 3.2: EtherNet/IP Originator with embedded MODBUS integration [11, p. 2]

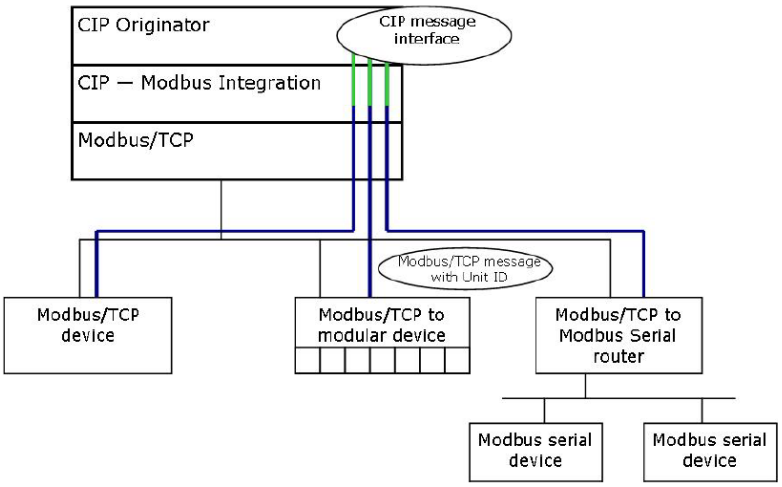

Figure 3.3: CIP Originator to Modbus/TCP Target Devices [11, p. 3]

3.2 History

MODBUS was originally developed by Modicon, today Schneider Electric, and is the de-facto standard for serial communication in automation industry since 1979. In April 2004, Schneider Automation transferred its copyright to Modbus-IDA, a non-profit organisation founded in 2002. Today, Modbus-IDA provides the infrastructure to distribute all kinds of

information about MODBUS. This also includes directories of suppliers and integrators as well a device database.

3.3 Protocol Specification

The basis of the MODBUS protocol is the General MODBUS frame, also called PDU (**P**rotocol **D**ata **U**nit), which only consists of two parts. It is completely independent of any underlying communication layer. These parts are:

- **Function:** A field of one byte containing the ID of the requested service.
- **Data:** A field of variable size containing the required payload for the requested service.

The so called ADU (**A**pplication **D**ata **U**nit) is the real frame containing the PDU and additional fields used by the underlying communication protocol. Figure 3.4 explains the nesting of ADU and PDU.

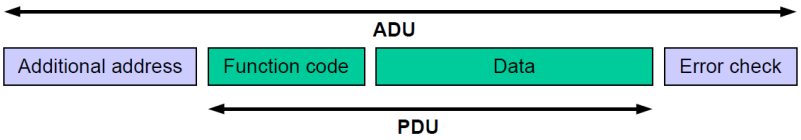

Figure 3.4: MODBUS Application Data Unit [6, p. 4]

As depicted, the ADU is terminated by the field “Error check”. Depending on the underlying communication protocol this field may be optional, if the functionality is already included in the particular protocol. For instance, TCP/IP already includes error checking and CRC, so this field is omitted.

3.3.1 Data model and addressing

The MODBUS standard defines four table types each of them containing entries with ascending index. Each table entry is a mapping from its index

to a device's physical application memory address. Table 3.1 describes the four types and what data may be lodged behind an entry.

Table type	Object type	Access
Discrete Input	Single Bit	Read
Coils	Single Bit	Read/Write
Input Registers	16-bit Word	Read
Holding Registers	16-bit Word	Read/Write

Table 3.1: MODBUS data model - table types

A table entry is addressed using its index, which has a width of 16 bits, so a maximum size of 65536 entries per table type is possible.

A completely different approach has also been tried by specifying "*Object Messaging Specification for the MODBUS/TCP Protocol*" [5].

3.3.2 Services

Transactions

Since MODBUS only supports the request/response pattern, sending of data can only be initiated by the client. The server (device) never sends anything without receiving an adequate request beforehand. Hence MODBUS only allows polling of attached devices.

One transaction involves the following steps:

1. A PDU is sent from one client to a server containing the requested service specified by the function code.
2. The server processes the request.
3. The received PDU (possibly filled with data) is returned to the client.
4. The client processes the response.

In case an exception arises on the server, step 3 is modified. The most significant bit in the function code of the PDU is set to logic one and exception data is optionally appended.

Function codes

Function codes specify the available services a device offers. The majority of codes are standardized, but a couple of codes are open to be implemented with vendor/user specific functionality.

Functions are available for different types of service: data access, diagnostics and others. Functions for data access are furthermore grouped into functions for Bit-access (Discrete Inputs and Coils), Word-access (Input Registers and Holding Registers) and file record access.

