CHAPTER

Industrial Network Protocols 6

INFORMATION IN THIS CHAPTER

- Overview of Industrial Network Protocols
- Fieldbus Protocols
- Backend Protocols
- AMI and the Smart Grid
- Industrial Protocol Simulators

Understanding how industrial networks operate requires a basic understanding of the underlying communications protocols that are used, where they are used, and why. There are many highly specialized protocols used for industrial automation and control, most of which are designed for efficiency and reliability to support the economic and operational requirements of large industrial control system (ICS) architectures. Industrial protocols are designed for real-time operation to support precision operations involving deterministic communication of both monitoring and control data.

This means that most industrial protocols forgo any feature or function that is not absolutely necessary for the sake of efficiency. More unfortunate is that this often includes the absence of even basic security features, such as authentication or encryption, both of which require additional overhead. To further complicate matters, many of these protocols have been modified to run over Ethernet and Internet Protocol (IP) networks as suppliers moved away from proprietary networks and networking hardware and leveraged commercial off-the-shelf (COTS) technologies. This, however, has now left these "fragile" protocols potentially vulnerable to cyber-attack.

OVERVIEW OF INDUSTRIAL NETWORK PROTOCOLS

Industrial network protocols are deployed throughout a typical ICS network architecture spanning wide-area networks, business networks, plant networks, supervisory networks, and fieldbus networks. Most of the protocols discussed have the ability to perform several functions across multiple network zones, and so will be referred to here more generically as industrial protocols.

Industrial protocols are real-time communications protocols, developed to interconnect the systems, interfaces, and instruments that make up an industrial control system. Many were designed initially to communicate serially over RS-232/485 physical connections at low speeds (typ. 9.6 kbps to 38.4 kbps), but have since evolved to operate over Ethernet networks using routable protocols, such as TCP/IP and UDP/IP.

Industrial protocols for the purposes of this book will be divided into two common categories: fieldbus and backend protocols. Fieldbus is used to represent a broad category of protocols that are commonly found in process and control (see Chapter 5, "Industrial Network Design and Architecture"). Beginning in the early 1980s, there was a push from ICS vendors and end users to establish a global fieldbus standard. This effort continued for over 20 years and resulted in the creation of a wide range of standards devoted to industrial protocols. The IEC 61158 standard was one of the early documents that established a base of eight different protocol sets called "types." Some of the major protocols at that time (HART and Common Industrial Protocol or CIP to name a few) were missing from this list. The IEC 61784 standard was introduced in the early 2000s to amend the list originally contained in the IEC 61158 standard, and includes a total of nine protocol "profiles": FOUNDATION Fieldbus, CIP, PROFIBUS/PROFINET, P-NET, WorldFIP, INTERBUS, CC-Link, HART, and SERCOS.[1] Fieldbus protocols in this book are commonly deployed to connect process-connected devices (e.g. sensors) to basic control devices (e.g. programmable logic controller or PLC), and control devices to supervisory systems (e.g. ICS server, human–machine interface or HMI, historian).

Backend protocols are those protocols that are commonly deployed on or above supervisory networks, and are used to provide efficient system-to-system communication, as opposed to data access. Examples of backend protocols include connecting a historian to an ICS server, connecting an ICS from one supplier to another supplier's systems, or connecting two ICS operation control centers.

Four common industrial network protocols will be discussed in some depth, others will be touched upon more briefly, and many will not be covered here. There are literally dozens of industrial protocols, many developed by manufacturers for their specific purposes. The two fieldbus protocols analyzed include the Modicon Communication Bus (Modbus) and the Distributed Network Protocol (DNP3). Two backend protocols will also be discussed in detail; Open Process Communications (OPC) and the Inter-Control Center Protocol (ICCP, also referenced by standard IEC 60870-3 TASE.2 or Telecontrol Application Service Element). These particular protocols have been selected for more in-depth discussion because they are all widely deployed and they represent several unique qualities that are important to understand within the context of security. These unique qualities include the following:

- Each is used in different (though sometimes overlapping) areas within an industrial network.
- Each provides different methods of verifying data integrity and/or security.
- The specialized requirements of industrial protocols (e.g. real-time, synchronous communication) often make them highly susceptible to disruption.

It should be possible to assess the risks of other industrial network protocols that are not covered here directly by understanding the basic principles of how to secure these protocols.

FIELDBUS PROTOCOLS

MODICON COMMUNICATION BUS

The programmable logic controller dates as far back as 1968 when General Motors set out to find a new technology to replace their hard-wired electromechanical relay system with an electronic device. The first PLC was developed by Bedford Associates and designated 084 (representing the Bedford's eighty-fourth project), and released by the product name Modicon or MOdular DIgital CONtroller.[2] The Modbus protocol was designed in 1979 to enable process controllers to communicate with real-time computers (e.g. MODCOMP FLIC, DEC PDP-11), and remains one of the most popular protocols used in ICS architectures. Modbus has been widely adopted as a de facto standard and has been enhanced over the years into several distinct variants.

Modbus' success stems from its relative ease of use by communicating raw messages without restrictions of authentication or excessive overhead. It is also an open standard, is freely distributed, and is widely supported by members of the Modbus Organization, which still operates today.

What it Does

Modbus is an application layer messaging protocol, meaning that it operates at Layer 7 of the OSI model. It allows for efficient communications between interconnected assets based on a "request/reply" methodology. Extremely simple devices, such as sensors or motors, use Modbus to communicate with more complex computers, which can read measurements and perform analysis and control. To support a communications protocol on a simple device requires that the message generation, transmission, and receipt all require very little processing overhead. This same quality also makes Modbus suitable for use by PLCs and remote terminal units (RTUs) to communicate supervisory data to an ICS system.

Because Modbus is a Layer 7 protocol, it operates independently of underlying network protocols residing at Layer 3, allowing it to be easily adapted to both serial and routable network architectures. This is shown in Figure 6.1.[3]

How it Works

Modbus is a request/response protocol using three distinct protocol data units (PDU): Modbus Request, Modbus Response, and Modbus Exception Response, as illustrated in Figures 6.2 and 6.3.[4]

Modbus can be implemented on either an RS-232C (point-to-point) or RS-485 (multidrop) physical layer. Up to 32 devices could be implemented on a single RS-485 serial link, requiring each device communicating via Modbus be assigned a unique address. A command is addressed to a specific Modbus address, and while other devices may receive the message, only the addressed device will respond. Implementations using RS-232C were relatively simple to commission; however, due to the many variations in the way RS-485 could be implemented (two-wire, four-wire, grounding, etc.), it was sometimes very challenging to commission a multidrop topology when using devices from many different vendors.

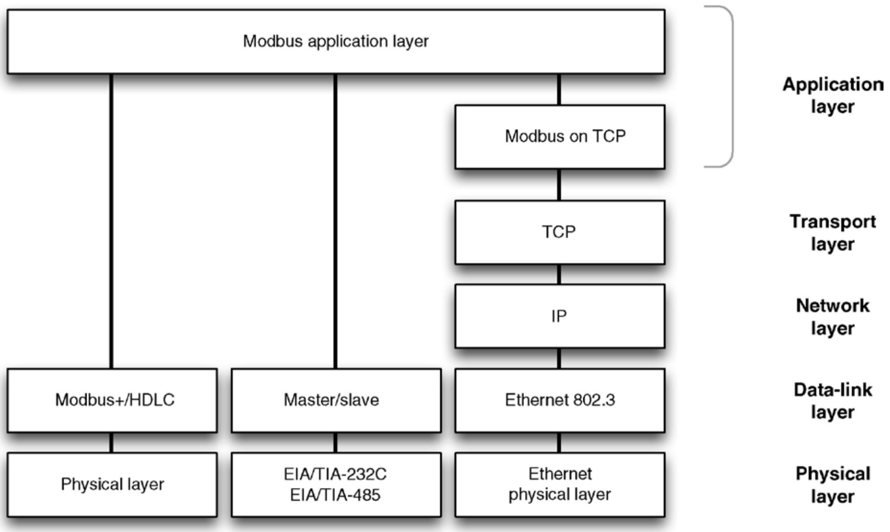


FIGURE 6.1 Modbus alignment with OSI 7-Layer model.

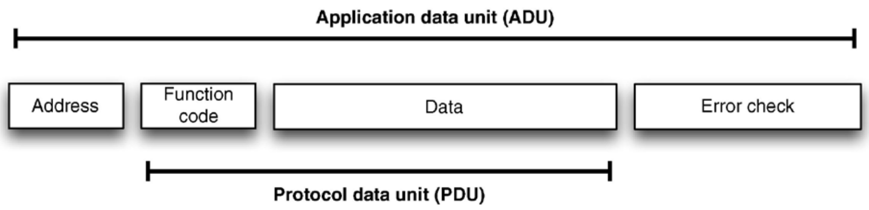


FIGURE 6.2 General Modbus frame.

A "transaction" begins with the transmission of an initial Function Code and a Data Request within a Request PDU. The receiving device responds in one of two ways. If there are no errors, it will respond with a Function Code and Data Response within a Response PDU. If there are errors, the device will respond with an Exception Function Code and Exception Code within a Modbus Exception Response.

Data are represented in Modbus using four primary tables as shown in Table 6.1. The method of handling each of these tables is device specific, as some may offer a single data table for all types, while others offer unique tables. Careful review of the device documentation is needed in order to understand the device's data model, because the original Modbus definitions provided for only addresses in the range 0–9999. The specification has since been appended to allow up to 65,536 addresses across all four data tables. Another caveat within the standard is that the original definition provided for the first digit of the register to identify the data table.

Function Codes used in Modbus are divided into three categories and provide the device vendor with some flexibility in how they implement the protocol within

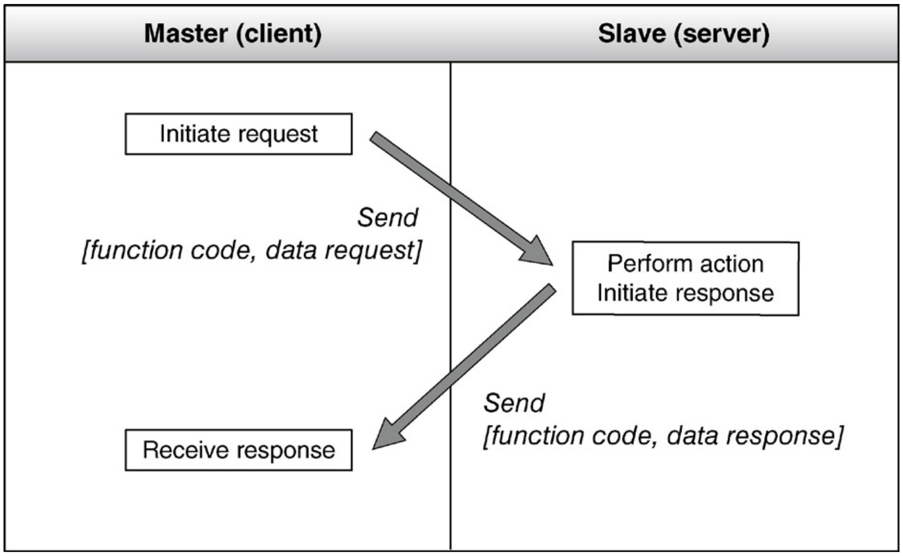


FIGURE 6.3 Modbus protocol transaction (error-free).

Table 6.1 Modbus Data Tables

Data Table	Object Type	Access	Data Provided by	Register Range (0–9999)	Register Range (0–65535)
Discrete input	Single bit	Read-only	Physical I/O	00001–09999	000001–065535
Coil	Single bit	Read-write	Application	10001–19999	100001–165535
Input register	16-bit word	Read-only	Physical I/O	30001–39999	300001–365535
Holding register	16-bit word	Read-only	Read–write	40001–49999	400001–465535

the device. Function codes in the range of 01–64, 73–99, and 111–127 are defined as “Public” and are validated by the Modbus-IDA community and are guaranteed unique. This range is not entirely implemented, allowing codes to be defined in the future. “User-Defined” function codes in the range 65–72 and 100–110 are provided to allow a particular vendor to implement functionality to suit their particular device and application. These codes are not guaranteed to be unique and are not supported by the standard. The final category of codes represents “Reserved” functions that are used by some companies for legacy products, but are not available for general public use. These reserved codes include 8, 9, 10, 13, 14, 41, 42, 90, 91, 125, 126, and 127.

Function Codes and Data Requests can be used to perform a wide range of commands. Some examples of Modbus commands include the following:

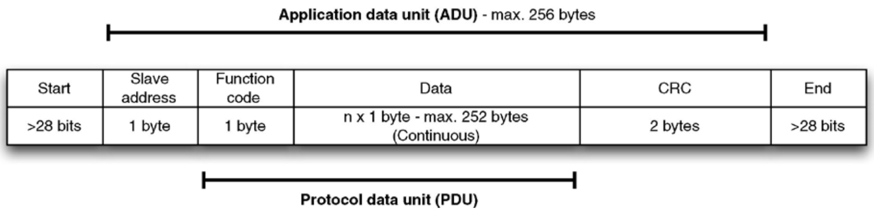
- Read the value of a single register
- Write a value to a single register
- Read a block of values from a group of registers
- Write a block of values to a group of registers
- Read files
- Write files
- Obtain device diagnostic data.

Variants

The popularity of Modbus has led to the development of several variations to suit particular needs. These include **Modbus RTU** and **Modbus ASCII**, which support binary and ASCII transmissions over serial buses, respectively. Modbus TCP is a variant of Modbus developed to operate on modern networks using the IP. **Modbus Plus** is a variant designed to extend the reach of Modbus via interconnected busses using token passing techniques.[5]

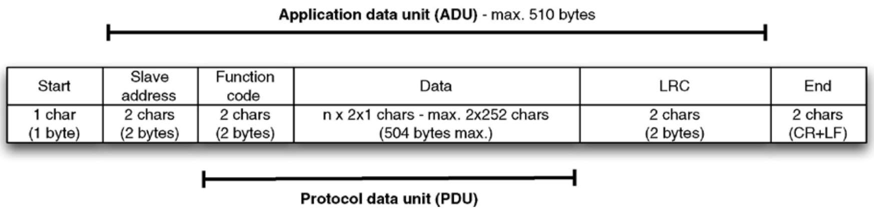
Modbus RTU and Modbus ASCII

These similar variants of Modbus are used in asynchronous serial communications, and they are the simplest of the variants based on the original specification. Modbus RTU (Figure 6.4) uses binary data representation, whereas Modbus ASCII (Figure 6.5) uses ASCII characters to represent data when transmitting over the serial link. Modbus RTU is the more common version and provides a very compact frame over Modbus ASCII. Modbus ASCII represents data as a hexadecimal value coded as



Start	Slave address	Function code	Data	CRC	End
>28 bits	1 byte	1 byte	n x 1 byte - max. 252 bytes (Continuous)	2 bytes	>28 bits

FIGURE 6.4 Modbus frame (Modbus RTU).



Start	Slave address	Function code	Data	LRC	End
1 char (1 byte)	2 chars (2 bytes)	2 chars (2 bytes)	n x 2x1 chars - max. 2x252 chars (504 bytes max.)	2 chars (2 bytes)	2 chars (CR+LF)

FIGURE 6.5 Modbus frame (Modbus ASCII).

ASCII, with two characters required for each byte of data (ASCII PDU is twice the size of RTU PDU). Each uses a simple message format carried within an ADU (see Figure 6.2), consisting of an address, function code, a payload of data, and a checksum, to ensure the message was received correctly.

Modbus TCP

Modbus can also be transported over Ethernet using TCP in two forms. The basic form takes the original Modbus RTU ADU (as shown in Figure 6.4) and applies a Modbus Application Protocol (MBAP) header to create a new frame (Figure 6.6) that is passed down through the remaining layers of the communication stack adding appropriate headers (Figure 6.7) before being placed on the Ethernet network. This new frame includes all of the original error checking and addressing information. This form of protocol is very common with older, legacy devices that contain a Modbus RTU serial interface and are connected to a "device server," which places this information on an industrial network and is received by a similar "device server" converting it back to serial RTU form.

Modbus TCP is the more common form and uses TCP as a transport over IP to issue commands and messages over modern routable networks. Modbus/TCP removes the legacy address and error checking, and places only the Modbus PDU together with a MBAP header into a new frame (see Figure 6.8). The "Unit ID" acts as the new network device address and is part of the MBAP header. Error checking is performed as part of the composite Ethernet frame.

Modbus Plus or Modbus+

Modbus Plus is actually not a variant of the base Modbus protocol, but a different one that utilizes token passing mechanisms to send embedded Modbus messages

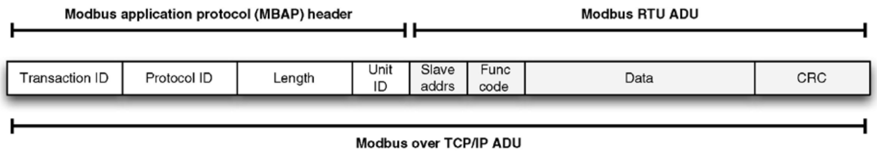


FIGURE 6.6 Modbus frame (Modbus over TCP/IP).

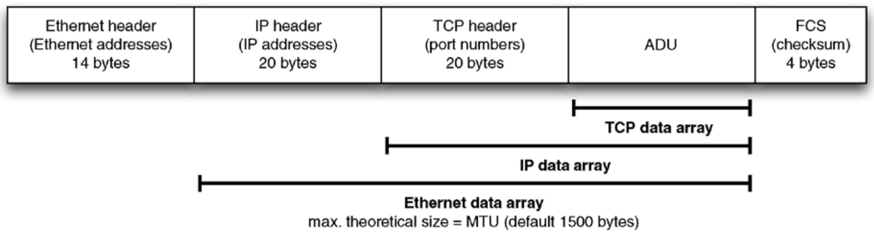

Ethernet header (Ethernet addresses) 14 bytes
IP header (IP addresses) 20 bytes
TCP header (port numbers) 20 bytes
ADU
FCS (checksum) 4 bytes
TCP data array
IP data array
Ethernet data array
max. theoretical size = MTU (default 1500 bytes)

FIGURE 6.7 Modbus ADU with supplemental headers.

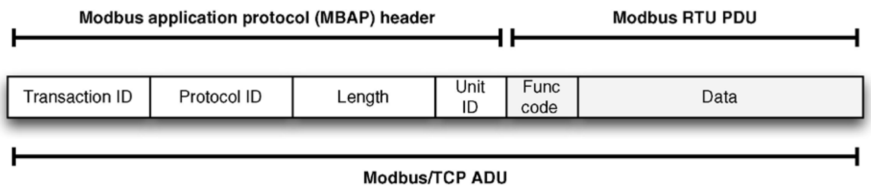


FIGURE 6.8 Modbus frame (Modbus/TCP).

over an RS-485 serial communications link with transmission rates up to 1 Mbps using single- (nonredundant) and dual-cable (redundant) topologies. The network supports the ability to broadcast data to all nodes, and allows "bridges" to be added to a network creating segmented Modbus networks that each can contain up to 64 addressable nodes. This allows for very large Modbus networks to be created. Modbus+ remains a proprietary protocol to Schneider-Electric.[6]

Where it is Used

Modbus is typically deployed between PLCs (slave) and HMIs (master), or between a master PLC and several slave devices, such as PLCs, drives, and sensors, as shown in Figure 6.9. Modbus devices can act as a "master" to some, while acting at the same time as a "slave" to other devices. This function is common in a master terminal unit

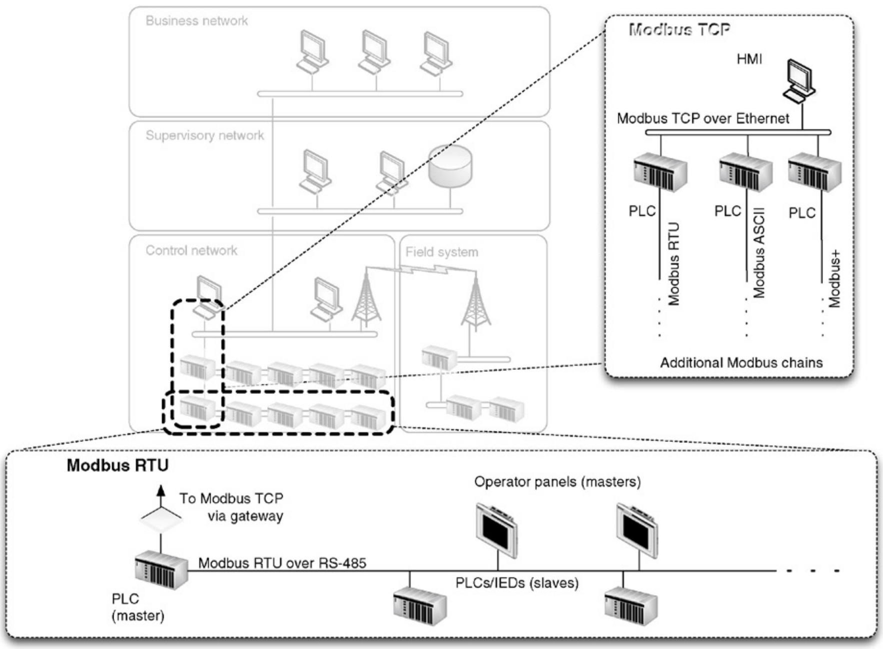


FIGURE 6.9 Typical Modbus use within the industrial network architecture.

(MTU) that is polling data as a master from several slave PLCs and intelligent electronic devices (IEDs), while supporting requests for data as a slave to other master devices like ICS servers and HMIs.

Security Concerns

Modbus represents several security concerns:

- Lack of authentication – Modbus sessions only require the use of a valid Modbus address, function code, and associated data. The data must contain the values of legitimate registers or coils contained in the slave device, or the message will be rejected. This requires additional information of the target in order to provide a valid message; however, this can be obtained from either analysis of network traffic or the configuration of the device. Modbus supports additional function codes that can be used without specific knowledge of the target (e.g. function code 43). There is no verification that the message originated from a legitimate device allowing for simple man-in-the-middle (MitM) and replay style attacks.
- Lack of encryption – Commands and addresses are transmitted in clear text and can therefore be easily captured and spoofed or replayed due to the lack of encryption. Network packet capturing of communications to/from a Modbus device can also disclose significant information pertaining to the configuration and use of the device.
- Lack of message checksum (Modbus/TCP only) – A command can easily be spoofed by building up the Modbus/TCP ADU with the desired parameters, as the checksum is generated at the transmission layer, not the application layer.
- Lack of broadcast suppression (serial Modbus variants only used in a multidrop topology). All serially connected devices will receive all messages, meaning a broadcast of unknown addresses can be used for effective denial of service (DoS) to a chain of serially connected devices.

Security Recommendations

Modbus, like many industrial control protocols, should only be used to communicate between sets of known devices, using expected function codes. In this way it can be easily monitored by establishing clear network zones and by baselining acceptable behavior. This baseline behavior can then be used to establish access controls on the conduit into the zone via appliances that provide protocol inspecting and filtering capabilities (e.g. industrial firewall with deep-packet inspection capabilities). It is also possible at the network level to create fingerprints of normal behavior patterns that facilitate network **whitelists** that can be implemented on in-line and out-of-band devices. For more information about creating whitelists, this topic is discussed in detail in Chapter 11, “Exception, Anomaly and Threat Detection.”

Some specific examples of Modbus messages that should be of concern include the following:

- Modbus TCP packets that are of wrong size or length.
- Function codes that force slave devices into a “listen only” mode.

- Function codes that restart communications.
- Function codes that clear, erase, or reset diagnostic information, such as counters and diagnostic registers.
- Function codes that request information about Modbus servers, PLC configurations, or other device-specific, need-to-know information.
- Traffic on port 502/tcp that is not Modbus or is using Modbus over malformed protocol(s).
- Any message within an Exception PDU (i.e. any Exception Code).
- Modbus traffic from a server to many slaves (i.e. a potential DoS).
- Modbus requests for lists of defined points and their values (i.e. a configuration scan).
- Commands to list all available function codes (i.e. a function scan).

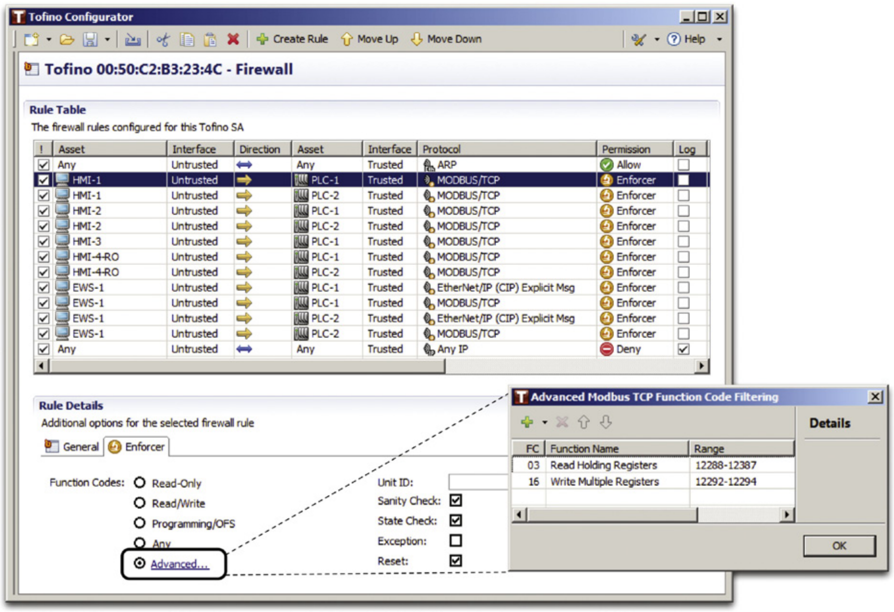
ICS-aware intrusion protection systems can be configured to monitor for these activities using Modbus signatures, such as those developed and distributed by Digital Bond under the QuickDraw project. In more critical areas, an application-aware firewall, industrial protocol filter, or application data monitor may be required to validate Modbus sessions and ensure that Modbus has not been "hijacked" and used for covert communication, command, and control (i.e. the underlying TCP/IP session on port 502/tcp has not been altered to hide additional communications channels within otherwise normal-looking Modbus traffic). This device can also be used to limit function codes communicated into the zone to only those allowed for normal operation. This is discussed in detail in Chapter 9, "Establishing Zones and Conduits." Figure 6.10 illustrates configuration of an application-layer firewall on the conduit into a plant zone separating four HMIs, one EWS and two PLCs using both Modbus/TCP and EtherNet/IP protocols (Figure 6.1).

DISTRIBUTED NETWORK PROTOCOL

The Distributed Network Protocol began as a serial protocol much like Modbus designed for use between "master stations" or "control stations" and slave devices called "outstations. It is also commonly used to connect RTUs configured as "master stations" to IED "outstations" in electric substations. The ICCP discussed later in this chapter is commonly used for communication between master stations. DNP3 was initially introduced in 1990 by Westronic (now GE-Harris Canada) and was based on

CAUTION

Intrusion Prevention Systems are able to actively block suspect traffic by dropping packets or resetting TCP connections. However, Intrusion Prevention Systems deployed on industrial networks should only be configured to block traffic after careful consideration and tuning. Unless you are confident that a given signature will not inadvertently block a legitimate control command, the signature should be set to alert, rather than block (i.e. operate in "detection" mode rather than active "prevention" mode).