It is not required for a device to implement all functions codes. In case a client requests a function that is not implemented, an exception has to be returned. Refer to [6, p. 48] for a detailed description of exception responses.

3.4 Communication over TCP/IP

In cases where TCP/IP is the underlying communication protocol of choice, the "Additional address" field of the ADU in Figure 3.4 is replaced by the MBAP Header (MODBUS Application Protocol Header). The "Error check" field is omitted in favour of the error checking mechanisms of TCP/IP and Ethernet.

The MBAP Header structure is described in Chapter 3.1.3 in [7, p. 5].

For transactions over TCP/IP, port 502 has been reserved for MODBUS servers (devices). Any request ADU of a client will be addressed to the respective server on port 502. Message switching and routing are not in the scope of MODBUS, but of TCP/IP and therefore any device on the market can be used for these purposes.

The field "Unit Identifier" in the MBAP Header is of special interest when gateways are used to integrate devices e.g. connected via a serial line into the TCP/IP world. This field is used by the gateway to determine the destination of the request within the subordinated bus.

4 Conclusion

Modern automation requires high integration of management and controlling tasks into the production process. Cost efficiency, interoperability and interchangeability are more important than ever. To achieve such a high integration a common communication protocol, covering everything from the sensor to the SCADA system, is needed. Since Ethernet and TCP/IP have been a standard in corporate networks for years now, it seems reasonable to concentrate the development of fieldbuses on them.

Using Ethernet based solutions has big advantages for companies, such as:

- Lots of protocol specific know-how is already present in the company.
- Necessary equipment for TCP/IP networks is present as well, so it is not necessary to install additional cabling for the fieldbuses.
- Integration of high level management functions is possible, without the introduction of new interfaces, leading to cost reduction.

In the following sections the described Ethernet protocols EtherNet/IP and MODBUS TCP/IP are compared on aspects important for future automation challenges.

Although it is possible to integrate MODBUS into EtherNet/IP networks, it may be useful for companies to evaluate, which protocol better fits their needs as there may be certain issues, preventing a successful integration of older MODBUS devices into EtherNet/IP. [15]

4.1 Availability

This aspect mainly affects the costs of an automation system.

Whereas MODBUS is freely available to everyone, EtherNet/IP requires to be subscribed to ODVA[1] in order to obtain the specifications of the protocols.

The MODBUS specification can be downloaded from the Modbus-IDA Website[2] and is free of charge. In contrast, the EtherNet/IP specifications are sold on CDs and can cost up to several thousand dollars. For vendors it is highly recommended becoming a member of ODVA and gaining an official Vendor ID.

4.2 Data management and addressing

Both protocols can address the same number of devices. The effective amount only depends on the configured IP subnet.

The amount of addressable data (objects) differs quite a lot. As MODBUS reserves 16-bit for addressing, a maximum of 65536 items per table type can be addressed. EtherNet/IP is able to address a far greater range of objects (and attributes), depending on the chosen CIP Segment format.

Assuming the biggest format with 4 bytes, a total amount of 2^{32} instances per class ID and 2^{32} attributes per instance can be addressed. Each attribute can be one of the CIP data types, which follow the requirements of IEC 61131-3.

Last but not least MODBUS completely misses a standard how clients can gather meta-information of data objects. Examples of missing information are: valid range of a value, physical unit, scale, etc.

Due to the detailed specification of CIP, EtherNet/IP offers a far better basis for building interoperable and interchangeable devices, as the device profiles and electronic data sheets explicitly explain the provided data objects (including parameters and attributes) and services.

[1] http://odva.org/Home/MEMBERS/tabid/108/lng/en-US/language/en-US/Default.aspx

[2] http://modbus.org/specs.php

4.3 Performance

The maximum data rate the protocols offer, is primary depending on the underlying Ethernet protocol. Having nets with up to $10Gbit/s$, services with high data rate requirements are possible. However, with embedded processors we are facing the problem, that a device might not be able to process data with such high rates.

Moreover, taking a closer look at the protocols reveals other problems. Typically industrial communication data size is not bigger than a couple of bytes, but the minimum frame size of $1Gbit/s$ Ethernet is $512bytes$, resulting in a data efficiency of approximately 5% or less. Also application protocol efficiency is a factor. MODBUS has a quite good protocol efficiency of about 60%. But MODBUS only offers a request/response communication pattern, whereas EtherNet/IP additionally supports a producer/consumer pattern utilizing UDP. However, UDP has a much better protocol efficiency than TCP, because of the missing acknowledgement transactions.

4.4 Security

This topic is probably the most sensitive and problematic in today's information technology world. Being aware that security threats have been present for years, there have been rather little efforts to tackle this subject in automation. But things will change, at least since this omnipresent virus called Stuxnet has been around. [8]

Looking at these massive efforts taken to finally modify a PLC, it is clear that things will get even worse, when target devices are on the same net as the normal PCs, which will reduce the necessary steps to reach the final bus system to which the target is connected.