Tofino 00:50:C2:B3:23:4C - Firewall

Rule Table

The firewall rules configured for this Tofino SA

!	Asset	Interface	Direction	Asset	Interface	Protocol	Permission	Log
✓	Any	Untrusted	↔	Any	Trusted	ARP	Allow	☐
✓	HMI-1	Untrusted	→	PLC-1	Trusted	MODBUS/TCP	Enforcer	☐
✓	HMI-1	Untrusted	→	PLC-2	Trusted	MODBUS/TCP	Enforcer	☐
✓	HMI-2	Untrusted	→	PLC-1	Trusted	MODBUS/TCP	Enforcer	☐
✓	HMI-2	Untrusted	→	PLC-2	Trusted	MODBUS/TCP	Enforcer	☐
✓	HMI-3	Untrusted	→	PLC-1	Trusted	MODBUS/TCP	Enforcer	☐
✓	HMI-4RO	Untrusted	→	PLC-1	Trusted	MODBUS/TCP	Enforcer	☐
✓	HMI-4RO	Untrusted	→	PLC-2	Trusted	MODBUS/TCP	Enforcer	☐
✓	EWS-1	Untrusted	→	PLC-1	Trusted	EtherNet/IP (CIP) Explicit Msg	Enforcer	☐
✓	EWS-1	Untrusted	→	PLC-1	Trusted	MODBUS/TCP	Enforcer	☐
✓	EWS-1	Untrusted	→	PLC-2	Trusted	EtherNet/IP (CIP) Explicit Msg	Enforcer	☐
✓	EWS-1	Untrusted	→	PLC-2	Trusted	MODBUS/TCP	Enforcer	☐
✓	Any	Untrusted	↔	Any	Trusted	Any IP	Deny	✓

Rule Details

Additional options for the selected firewall rule

General | Enforcer

Function Codes: ○ Read-Only ○ Read/Write ○ Programming/OFS ○ Any [✓ Advanced...]

Unit ID:
Sanity Check: ✓
State Check: ✓
Exception: ☐
Reset: ✓

FC	Function Name	Range
03	Read Holding Registers	12288-12387
16	Write Multiple Registers	12292-12294

FIGURE 6.10 Application-layer firewall - Modbus/TCP zone protection

(image courtesy of Tofino Security - A Belden Brand).

early drafts of the IEC 60870-5 standard. The primary motivation for this protocol was to provide reliable communications in environments common within the electric utility industry that include high level of electromagnetic interference (EFI) and poor transmissions media (at that time based on analog telephone lines). DNP3 was extended to work over IP via encapsulation in TCP or UDP packets in 1998, and is now widely used in not only electric utility, but also oil and gas[7], water, and wastewater industries. One of the leading reasons for some industry migration from Modbus to DNP3 includes features that apply to these other industries including report by exception, data quality indicators, time-stamped data including sequence-of-events, and a two-pass "select before operate" procedure on outputs.[8] Other markets, including Europe, have adopted the IEC 60870-5 version of the protocol as it was ratified. Though DNP3 was based on IEC 60870-5, differences do exist between the two.

One distinction of DNP3 is that it is very reliable, while remaining efficient and well suited for real-time data transfer. It also utilizes several standardized data formats and supports time-stamped (and time-synchronized) data, making real-time transmissions more efficient and thus even more reliable. Another reason that DNP3 is considered highly reliable is due to the frequent use of cyclical redundancy checks (CRC)—a single DNP3 frame can include up to 17 CRCs: one in the header and one per data block within the payload (see the section "How it Works"). There are also optional link-layer acknowledgments for further reliability assurance, and—of particular note—variations of DNP3 that support link-layer authentication as well. Because all of this is done within the link-layer frame, it means that additional network-layer checks may also apply if DNP3 is encapsulated for transport over Ethernet.

Unlike Modbus and ICCP, DNP3 is bidirectional (supporting communications from both Master to Slave and from Slave to Master) and it supports exception-based reporting. It is therefore possible for a DNP3 outstation to initiate an unsolicited response, in order to notify the master station of an event outside of the normal polling interval (such as an alarm condition).

What it Does

Like the other industrial protocols, DNP3 is primarily used to send and receive messages between control system devices—only in the case of DNP3, it also does it with a high degree of reliability. Assuming that the various CRCs are all valid, the data payload is then processed. The payload is very flexible and can be used to simply transfer informational readings. It can also be used to send control functions, or even direct binary or analog data for direct interaction with devices, such as RTUs and IEDs.

Both the link-layer frame (or LPDU) header and the data payload contain CRCs, and the data payload actually contains a pair of CRC octets for every 16 data octets. This provides a high degree of assurance that any communication errors will be detected. DNP3 will retransmit the faulty frames if any errors are detected. There are also physical layer integrity issues in addition to frame integrity. However, it still remains possible that a correctly formed and transmitted frame will not arrive at its destination. DNP3 uses an additional link layer confirmation to overcome this risk. When link layer confirmation is enabled, the DNP3 transmitter (source) of the frame requests that the receiver (destination) confirms the successful receipt of the frame. If a requested confirmation is not received, the link layer will retransmit the frame. This confirmation is optional because although it increases reliability, it adds overhead that directly impacts the efficiency of the protocol. In real-time environments, this added overhead might not be appropriate.[9]

Once a successful and (if requested) confirmed frame arrives, the frame is processed. Each frame consists of a multipart header and a data payload. The header is significant as it contains a well-defined function code, which can tell the recipient whether it should confirm, read, write, select a specific point, operate a point (initiate a change to a point), directly operate a point (both selecting and changing a point in one command), or directly operate a point without acknowledgment.[10]

These functions are especially powerful when considering that the data payload of the DNP3 frame supports analog data, binary data, files, counters, and other types of data objects. At a high level, DNP3 supports two kinds of data, referred to as class 0 or static data (data that represents a static value) and event data (data that represents a change such as an alarm condition). Event data are rated by priority from class 1 (highest) to class 3 (lowest). The differentiation of static and event data, as well as the classification of event data, allows DNP3 to operate more efficiently by allowing higher-priority information to be polled more frequently, for example, or to enable or disable unsolicited responses by data type. The data itself can be binary, analog input or output, or a specific control output.[11]

How it Works

DNP3 provides a method to identify the remote device's parameters and then use message buffers corresponding to event data classes 1 through 3 in order to identify incoming messages and compare them to known point data. In this way, the master station is only required to retrieve new information resulting from a point change or change event on the outstation.

Initial communications are typically a class 0 request from the master station to an outstation, used to read all point values into the master station's database. Subsequent communications will typically either be direct poll requests for a specific data class from the master station; unsolicited responses for a specific data class from an outstation; control or configuration requests from the master station to an outstation, or subsequent periodic class 0 polls. When a change occurs on an outstation, a flag is set to the appropriate data class. The master station is then able to poll only those outstations where there is new information to be reported.

This is a major departure from constant data polling that directly results in improved responsiveness and more efficient data exchange. The departure from a real-time polling mechanism does require time synchronization, because the time between a change event and a successful poll/request sequence is variable. This means that all responses are time-stamped so that the events between polls can be reconstructed in the correct order.

Communication is initiated by the master station to the outstation, or in the case of unsolicited responses (alarms) from the outstation to the master station, as shown in Figure 6.11. Because DNP3 operates bidirectionally and supports unsolicited responses, as shown in Figure 6.12, each frame requires both a source address and a destination address so that the recipient device knows which messages to process, and which device to return responses to. The addition of a source address does add some overhead. Remember that with purely master/slave protocols, there is no need for a source address as the originating device is always the master. This overhead provides a return benefit of dramatically increased scalability and functionality. As many as 65,520 individual device addresses are available within DNP3, and any one of them can initiate communications. An address equals one device (every DNP3 device requires a unique address), although there are reserved DNP3 addresses, including one for broadcast messages (which will be received and processed by all connected DNP3 devices).[12]

Secure DNP3

Secure DNP3 is a DNP3 variant that adds authentication to the response/request process, as shown in Figure 6.13. Authentication is issued as a challenge by the receiving device. A challenge condition occurs upon session initiation (when a master station initiates a DNP3 session with an outstation), after a preset period of time (the

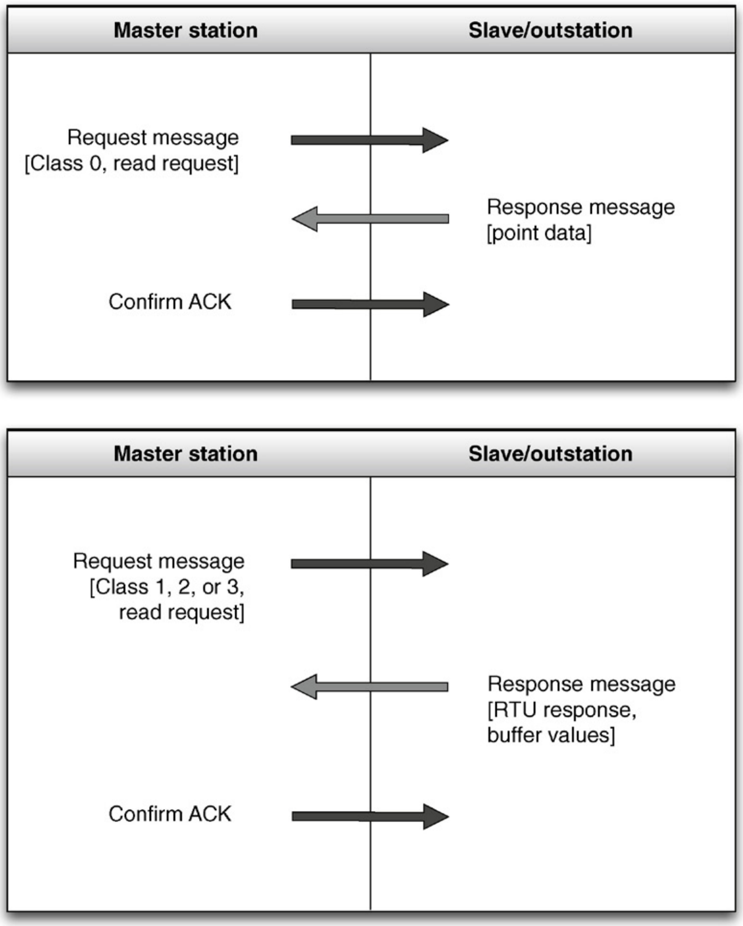

Master station
Slave/outstation
Request message [Class 0, read request]
Response message [point data]
Confirm ACK
Master station
Slave/outstation
Request message [Class 1, 2, or 3, read request]
Response message [RTU response, buffer values]
Confirm ACK

FIGURE 6.11 DNP3 protocol operation.

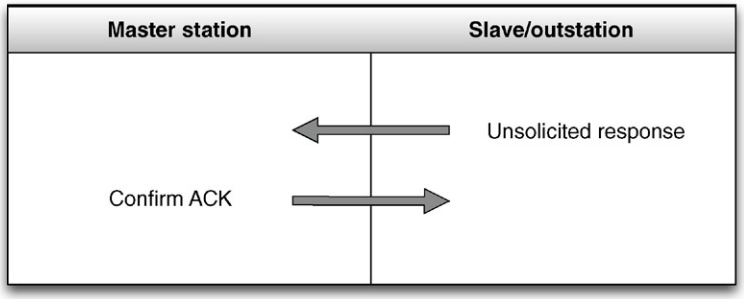


FIGURE 6.12 DNP3 protocol operation: Unsolicited responses allow remote alarm generation.

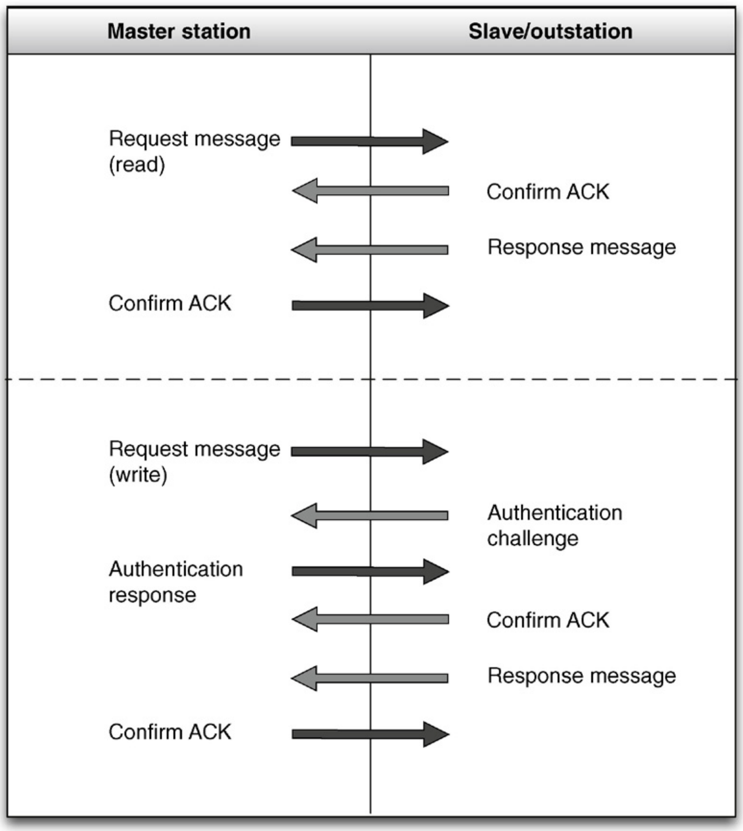


FIGURE 6.13 Message confirmation and secure DNP3 authentication operation.

default is 20 min), or upon a "critical" request, such as writes, selects, operates, direct operates, starts, stops, and restarts. It is possible to know which requests are critical because the data types and functions of DNP3 are well defined.[13]

Authentication occurs using a unique session key that is hashed together with message data from the sender and from the challenger. The result is an authentication method that verifies authority (checksum against the secret key), integrity (checksum against the sending payload), and pairing (checksum against the challenge message) at the same time. In this way, it is very difficult to perform data manipulation or code injection, or to spoof or otherwise hijack the protocol.[14]

The DNP3 Layer 2 frame provides the source, destination, control, and payload, and can operate over a variety of application layers including TCP and UDP transports over IP (defaults include 19999/tcp when using Transport Layer Security (TLS) for confidentiality and 20000/tcp or 20000/udp when using application-layer only secure authentication). The function codes are resident within the Control bytes in the DNP3 frame header, as shown in Figure 6.14.

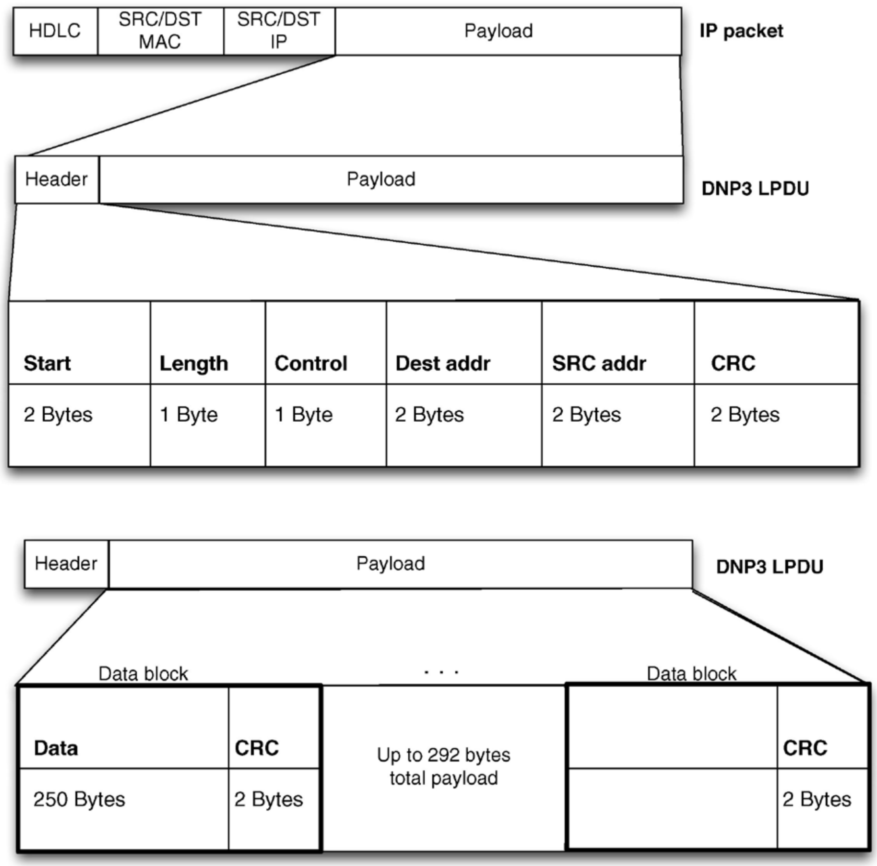


FIGURE 6.14 DNP3 protocol framing.

Where it is Used

DNP3 is primarily used between a master control station and an RTU in a remote station as shown in Figure 6.15. Transmission medium can include wireless, radio, and dial-up. DNP3 is also widely used to interconnect RTUs and IEDs. It can be applied in many applications like the Modbus protocols throughout a typical ICS architecture. Unlike Modbus, however, DNP3 is well suited for hierarchical and aggregated point-to-multipoint topologies in addition to the linear point-to-point and serial point-to-multipoint topologies that are supported by Modbus.[15]

Security Concerns

While much attention is given to the integrity of the data frame, there is no authentication or encryption inherent within DNP3 (although there is within Secure DNP3). It then becomes relatively easy to manipulate a DNP3 session because of the

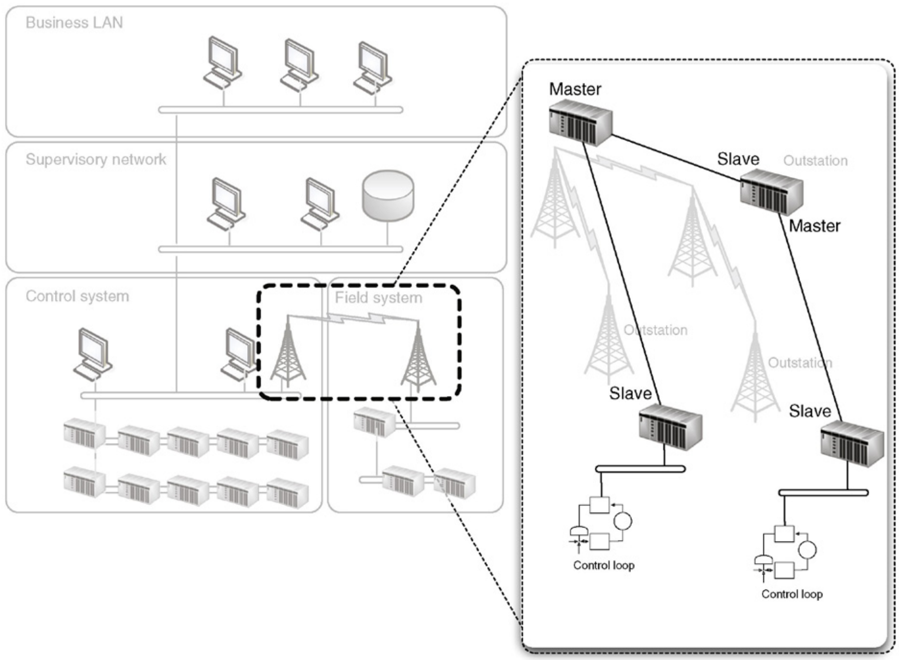


FIGURE 6.15 Typical DNP3 use within the industrial network architecture.

well-defined nature of DNP3 function codes and data types in much the same way as it was the Modbus protocol.

DNP3 does include security measures; however, this added complexity of the protocol increases the chances of vulnerabilities. As of this writing, there are several known vulnerabilities with DNP3 that have been reported by ICS-CERT. Proper system hardening, regular security assessments, and patching of DNP3 interconnections (both master stations and outstations) is recommended because there are known exploits in the wild and DNP3 is a heavily deployed protocol within certain industry segments.

Some examples of realistic hacks against DNP3 include the use of MitM attacks to capture addresses, which can then be used to manipulate other system components. Examples of such manipulation include

- Turning off unsolicited reporting to suppress alarms.[16]
- Spoofing unsolicited responses to the master station to falsify events and trick an operator into taking inappropriate actions.
- Performing a DoS attack through the injection of broadcasts, creating storm behavior within the full extent of the DNP3 system.
- Manipulating the time synchronization data, resulting in synchronization loss and subsequent communication errors.

- Manipulating or eliminating confirmation messages forcing a state of continuous retransmission.
- Issuing unauthorized stops, restarts, or other functions that could disrupt operations.

Security Recommendations

Because a secure implementation of DNP3 is available, the primary recommendation is to implement only Secure DNP3. This can pose problems with legacy installations due to backward compatibility, as Version 5 of the standard (adopted as IEEE-1815-2012) is not backward compatible, and Version 2 (adopted as IEEE-1815-2010) is now deprecated and should be upgraded. It may not always be possible to implement Secure DNP3 due to varying vendor support and other factors. Secure use of the transport layer protocol is advised in these cases, such as the use of TLS. In other words, treat your encapsulated DNP3 traffic as highly sensitive information and use every TCP/IP security best practice to protect it.

DNP3 master stations and outstations should always be isolated into a unique zone consisting only of authorized devices (multiple zones can be defined for devices communicating to multiple clients, or for hierarchical master/slave pairs), and the zone(s) should be thoroughly secured using standard defense-in-depth practices, including an industrial firewall and/or intrusion protection system that enforces strict control over the type, source, and destination of traffic over the DNP3 link across conduits between zones. Preference should be given to security practices that are capable of deep-packet inspection of DNP3 traffic. Many of the recommendations described for Modbus are equally applicable for DNP3, including the creation of network baselines and deployment of network whitelists.

Many threats can be detected through monitoring of DNP3 sessions, and looking for specific function codes and behaviors, including the following:

- Use of any non-DNP3 communication on a DNP3 Port (19999/tcp, 20000/tcp, 20000/udp).
- Use of configuration function code 23 (Disable Unsolicited Responses).
- Use of control function codes 4, 5, or 6 (Operate, Direct Operate, and Direct Operate without Acknowledgment).
- Use of application control function 18 (Stop Application).
- Multiple, unsolicited responses over time (Response Storm).
- Any unauthorized attempt to perform an action requiring authentication.
- Any authentication failures.
- Any DNP3 communication sourced from or destined to a device that is not explicitly identified as a DNP3 master station or outstation device.

As with other industrial protocols, ICS-aware intrusion protection systems can be configured to monitor for these activities using DNP3 signatures, such as those developed and distributed by Digital Bond under the QuickDraw SCADA IDS project. An application-aware firewall or application data monitor may be required to validate DNP3 sessions.

CAUTION

Intrusion Prevention Systems are able to actively block suspect traffic by dropping packets or resetting TCP connections. However, Intrusion Prevention Systems deployed on industrial networks should only be configured to block traffic after careful consideration and tuning. Unless you are confident that a given signature will not inadvertently block a legitimate control command, the signature should be set to alert, rather than block (i.e. operate in "detection" mode rather than active "prevention" mode).

PROCESS FIELDBUS

PROFIBUS (PROcess FIeldBUS) is a fieldbus protocol that was originally developed in the late 1980s in Germany by a group of 21 companies and institutions known as the Central Association for the Electrical Industry (ZVEI). ZVEI published their first protocol specification known as PROFIBUS FMS (Fieldbus Message Specification) designed primarily to allow PLCs to communicate with host computers. This protocol was found to be too complex to implement in process control applications, so in 1993 the PROFIBUS DP (Decentralized Periphery) specification was released providing easier configuration and faster messaging. In 1989, the PROFIBUS User Organization (PROFIBUS Nutzer-organisation or PNO) was established to maintain the specifications, ensure device compliance, and certification. A larger user community was established in 1995 called PROFIBUS International (PI) to continue the advancement of PROFIBUS on a global level.

Several specialized variants of PROFIBUS exist, including PROFIBUS PA (for instrumentation used for process automation), PROFIsafe (for safety applications), and PROFIdrive (for high-speed drive applications). The most widely deployed variant is PROFIBUS DP, which itself has three variants: PROFIBUS DP-V0, DP-V1, and DP-V2, each of which represents a minor evolution of capabilities within the protocol. There are also three profiles for PROFIBUS communication: asynchronous, synchronous, and via Ethernet using ethertype 0x8892. PROFIBUS over Ethernet is also called PROFINET[17] and will be discussed separately as part of a category of protocols referred to as "Industrial Ethernet)

PROFIBUS is a master–slave protocol that supports multiple master nodes through the use of token sharing—when a master has control of the token, it can communicate with its slaves (each slave is configured to respond to a single master). Figure 6.16 illustrates how this token-based, master–slave topology operates. In PROFIBUS DP-V2, slaves can initiate communications to the master or to other slaves under certain conditions. A master PROFIBUS node is typically a PLC or RTU, and a slave is a sensor, motor, or some other control system device.

PROFIBUS DP supports several different physical layer deployments with RS-485 as the most common. The existing RS-485 specification was extended to allow PROFIBUS to operate at speeds up to 12 Mbps using two wires. The Process

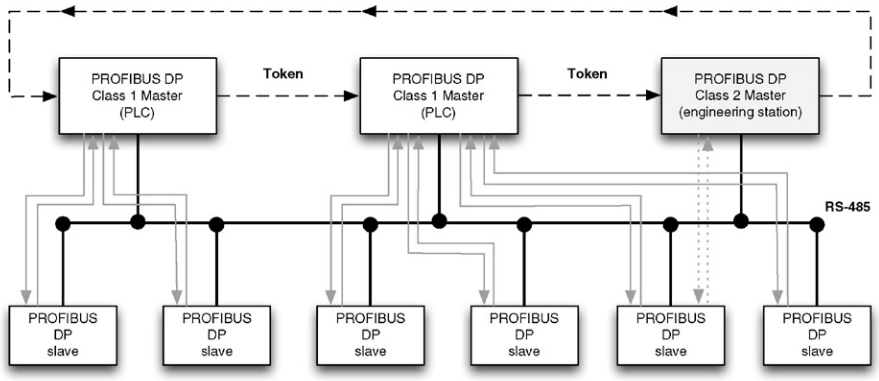


FIGURE 6.16 PROFIBUS DP communications.

Automation (PA) specification was developed to address the unique needs of field instrumentation in a manner similar to FOUNDATION Fieldbus. These installations must support wiring and communication with devices that are commonly installed in hazardous areas where explosive vapors and dusts are common. A concept known as "intrinsic safety" is used to limit the amount of available power on these communication lines to levels below that necessary to ignite the dust or vapor. The Manchester-encoded, bus-powered, intrinsically safe (MBP-IS) physical layer is used in these cases to address this requirement providing both limited levels of device power and communication on a single pair of wires.

Security Concerns

PROFIBUS lacks authentication inherent to many of its functions, allowing a spoofed node to impersonate a master node, which in turn provides control over all configured slaves. A compromised master node or a spoofed master node could also be used to capture the token, inject false tokens, or otherwise disrupt the protocol functions, causing a DoS. A rogue master node could alter clock synchronization to slave devices, snoop query responses (across all masters), or even inject code into a slave node. It is important to remember that PROFIBUS DP utilizes a serial connection between the master and slave devices, so the security concerns mentioned require physical access to connect to the DP network. This means that a DP network is not generally susceptible to industrial network-based attacks. However, the master device is typically connected to an Ethernet network and is therefore no less susceptible to attack from authorized network access than any other Ethernet-connected device. PROFIBUS over Ethernet (PROFINET) is a real-time Ethernet protocol, and as such it is susceptible to any of the vulnerabilities of Ethernet. When used over the IP, it is also susceptible to any vulnerabilities of IP.

NOTE

Stuxnet (see Chapter 3, "Industrial Cyber Security, History and Trends") is an example of PROFIBUS exploitation. Stuxnet compromised PLCs (PROFINET devices acting as PROFIBUS DP master nodes) via an initial network attack on an engineering workstation or HMI. It then monitored the PROFIBUS DP network and looked for specific behaviors associated with frequency controllers (PROFIBUS DP slave nodes). Once the sought-after conditions were detected, Stuxnet then issued commands to the relevant slave nodes to sabotage the mechanical equipment (centrifuges used to enrich Uranium) by altering their operating parameters (speed of the centrifuges).