For instance, MODBUS itself only provides protection against malformed ADUs by checking the proper format of the frame upon receipt. All other potential security threats are related to the underlying communication technology and have to be addressed in these layers. For TCP/IP,

a properly configured firewall would help to protect the network from external attackers.

Having this in mind, it seems quite horrible, that neither MODBUS nor EtherNet/IP have any security mechanism. So if a fieldbus LAN is directly accessible by normal PC stations or even the Internet, one's really on high risk.

Both protocols lack of functions like access tables, defining which device on the net has the right to send a specific request or command. Since the traffic is not encrypted, all devices connected to the bus are able to listen to all transmissions and - even worse - are able to manipulate data.

To sum up, companies have to find a trade-off between high integration of fieldbuses to their company net and the risk of security issues.

4.5 Real-Time

Protocols for safety critical applications require one thing: real-time data transfers with deterministic response times.

Facing this requirement Ethernet-based applications seem to be a dead end. Due to the CSMA/CD principle used to control media access, Ethernet has an inherent non-determinism, which makes it not reliable enough to be used for safety critical applications, no matter how fast Ethernet will ever be.

Especially for MODBUS the available speed and response times are limited by Ethernet. It is not possible to reach the required response times for real-time applications like motion control without modifying the underlying technologies.

MODBUS itself has no capabilities of serving real-time services. Real-Time Publisher Subscriber (RTPS) protocol is a real-time extension for MODBUS. It uses the UDP protocol to create multicast messages used for the producer/consumer pattern. "*Contrary to the standard MODBUS protocol, the RTPS protcol is not used very much in practical industrial applications today, and therefore it is not known exactly what sort of performance this protocol really has to offer. ... required performance*

of the "process application class" [M.K.: less than $10ms$ *response time], ..., may be reached with this system.*" [2, p. 1122]

CIP also offers an add-on for real-time data transmission, called "CIP Sync". This extension removes the hard requirements from the Ethernet layers and establishes a common timebase throughout all device. The provided clock synchronisation algorithm achieves deviations of less than $200ns$, which is sufficient even for the most demanding motion control applications. All messages sent are time-stamped causing a small amount of jitter to be negligible.

Bibliography

[1] Jean Dominique Decotignie. Ethernet-based real-time and industrial communications. *Proc. IEEE*, 93(6):1102–1117, June 2005.

[2] Max Felser. Real-Time Ethernet - Industry Prospective. *Proc. IEEE*, 93(6):1118–1129, June 2005.

[3] International Electrotechnical Commission. IEC 61784-2, July 2010.

[4] International Organization for Standardization. ISO/IEC 7498-1, 1994.

[5] Modbus-IDA. *Object Messaging Specification for the MODBUS/TCP Protocol V1.1*, November 2004. URL http://modbus.org/docs/Object_Messaging_Protocol_ExtensionsVers1.1.doc.

[6] Modbus-IDA. *MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b*, December 2006. URL http://modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf.

[7] Modbus-IDA. *MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE V1.0b*, October 2006. URL http://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf.

[8] Eric Chien Nicolas Falliere, Liam O Murchu. Symantec security response w32.stuxnet dossier. Technical report, Symantec, February 2011. URL http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf.

[9] ODVA, Inc. *Common Industrial Protocol (CIP) (PUB122)*, 2006. URL http://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00122R0_CIP_Brochure_ENGLISH.pdf.

[10] ODVA, Inc. *The Common Industrial Protocol (CIP) and the Family of CIP Networks (PUB122)*, 2006. URL http://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00123R0_Common%20Industrial_Protocol_and_Family_of_CIP_Netw.pdf.

[11] ODVA, Inc. *Common Industrial Protocol (CIP) - Modbus Integration*, April 2008. URL http://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00193R0_CIP-Modbus_Integration_Overview.pdf.

[12] ODVA, Inc. *EtherNet/IP Quick Start for Vendors Handbook: A Guide for EtherNet/IP Developers (PUB213R0)*, 2008. URL http://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00213R0_EtherNetIP_Developers_Guide.pdf.

[13] Martin Polke. *Prozeßleittechnik*. Oldenbourg Verlag, 1994.

[14] Jean Pierre Thomesse. Fieldbuses and interoperability. *Cntr. Eng. Pract.*, 7(1):81–94, 1999.

[15] Todd A. Snide. Cip-modbus integration. Technical report, ODVA, Inc.,Modbus-IDA, April 2008. URL http://modbus.org/docs/CIP%20Modbus%20Integration%20Hanover%20Fair_0408.pdf.