Security Recommendations

PROFIBUS DP is a naturally segmented serial network utilizing a topology that is generally contained within a small geographical area, such as a section of a plant or manufacturing process. The network and connected devices are very susceptible to attack if unauthorized physical access is obtained. For the purposes of this book, physical security must always be provided, since the threat events that can be performed via local access are relatively easy and can provide significant disruption to the operation of the ICS. This is outside the scope of this book.

INDUSTRIAL ETHERNET PROTOCOLS

Industrial Ethernet is a term used to reference the adaptation of the IEEE 802.3 Ethernet standard to real-time industrial automation applications. One of the primary objectives of these extensions is the move toward more "synchronous" mechanisms of communication in order to prevent data collisions and minimize jitter inherent with "asynchronous" communications like standard Ethernet. This will allow the technology to be deployed in critical time-dependent applications like safety and industrial motion control. This concept may seem abstract in a time when 1Gbps switched networks are readily available; however, as one moves into the industrial sector, the applications must be applicable to not only "lightweight" and simple devices that may not have the capacity for these modern IT networks, but also the deployment of network topologies on the factory floor that can be more suited for bused or trunked style topologies (e.g. automobile networks).

Industrial Ethernet also provides physical enhancements to "harden" the office-grade nature of standard Ethernet technologies with ruggedized wiring, connectors, and hardware designed to meet the environment of industrial applications. Conditions that are addressed with Industrial Ethernet include electrical noise and interference (EMI), vibration, extended temperatures and humidity (high and low), power requirements, and extensions to support real-time performance (low latency, low jitter, minimal packet loss).[18]

There are some 30 different varieties of Industrial Ethernet[19]; however, for the purposes of this book, attention will be given to five as they are not only widely accepted and deployed in industry global (e.g. market leaders), but they introduce new concepts and concerns regarding industrial network security. These include EtherNet/IP, PROFINET, EtherCAT, Ethernet POWERLINK, and SERCOS III. Studies conducted by IMS and ARC show that approximately 75% of all Ethernet

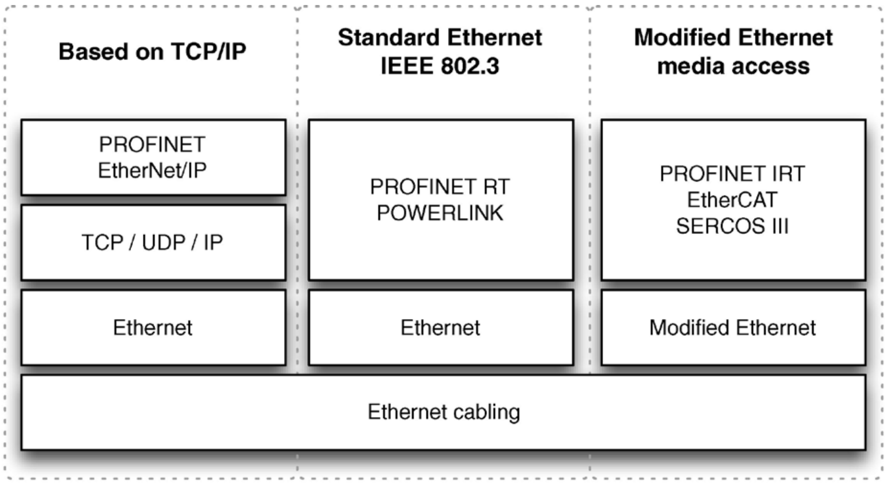


FIGURE 6.17 Methods for real-time Ethernet implementation.

installation in industrial environments use EtherNet/IP, PROFINET or Modbus/TCP (already discussed), with the next two leading technologies based on POWERLINK and EtherCAT[20]. Figure 6.17 provides an illustration of how these various technologies compare.

ETHERNET INDUSTRIAL PROTOCOL

It is important to understand the CIP in order to appreciate its versatility and application to the EtherNet/IP implementation. CIP, originally known as "Control and Information Protocol," is a publicly available protocol managed through the Open Device Vendors Association (ODVA). CIP is an application layer protocol that provides a consistent set of messages and services that can be implemented in a variety of ways using different network and link layer techniques, all supporting interoperability. These variations include EtherNet/IP (CIP on Ethernet), DeviceNet (CIP on CAN), CompoNet, and ControlNet (CIP on CTDMA) with extensions that include safety (CIP Safety), motion control (CIP Motion), and synchronization (CIP sync). Figure 6.18 illustrates the deployment model for CIP against the OSI layers.[21]

NOTE

The Controller Area Network (CAN) is a bus developed in 1985 by Bosch and adopted as international standard ISO 11898 in 1993 originally used for vehicle networks. It is a low-cost network utilizing a trunk-drop technology while supplying power and signal to interconnect simple devices.

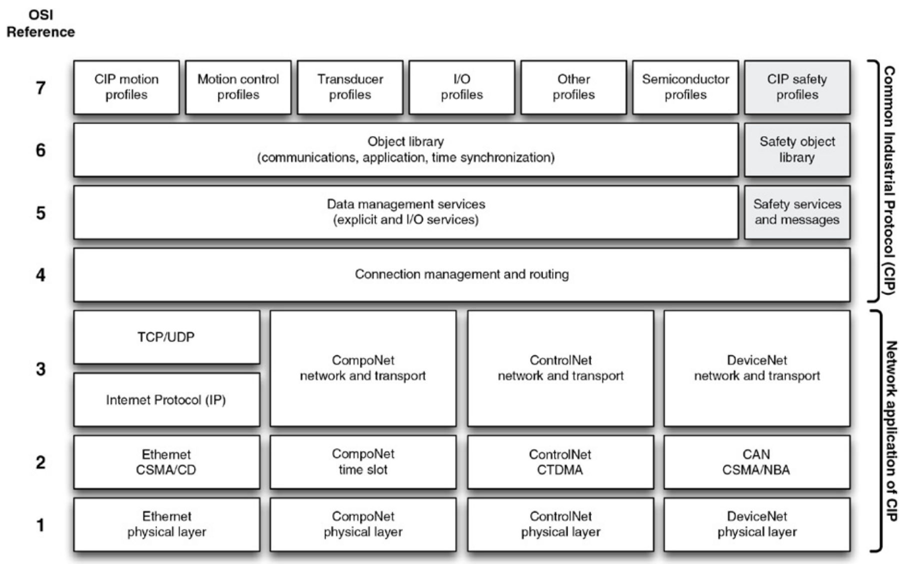

OSI Reference
7
6
5
4
3
2
1
CIP motion profiles
Motion control profiles
Transducer profiles
I/O profiles
Other profiles
Semiconductor profiles
CIP safety profiles
Object library (communications, application, time synchronization)
Safety object library
Data management services (explicit and I/O services)
Safety services and messages
Connection management and routing
TCP/UDP
Internet Protocol (IP)
CompoNet network and transport
ControlNet network and transport
DeviceNet network and transport
Ethernet CSMA/CD
CompoNet time slot
ControlNet CTDMA
CAN CSMA/NBA
Ethernet physical layer
CompoNet physical layer
ControlNet physical layer
DeviceNet physical layer
Common Industrial Protocol (CIP)
Network application of CIP

FIGURE 6.18 Overview of Common Industrial Protocol.[54]

NOTE

Concurrent Time Domain Multiple Access (CTDMA) provides the enhancements over traditional Carrier Sense Multiple Access/Collision Domain (CSMA/CD) found in Ethernet to support deterministic, high-speed communication of time-critical I/O and control data. The design allows for all addresses to have access to the network through the implementation of a time slice algorithm that provides both "scheduled" and "unscheduled" data transfers.

EtherNet/IP (EIP) or CIP on Ethernet uses standard Ethernet frames (ethertype 0x80E1) in conjunction with the CIP suite to communicate with nodes. As with all CIP implementations, EIP supports integration of I/O, control, data collection, and device configuration on a single network. For real-time I/O and control related data, EIP utilizes a connectionless multicast UDP transport called "implicit messaging" using port 2222/udp. This mechanism optimizes performance by establishing a "producer–consumer" relationship between devices sending data and those devices requiring the data—a common communications model within ICS architectures. A unicast TCP transport is also available to transmit larger quantities of data commonly associated with device configuration, diagnostics, and event information using an "explicit messaging" service commonly found on port 44818/tcp.

NOTE

The "IP" in EtherNet/IP derives from "Industrial Protocol" and not "Internet Protocol," because of the use of the Common Industrial Protocol. Similarly, the acronym "CIP" meaning "Common Industrial Protocol" should not be confused with "Critical Infrastructure Protection" of NERC CIP.

Common Industrial Protocol uses object models to define the various qualities of a device. Each CIP object possesses attributes (data), services (commands), connections, and behaviors (relationships between attribute values and services). There are three types of objects:

- Required Objects define attributes, such as device identifiers (e.g. manufacturer, serial number, date of manufacture) (Identity Object), routing identifiers for object-to-object messaging (Message Router Object), and physical connection data (Network Object).
- Application Objects define input and output profiles for devices.
- Vendor-specific Objects enable vendors to add proprietary objects to a device.

Objects (other than vendor-specific objects) are standardized by device type and function, to facilitate interoperability. If one brand of pump is exchanged for another brand, for example, the Application Objects will remain compatible, eliminating the need to build custom drivers. The wide adoption and standardization of CIP has resulted in an extensive library of device models, which can facilitate interoperability but can also aid in control network scanning and enumeration (see Chapter 8, "Risk and Vulnerability Assessments").

While the Required Objects provide a common and complete set of identifying values, the Application Objects contain a common and complete suite of services for control, configuration, and data collection that includes both implicit (control) and explicit (information) messaging.[22]

Security Concerns

EtherNet/IP is a real-time Ethernet protocol, and as such it is susceptible to any of the vulnerabilities of Ethernet. EIP implicit messaging over UDP is transaction-less and so there is no inherent network-layer mechanism for reliability, ordering, or data integrity checks. CIP also introduces some specific security concerns, due to its well-defined object model.

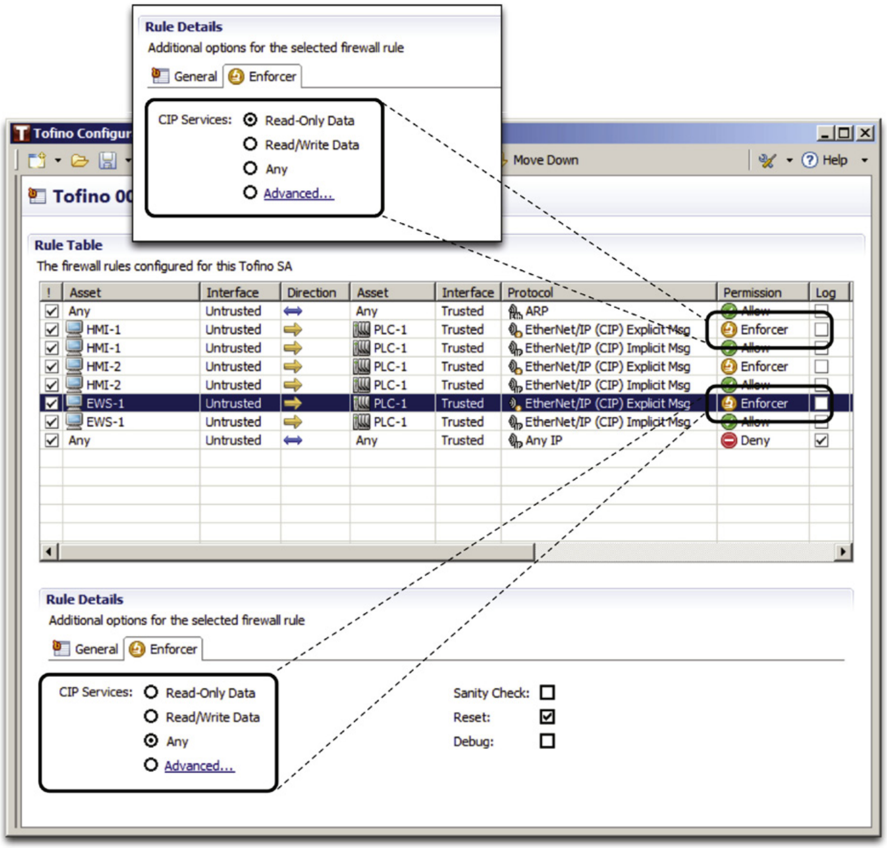
The following concerns are specific to EtherNet/IP:

- The CIP does not define any explicit or implicit mechanisms for security.
- The use of common Required Objects for device identification can facilitate device identification and enumeration, facilitating a targeted attack.
- The use of common Application Objects for device information exchange and control can enable broader industrial attacks, able to manipulate a broad range of industrial devices.
- EtherNet/IP's use of UDP and Multicast traffic—both of which lack transmission control—for real-time transmissions facilitate the injection of spoofed traffic or (in the case of multicast traffic) the manipulation of the transmission path using injected IGMP controls.

Security Recommendations

EtherNet/IP is a real-time Ethernet protocol using TCP and UDP transports making it necessary to provide Ethernet- and IP-based security at the perimeter of any EIP network. Consideration should be given to placing EIP devices in dedicated zones that include either an application-layer appliance capable of performing inspection in EIP packets and only allowing required functions within the zone. A stateful, packet-filtering firewall can be used to limit unnecessary inbound traffic (such as device configuration) to the zone. Figure 6.19 illustrates the configuration of an application-layer firewall on the conduit into an EIP zone separating four HMIs, one EWS, and two PLCs.

It is also recommended that passive network monitoring be used to ensure the integrity of the EIP network, ensuring that the EIP protocol is only being used by explicitly identified devices, and that no EIP traffic is originating from an unauthorized, outside source. This can be accomplished using an ICS-aware intrusion prevention system or other network monitoring device capable of detecting and interpreting the EIP. Additional guidance can be obtained through ODVA.[23]



Rule Details
Additional options for the selected firewall rule
General | Enforcer

CIP Services: [✓ Read-Only Data] Read/Write Data Any Advanced...

Rule Table
The firewall rules configured for this Tofino SA

!	Asset	Interface	Direction	Asset	Interface	Protocol	Permission	Log
✓	Any	Untrusted	⇔	Any	Trusted	ARP	Allow	☐
✓	HMI-1	Untrusted	⇒	PLC-1	Trusted	EtherNet/IP (CIP) Explicit Msg	Enforcer	☐
✓	HMI-1	Untrusted	⇒	PLC-1	Trusted	EtherNet/IP (CIP) Implicit Msg	Allow	☐
✓	HMI-2	Untrusted	⇒	PLC-1	Trusted	EtherNet/IP (CIP) Explicit Msg	Enforcer	☐
✓	HMI-2	Untrusted	⇒	PLC-1	Trusted	EtherNet/IP (CIP) Implicit Msg	Allow	☐
✓	EWS-1	Untrusted	⇒	PLC-1	Trusted	EtherNet/IP (CIP) Explicit Msg	Enforcer	☐
✓	EWS-1	Untrusted	⇒	PLC-1	Trusted	EtherNet/IP (CIP) Implicit Msg	Allow	☐
✓	Any	Untrusted	⇔	Any	Trusted	Any IP	Deny	✓

Rule Details
Additional options for the selected firewall rule
General | Enforcer

CIP Services: Read-Only Data Read/Write Data [✓ Any] Advanced...

Sanity Check: ☐
Reset: ☑
Debug: ☐

FIGURE 6.19 Application-layer firewall - EtherNet/IP zone protection.

(image courtesy of Tofino Security - A Belden Brand).

PROFINET

PROFINET is an open standard Industrial Ethernet developed by the PROFIBUS User Organization (PNO) and Siemens, and is included as part of the IEC 61158 and IEC 61784 international standards for fieldbus communications. PROFINET was designed for scalability, and can be deployed at varying degrees of determinism and network performance. The first version of PROFINET utilized standard Ethernet and TCP/IP packets without modification for non–real-time automation applications and general integration. The software-based Real-Time (RT) technology included in Version 2 added support for time-critical communications with cycle times of 5–10 ms incorporating an optimized protocol stack bypassing OSI layers 3 and 4, limiting communications to a single broadcast domain with no routing capability. PROFIBUS Isochronous Real Time (IRT) was introduced in Version 3 of the standard, and provides cycle times of less than 1 ms with jitter less than 1 μs common in high-speed motion control applications. PROFIBUS IRT is a hardware-based solution that incorporates extensions to the Ethernet stack (OSI Layer 2) requiring special application-specific integrated circuits (ASICs) at the device level and IRT-compatible network switches designed to minimize jitter. IRT is a Layer 2 technology, so there is no routing capability possible with these data packets. Figure 6.20 illustrates the different classes of PROFINET.

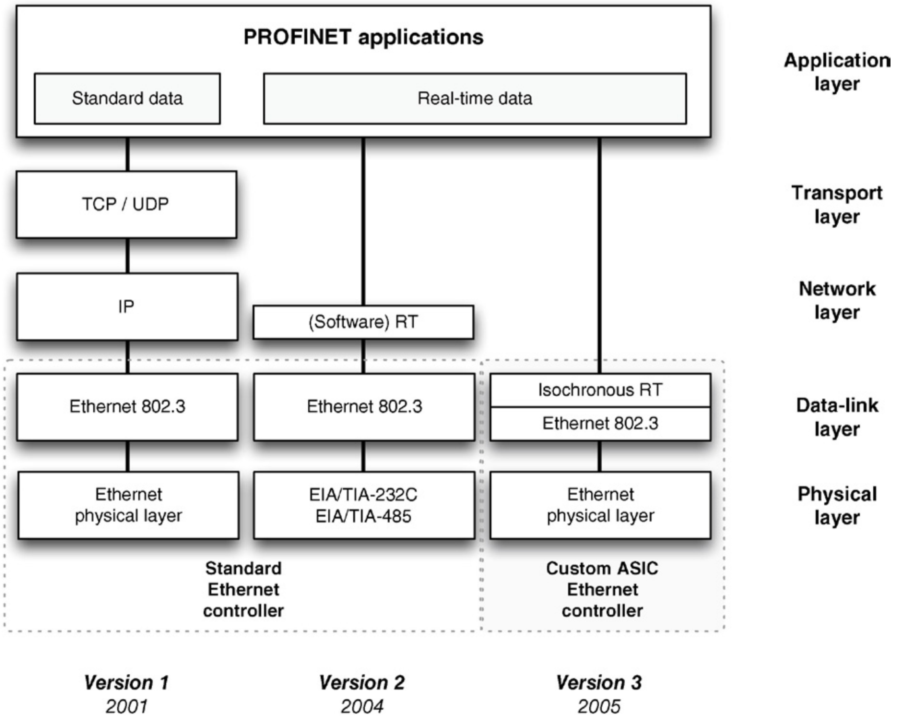


FIGURE 6.20 PROFINET implementation.

Security Concerns

PROFINET is a real-time Ethernet protocol, and as such it is susceptible to any of the vulnerabilities of Ethernet. The extent of the risk is highly dependent on the technology deployed, since newer devices can utilize proprietary hardware making unauthorized network access more challenging than the general-purpose TCP/IP implementation. When used over the IP, it is also susceptible to any vulnerabilities of IP; however, the real-time implementations of PROFINET also employ nonroutable network communications offering some protection against remote or adjacent network vectors.

Security Recommendations

As with many fieldbus protocols, the inherent lack of authentication and vulnerability of the protocol requires strong isolation of the bus. PROFINET TCP/IP represents the greatest risk as it can be transmitted over standard business and industrial networks. It should be tightly controlled and within less-trusted business networks, used only over authenticated and encrypted networks. It is not possible to segment PROFINET networks that contain devices that must communicate with each other (e.g. VLANs are not supported between PROFINET devices for logical segmentation); therefore, careful consideration in the deployment of zones and conduits should be taken (see Chapter 9, "Establishing Zones and Conduits"). Monitoring of Ethernet networks for unauthorized or suspicious use of PROFINET should be implemented including monitoring of all conduits into PROFINET zones. Firewalls and ICS-aware intrusion prevention systems should be configured to explicitly deny PROFINET traffic outside of well-defined areas. Additional guidance can be obtained through PNO.[24]

ETHERCAT

EtherCAT is another real-time Ethernet-based fieldbus protocol classified as "Industrial Ethernet" (see PROFINET for more information), which uses a defined ethertype (0x88A4) to transport ICS communications over standard Ethernet networks. These messages can either be transported directly in an Ethernet frame or encapsulated as a UDP payload using port 34980/udp (0x88A4). EtherCAT communicates large amounts of distributed process data with a single Ethernet frame to maximize the efficiency of distributed process data communications requiring only a few bytes per cycle over Ethernet frames that may vary in size from 46 to 1500 bytes. This means that only one or two Ethernet frames are required for a complete cycle allowing for very short cycle times with low jitter easily allowing network synchronization tasks to occur as required by the IEEE 1588 Precision Time Protocol (PTP) standard. EtherCAT is able to meet the requirements of PTP without any additional hardware (not the case with other industrial protocols discussed). Slaves pass the frame(s) to other slaves in sequence, appending its appropriate response, until the last slave returns the completed response frame back.[25]

Security Concerns

EtherCAT is a real-time Industrial Ethernet protocol, and as such it is susceptible to any of the vulnerabilities of standard Ethernet. EtherCAT over UDP is transaction-less and so there is no inherent network-layer mechanism for reliability, ordering, or data integrity checks.

EtherCAT is sensitive and highly susceptible to DoS attacks as with many real-time Ethernet protocols. EtherCAT is easily disrupted via the insertion of rogue Ethernet frames into the network to interfere with time synchronization and is subject to spoofing and MitM attacks due to the lack of bus authentication, requiring the separation of EtherCAT from other Ethernet systems.

Security Recommendations

EtherCAT is a real-time Industrial Ethernet protocol making it necessary to provide Ethernet-based security at the perimeter of any EtherCAT network. It is also recommended that passive network monitoring be used to ensure the integrity of the EtherCAT network, and that the EtherCAT protocol is only being used by explicitly identified devices. No EtherCAT traffic should be allowed that is originating from an unauthorized, outside source. This can be accomplished using an ICS-aware intrusion prevention system or other network monitoring device capable of detecting and interpreting the EtherCAT protocol via UDP/IP. Static Ethernet address tables (MAC address) can be deployed to further protect real-time EtherCAT devices from external attack. Many switches provide features to provide MAC address control as well as tables to further restrict communications between EtherCAT devices. A network monitoring product or probe can also be used to detect Ethernet packets using EtherCAT's specific ethertype.

ETHERNET POWERLINK

Ethernet POWERLINK is also an "Industrial Ethernet" technology that uses Fast Ethernet as the basis for real-time transmission of control messages via the direct encapsulation of Ethernet frames. A master node is used to initiate and synchronize cyclic polling of slave devices. Communication is divided into three time periods, with the first being the transmission of a master "Start of Cycle" frame that provides a basis for the network synchronization. The master then polls each station. The second time period is devoted to synchronous communication allowing the slaves to respond only if they receive a poll request frame, ensuring that all master/slave communications occur in sequence. Slave responses are broadcast, eliminating source address resolution. Asynchronous communication occurs in the third period where larger, non–time-critical data are transmitted. POWERLINK is best used homogeneously because collisions are avoided solely via the carefully controlled request/response cycles. The introduction of other Ethernet-based systems could disrupt synchronization and cause a failure.[26]

POWERLINK is often used in conjunction with CANopen, an application-layer protocol based on CAN (Controller Area Network). CANopen enables the communication between devices of different manufacturers, and the protocol stacks are widely available including open-source distribution for both Windows and Linux platforms. The open nature of CANopen makes POWERLINK/CANopen a desirable combination for industrial networks requiring inexpensive solutions in Linux environments.[27]

Security Concerns

POWERLINK is a real-time Industrial Ethernet protocol, and as such it is susceptible to any of the vulnerabilities of other forms of Ethernet communication.

As with many real-time Ethernet protocols, POWERLINK is sensitive and highly susceptible to DoS attacks. POWERLINK is easily disrupted via the insertion of rogue Ethernet frames into the network, requiring the separation of POWERLINK from other Ethernet systems. The protocol itself is sensitive and highly susceptible to DoS attacks.

Security Recommendations

POWERLINK implementations will most likely have a clear demarcation from other networks because sensitivity of the cyclic polling mechanism requires separation from other non-POWERLINK Ethernet services. This demarcation can be leveraged to further isolate the industrial protocol, through the establishment of appropriate security zones and the definition of strong perimeter defenses at these boundaries. Static Ethernet address tables (MAC address) can be deployed to further protect real-time POWERLINK devices from external attack, since these are pure Ethernet-based messages and typically represent the most critical communications. Many switches provide features to provide MAC address control as well as tables to further restrict communications between EtherCAT devices.

SERCOS III

SERCOS (Serial Real-time Communications System) is a standardized open digital interface for communication between industrial controls, motion devices, and I/O devices. Version I and II of the interface was based on a fiber-optic ring to establish inter-device communication. Version III of the interface is an “Industrial Ethernet”-based implementation of the SERCOS interface that supports deterministic real-time control of motion and I/O applications. Like EtherCAT and POWERLINK, SERCOS III has the ability to directly place Ethernet frames on the network in order to obtain high-speed communications with very low jitter.[28] Networks can support up to 511 slave devices in either straight or ring topologies.

SERCOS III is a master–slave protocol that operates cyclically, using a mechanism in which a single Master Synchronization Telegram is used to communicate to slaves, and the slave nodes are given a predetermined time (again synchronized by the master node) during which they can place their data on the bus. All messages for all nodes are packaged into a Master Data Telegram, and each node knows which portion of the MDT it should read based upon a predetermined byte allocation.[29]

SERCOS III dedicates the use of the bus for synchronized real-time traffic during normal cycles; however, like other Industrial Ethernet protocols discussed, it allows unallocated time within a cycle to be freed up for other network protocols, such as TCP and UDP data, using IP. This “IP Channel” allows the use of broader network applications from the same device—for example, a web-based management interface that would be accessible to “office and wide area networks.”[30]

Security Concerns

SERCOS III is a real-time Industrial Ethernet protocol, and as such it is susceptible to any of the vulnerabilities of other forms of Ethernet communication. SERCOS III introduces new security concerns through the option to support embedded, open TCP/IP and UDP/IP communications. With this option enabled, a compromised RTU or PLC using SERCOS III could be used to launch an in-bound attack into other corporate communications systems, including industrial and business networks.

Security Recommendations

As with other Industrial Ethernet-based protocols, static Ethernet address tables (MAC address) can be deployed to further protect real-time SERCOS III devices from external attack, since these are pure Ethernet-based messages and typically represent the most critical communications. Many switches provide features to provide MAC address control as well as tables to further restrict communications between SERCOS III devices. SERCOS III should be isolated to control loops that require the protocol, and the use of IP channels should be restricted and avoided if possible. If IP channels are used, the extent and reach of the IP channel should be enclosed within an explicitly defined zone consisting of the SERCOS III master node and only those TCP/IP network devices that are absolutely required. Strong perimeter defenses should be installed in-band for all conduits into this zone using least privilege principles. Active monitoring of security device logs on the perimeter should be enabled due to the heightened risk from pivoting through networks using SERCOS III.

BACKEND PROTOCOLS

OPEN PROCESS COMMUNICATIONS

OLE for Process Control is not actually an industrial protocol, but "a series of standard specifications"[31] designed to simplify integration of various forms of data on systems from different vendors. In order to appreciate the impact OPC had on industrial automation, a brief history of OPC is warranted.

The original standard released in 1996 provided a mechanism for a standardized way for systems to exchange data across an Ethernet network using a core set of Microsoft technologies including Object Linking and Embedded (OLE), Component Object Model (COM), and Distributed Component Object Model (DCOM). The specification included standard sets of "objects," "interfaces," and "methods" to support this interoperability in industrial applications. The underlying mechanism to support this communication was based on interprocess communications using the Remote Procedure Call (RPC) protocol. The original set of standards that utilized the COM/DCOM infrastructure is today commonly referenced as "OPC Classic."

OPC has evolved significantly since its introduction nearly 20 years ago, and for that reason the OPC Foundation (the organization that oversees the standards) has introduced new meaning to the dated acronym from 'OLE for Process Control' to 'Open Process Communications." The "classic" set of standards originally focused on real-time data access (OPC-DA released 1996), historical data access (OPC-HDA released 2001), and alarms and events data (OPC-AE released 1999). This set was expanded to include data access via web services using extensible markup language (OPC-XMLDA released 2003), server-to-server and machine-to-machine communications (OPC-DX released 2003), and batch applications (OPC Batch released 2000). Since OPC relied on the DCOM infrastructure, users encountered significant problems in trying to manage OPC communication between security zones that were protected with firewalls including the lack of network address translation (NAT) support and session callbacks.

Technology was moving away from the DCOM infrastructure and toward the .NET Framework. Using Windows Communication Foundation (WCF), OPC .NET (formerly known as OPC-Xi or eXpress Interface) incorporates the functionality of DA, HDA, and AE on a simplified data model. This new technology provided users with significant security improvements to how OPC .NET traffic was managed on industrial networks across zones. The downside was that there was little vendor support for this enhanced standard resulting in a relatively small number of "gateway" type products.[32]

All standards up to this point depended on some form of underlying Microsoft technology—COM, DCOM, or .NET. This significantly limited the deployment within ICS architectures much below the supervisory networks due to the fact that most of the embedded devices (BPCS controllers, PLCs, RTUs, etc.) were not based on a Windows operating system that would support these classic standards. The idea was to move the communications model from COM/DCOM to a cross-platform service-oriented architecture (SOA) to support broader deployment to non-Windows devices, and ... better security! The OPC Unified Architecture (OPC-UA) specification was first released in 2006, and offers numerous improvements to the "classic" specifications while still supporting the underlying data integration requirements.

OPC Data Access "classic" is still one of the most widely deployed OPC specifications, and for the purposes of this book, is the one that will be discussed in more detail.

What it Does

OPC is one of the major "backend" protocols because it is designed to provide a higher level of integration between systems and subsystems, versus a fieldbus protocol that generally provides low-level data access and configuration.

OPC was originally motivated by the needs of end "users" and not system "vendors" to provide a common communications interface between diverse ICS components. The idea was to create a process industries technology that mirrored what Microsoft had done with device drivers in their newer Windows object-oriented operating systems. To digress briefly, many may remember the days of Windows 3.11 and the requirement for every application to possess drivers necessary to utilize a dot-matrix printer. Microsoft solved that problem when they released Windows 95. The manufacturing community was no different—significant time and effort were spent in the 1980s and 1990s simply providing basic integration between the various systems that now are common components within the integrated ICS architecture.

This was accomplished by leveraging Microsoft's DCOM communications API, reducing the need for device-specific drivers. In place of specific communications drivers for each device, simple device drivers could be written to interface with OPC. The use of OPC therefore minimized driver development and allowed for better optimization of core OPC interfaces.[33]

OPC's strengths and weaknesses come from its foundation, which is based upon Microsoft's OLE technology. OLE is used extensively in office document generation allowing the presentation of data to be separated from the application that generated it. A Word document could either "link" to a value calculated by a local or remote spreadsheet or "embed" the spreadsheet inside the document. This now allows OPC-connected devices to communicate and interact with minimal operator feedback (as in the case of the Office documents). The concept of cyber security did not really exist in 1996, which meant that there were significant security challenges that lie ahead to those implementing OPC.[34]

How it Works

OPC works in a client/server manner, where a client application calls a local process, but instead of executing the process using local code, the process is executed on a remote server. The remote process is linked to the client application and is responsible for providing the necessary parameters and functions to the server, utilizing a remote procedure call (RPC).

In other words, the stub process is linked to the client, but when a function is performed, the process is performed remotely, on the server. The server RPC functions then transmit the requested data back to the client computer. The client process then receives the data over the network, provides it to the requesting application, and closes the session, as shown in Figure 6.21.

In Windows systems, the requesting application typically loads RPC libraries at run-time, using a Windows dynamic link library (DLL).[35]

OPC is more complex than previous client/server industrial protocols because of this interaction with the calling application and the underlying DCOM architecture.

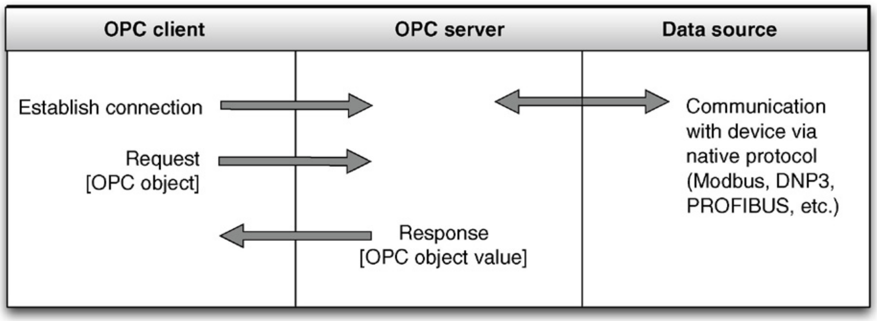

OPC client
OPC server
Data source
Establish connection
Request [OPC object]
Response [OPC object value]
Communication with device via native protocol (Modbus, DNP3, PROFIBUS, etc.)

FIGURE 6.21 Typical OPC protocol operation.

It interacts with various aspects of the host operating system, tying it closely to other host processes and exposing the protocol to a very broad attack surface. OPC also inherently supports remote operations that allow OPC to perform common control system functions.[36]

One aspect that makes OPC and DCOM very challenging when characterizing industrial networks and the communications that occur across these networks and through various conduits is how DCOM begins the session on one port and then transfers to another. Figure 6.22 illustrates a typical OPC session that does not incorporate server "callbacks."

Figure 6.22 shows how an initial request from an OPC Client to a corresponding OPC Server begins using a DCE BIND request to the Endpoint Mapper service listening on 135/tcp of the Server. Once the Client is authenticated against the Server and an OPC Instance created on the Server, the session shifts to a different connection, where the actual exchange of OPC data occurs. If a custom port range is not configured, this new port can be any randomly assigned port between 1024 and

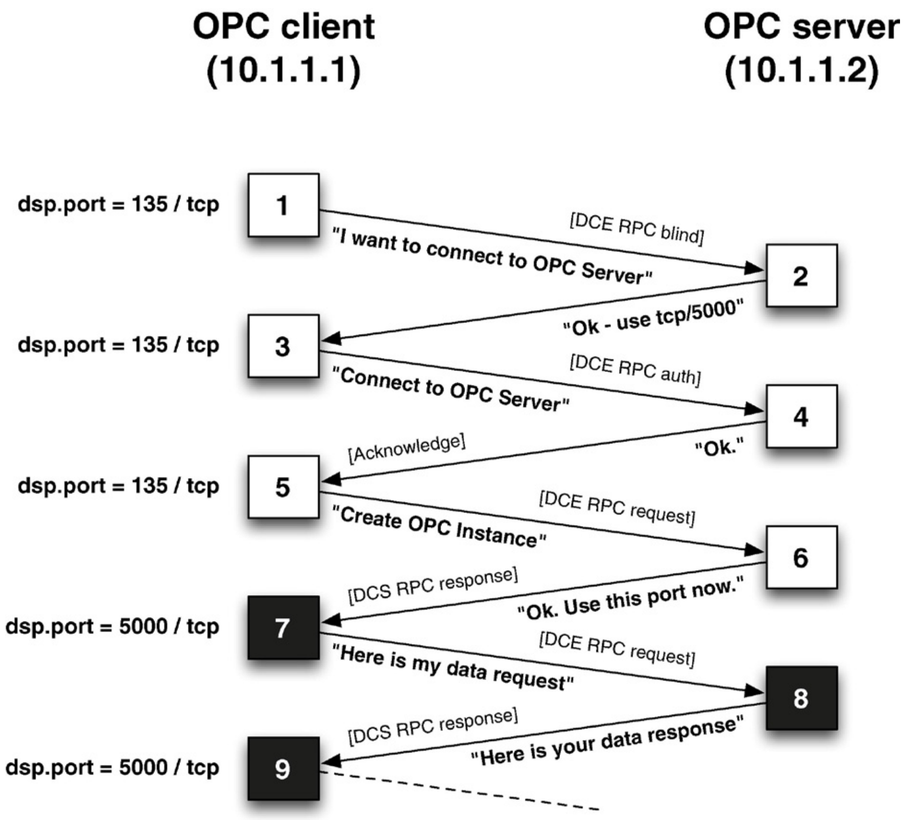


FIGURE 6.22 OPC client–server communications.

65535 depending on the operating system. If Server callbacks are used, the original session actually disconnects after the OPC Instance is created, and the OPC Server initiates a new session with the OPC Client. In other words, the OPC Server is now the network "source address" and the OPC Client is now the "destination address." A "tunneler" application can be installed to address this problem by allowing a point-to-point tunnel be created using a single predefined port where all RPC traffic (135 and the subsequent session port) is directed. The tunneler must be installed on the OPC Client and Server hosts, and should be qualified by the respective vendor to ensure that there is no impact to the performance of the other applications and services.

Where it is Used

As the name implies, Open 'Process' Communications is primarily used within industrial networks (i.e. not a common business network technology), including data transfer to data historians, data collection within HMIs, connectivity between serial fieldbus protocols like Modbus and DNP3 and ICS servers, and other supervisory controls, as shown in Figure 6.23. The deployment of OPC servers within ICS architectures can greatly simplify the data integration in the core ICS servers allowing all proprietary protocols and interfaces to be managed via local, distributed OPC Servers that contain the appropriate physical and application connectivity to a particular subsystem or device. This Server is then connected to various ICS servers

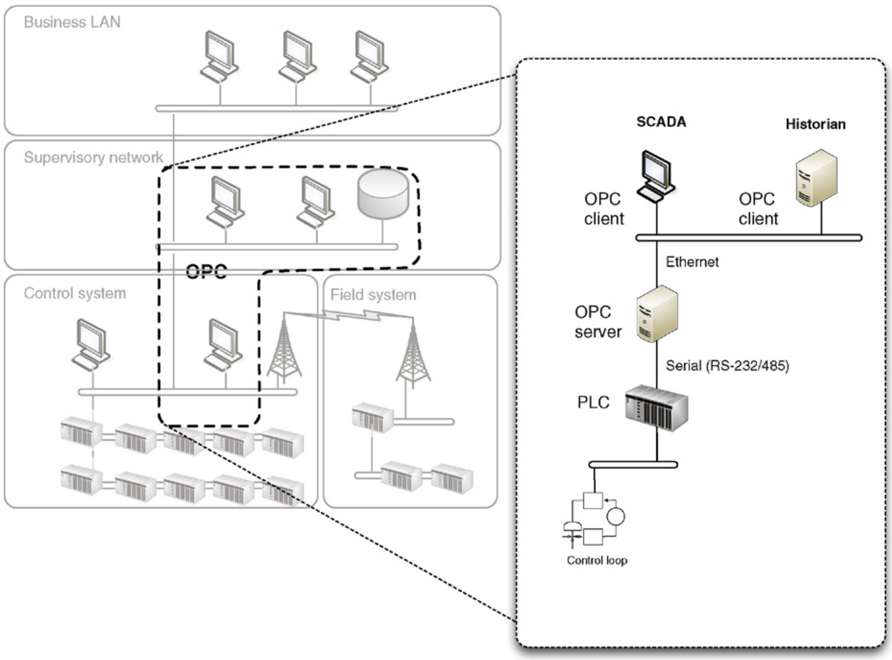


FIGURE 6.23 Typical OPC use within the industrial network architecture.

and components using a single, consistent mechanism. OPC is a Windows interconnection, so all communications occur either between Windows-based devices, or via OPC gateways that translate the RPC to the native fieldbus format. Because of the common use of RPC protocols within OPC, this opens the ICS environment to a very broad attack surface.

Security Concerns

OPC's use of DCOM and RPC makes it highly vulnerable to attack using multiple vectors, as it is subject to the same vulnerabilities as the more ubiquitously used OLE.[37] Classic OPC is rooted in the Windows operating system and is therefore susceptible to attack through exploitation of any vulnerability inherent to the OS.[38] Support for Windows XP with Service Pack 3 ended on April 2014 (XP-SP2 ended July 2010), meaning that OPC applications hosted on unsupported OSes can introduce significant risk to the integrity of manufacturing operations and potential health, safety, and environment (HSE) impact.

OPC and related ICS vulnerabilities can be tracked via a variety of sources including the US Department of Homeland Security Industrial Control System Cyber Emergency Response Team (ICS-CERT) and the Open Source Vulnerability Database (OSVDB). Many OLE and RPC vulnerabilities exist and are well known, including exploit modules for a variety of open-source and fee-based security frameworks like Metasploit and Canvas (see Chapter 7, "Hacking Industrial Systems"). It is difficult to patch production systems within an industrial network (see Chapter 8, "Risk and Vulnerability Assessments" and Chapter 10, "Implementing Security and Access Controls") so many of these vulnerabilities may still be in place, even if there is an available patch from Microsoft. The SQL Slammer worm actually caused global damage despite the fact that Microsoft released a patch to correct the vulnerability six months prior to the release of the worm.

Many basic host security concerns apply because OPC is supported on Windows. RPC requires local authentication to occur on both client and server hosts. This requires the creation of either a local or domain-based account that can be used by RPC for the OPC sessions. This account can introduce significant risk if it is not properly secured using a least privilege approach for just the essential OPC/DCOM services. This account is common to all hosts utilizing OPC, and if not properly protected and managed can lead to a widespread compromise in large ICS architectures. Many OPC hosts utilize weak authentication, and passwords are often weak when authentication is enforced. Many systems support additional Windows services that are irrelevant to ICS systems, resulting in unnecessary processes, which often correspond to open "listening" communication ports accessible via the network. Inadequate or nonexistent logging exacerbates these potential weaknesses by providing insufficient forensic detail should a breach occur, as Windows 2000/XP auditing settings do not record DCOM connection requests by default.[39]

Unlike the simple and single-purpose fieldbus protocols discussed earlier, OPC must be treated as an overall system integration framework, and implemented and maintained according to modern OS and network security practices.

Other security concerns of OPC include the following:

- Legacy authentication services – Systems within industrial networks are difficult to upgrade (due to limited maintenance windows, compatibility and interoperability concerns, and other factors); insecure authentication mechanisms remain in use. For example, Windows 2000 LAN Manager (LM) and NT LAN Manager (NTLM) authentication mechanisms are still used by default in many systems (enabled by default up to and including Windows XP and 2003 Server). These and other legacy authentication mechanisms may be vulnerable and susceptible to exploitation.[40]
- RPC vulnerabilities – OPC uses RPC making it susceptible to all RPC-related vulnerabilities, including several vulnerabilities that are exposed prior to authentication. Exploitation of underlying RPC vulnerabilities could result in arbitrary code execution, or DoS.[41]
- Unnecessary ports and services – OPC supports network protocols other than TCP/IP, including NetBIOS Extended User Interface (NetBEUI), Connection Oriented NetBIOS over InterNetwork packet Exchange (IPX), and Hyper Text Transport Protocol (HTTP) Internet services.[42]
- OPC Server Integrity – It is possible to create a rogue OPC server and to use that server for disruption of service, DoS, information theft through bus snooping, or the injection of malicious code.[43]

Security Recommendations

The newer and designed for security Unified Architecture (OPC-UA) specifications should be used where possible.

Regardless of the OPC specification used (Classic or Unified Architecture), all unnecessary ports and services should be removed or disabled from the OPC server. This includes any and all irrelevant applications, and all unused network protocols. All unused services may introduce vulnerabilities to the system that could result in a compromise of the Windows host, and therefore the OPC network.[44]

OPC servers should be isolated into a unique zone consisting only of authorized devices, and the zones(s) should be thoroughly secured using standard defense-in-depth practices, including a firewall and/or intrusion protection system that enforces strict control over the type, source, and destination of traffic to and from the OPC zone. Consideration should be given to application-aware firewalls that are capable of following the RPC session from the initial request (via 135/tcp) to response (a different port) and possible server "callbacks."

Because OPC is primarily used in a supervisory capacity, intrusion "prevention" systems can be considered in place of "detection" only, understanding that an IPS may block legitimate ICS traffic and result in a lack of visibility into control system operations potentially causing a Loss of View (LoV) or Loss of Control (LoC) situation. If information loss will be damaging to the control process or detrimental to business operations, use only an IDS.

Many threats can be detected through monitoring OPC networks and/or OPC servers (server activity can be monitored through the collection and analysis of Windows logs), and looking for specific behaviors, including the following:

- The use of non-OPC ports and services initiated from the OPC server (requires DCOM services to be configured to use specific port range to eliminate a wide range of "randomly" generated response ports).
- The presence of known OPC (including underlying OLE RPC and DCOM) exploits.
- OPC services originating from unknown OPC servers (indicating the presence of a rogue server).
- Failed authentication attempts or other authentication anomalies on the OPC server.
- Successful authentication attempts on the OPC server from unknown or unauthorized users.

Most commercially available IDS and IPS devices support a wide range of detection signatures for OLE and RPC and therefore can also detect many of the underlying vulnerabilities of OPC. Most open-source and commercial log analysis and threat detection tools are capable of collecting and assessing Windows logs.

Guidelines also have been created for proper hardening of OPC hosts, including "audit" files developed by Digital Bond as part of the Bandolier project that can be used with the Nessus vulnerability scanner to compare host settings against recommended vendor setting.[45]

TIP

OPC vulnerabilities may require the use of an ICS-aware intrusion protection system rather than an enterprise equivalent. Enterprise devices typically detect exploits via inspection of OLE, RPC, and DCOM but may not be able to detect all threats targeting OPC. In some cases, enterprise IDS/IPS devices may be adapted to detect a wider range of OPC threats, using SNORT compatible preprocessors and detection signatures available from Digital Bond as part of the QuickDraw IDS project.[46]

INTER-CONTROL CENTER COMMUNICATIONS PROTOCOL

The Inter-Control Center Communications Protocol (also known as TASE.2 or IEC60870-6, but more commonly referred to as simply ICCP) is a protocol designed for communication between control centers within the electric utility industry. Unlike fieldbus protocols such as Modbus and DNP3, ICCP is classified as a "backend" protocol like OPC because of the fact that it was designed for bidirectional Wide Area Network (WAN) communication between a utility control center and other control centers, power plants, substations, and even other utilities.

Much like the fundamental driver in the process industries developing OPC, electric utilities were also faced with ICS vendors and equipment suppliers utilizing many custom and proprietary protocols. A common protocol was needed to allow for reliable and standardized data exchange between utility control centers—especially when these control centers are operated by different owners, produce different products, or perform different operations. Standardization became necessary to support the unique business and operational requirements of the electrical utilities that require careful load balancing within a bulk system operated by many disparate facilities. In North America, the division of utilities among several responsible regional entities requires a means of sharing information between utilities as well as the regional entity. National and global energy markets require real-time information exchange for load distribution and trading that spans the boundaries of individual utilities.

A working group was formed in 1991 to develop and test a standardized protocol and to submit the specification to the International Electrotechnical Commission (IEC) for ratification and approval. The initial protocol was called ELCOM-90, or Telecontrol Application Service Element-1 (TASE.1). TASE.1 evolved into TASE.2, which is the most commonly used form of ICCP.[47]

What it Does

ICCP is used to perform a number of communication functions between control centers, including the following:

- Establishing a connection
- Accessing information (read requests)
- Information transmission (such as e-mail messages or energy market information)
- Notifications of changes, alarms, or other exception conditions
- Configuration of remote devices
- Control of remote devices
- Control of operating programs.

How it Works

The ICCP defines communication between two control centers using a client–server model. One control center (the server) contains application data and defined functions. Another control center (the client) issues requests to read from the server with appropriate server responses. Communications over ICCP occur using a common format in order to ensure interoperability.

ICCP support is typically integrated either directly into an ICS, provided via a gateway product, or provided as a software that can then be installed to perform gateway functions.

ICCP is primarily a unidirectional client–server protocol; however, most modern implementations support both functions, allowing a single ICCP device to function as both a client and a server, supporting bidirectional communication over a single connection.

ICCP can operate over essentially any network protocol, including TCP/IP; however, it is commonly implemented using the ISO transport on port 102/tcp, as defined in RFC 1006. ICCP is effectively a point-to-point protocol due to the use of a "bilateral table" that explicitly defines an agreement between two control centers connected with an ICCP link, as shown in Figure 6.24. The bilateral table acts as an access control list that identifies which data elements a client can access. The permissions defined within the bilateral tables in the server and the client are the authoritative control over what is accessible to each control center. The entries in the bilateral tables must also match on the client and the server, ensuring that the permissions are agreed upon by both centers (remembering that ICCP is used to interconnect to other organizations in addition to internal WAN links to substations).[48]

Where it is Used

ICCP is widely used between control system zones and between distinct control centers, as shown in Figure 6.25. It is also commonly deployed between two electric utilities, between two control systems within a single electric utility, and between a main control center and a number of substations.

Security Concerns

ICCP represents several security concerns much like most of the other fieldbus and backend protocols discussed. ICCP is susceptible to spoofing, session hijacking, and any number of attacks made possible because of the following:

- Lack of authentication and encryption – ICCP does not mandate authentication or encryption, most often deferring these services to lower protocol layers. Although "Secure ICCP"[49] does exist, it is not ubiquitously deployed.
- Explicitly defined trust relationships – The exploitation of bilateral tables could directly compromise security of ICCP servers and clients.

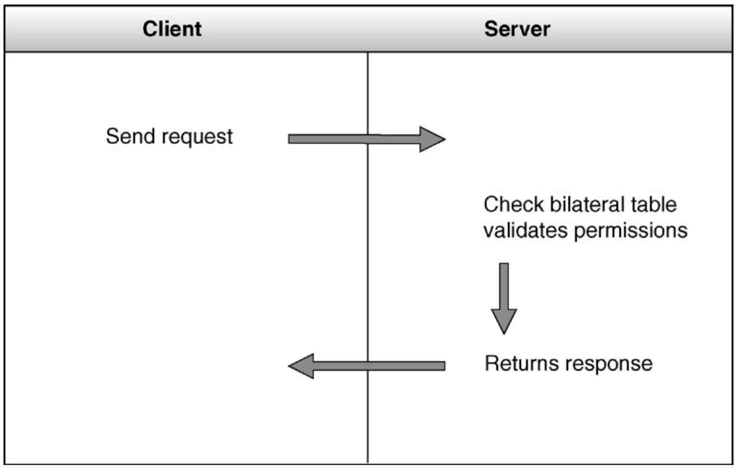


FIGURE 6.24 ICCP protocol operation.

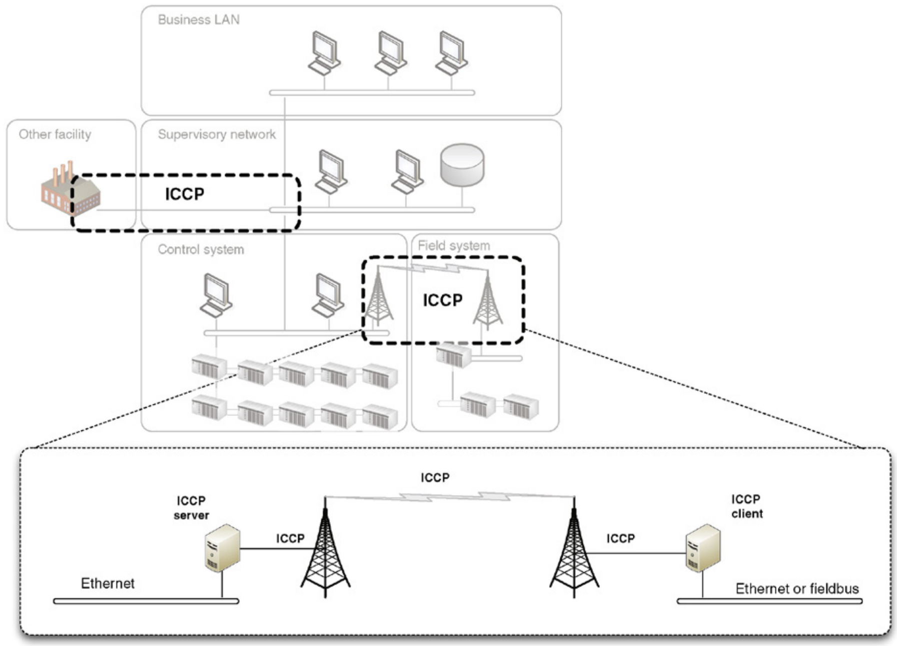


FIGURE 6.25 Typical ICCP use within the industrial network architecture.

- Accessibility – ICCP is a Wide Area Protocol making it highly accessible and susceptible to many attacks including DoS attacks from being exposed to public and/or shared networks versus traditional closed or private industrial networks within a plant environment.

The limited security mechanisms within ICCP are configured on the ICCP server, meaning that the successful breach of the server through an MitM or other attack opens the entire communication session up to manipulation.

Security Improvements Over Modbus and DNP

ICCP offers several improvements over more basic fieldbus protocols, such as Modbus and DNP3, including the following:

- ICCP's use of bilateral tables provides basic control over the communication path by explicitly defining which ICCP clients and servers can communicate.
- A secure version of ICCP exists that incorporates digital certificate authentication and encryption.

Security Recommendations

Secure ICCP variants should be used wherever possible and supported by the current vendors installed within a particular site. There are several known vulnerabilities with ICCP that have been reported by ICS-CERT. Proper system hardening and regular system assessments and patching of ICCP servers and clients is recommended because there are known exploits in the wild and ICCP is a WAN protocol.

Extreme care should be taken in the definition of the bilateral table. The bilateral table is the primary enforcement of policy and permissions between control centers. Malicious commands issued via ICCP could directly alter or otherwise impact control center operations.

ICCP clients and servers should also be isolated into a unique zone consisting only of authorized client–server pairs (multiple zones can be defined for devices communicating to multiple clients), and the zones(s) should be thoroughly secured using standard defense-in-depth practices, including a firewall (industrial grade if installed in production environments) and/or intrusion protection system that enforces strict control over the type, source, and destination of traffic over the ICCP link. As with other industrial protocols, preference should be given to security practices that are capable of deep-packet inspection of ICCP traffic, if available. Many of the recommendations described for other industrial protocols are equally applicable for ICCP, including the creation of network baselines and deployment of network whitelists.

Many malicious behaviors can be detected through monitoring of the ICCP link, including the following:

- Intruders gaining unauthorized access to the control center network, via overlooked access points, such as dial-up or remote access connections to partner or vendor networks with weak access control mechanisms.
- Insider threats, including unauthorized information access and transmission, alteration of secure configurations, or other malicious actions can be the result of a physical security breach within a control center, or of a disgruntled employee.
- A DoS attack resulting from repeated information requests (“spamming”) that utilize the server’s available resources and prevent legitimate operation of the ICCP link.
- Malware infecting the ICCP server or other devices on the network could be used to exfiltrate sensitive information for purposes of sabotage (e.g. theft of command function codes), financial disruption (e.g. alteration of energy metrics used in trading), or various other malicious intents.
- Interception and modification of ICCP messages (i.e. MitM) attacks.

Monitoring of ICCP protocol functions can also detect suspicious or malicious behavior, such as

- Function “read” codes that could be used to exfiltrate protected information.
- Function “write” codes that could be used to manipulate client or server operations.
- Traffic on port 102/tcp that is not ICCP or other authorized protocol (PROFINET utilizes 102/tcp ISO-TSAP for its industrial Ethernet communications).

- ICCP traffic that is not sourced by and destined to defined ICCP servers or clients.

CAUTION

Intrusion Prevention Systems are able to actively block suspect traffic by dropping packets or resetting TCP connections. However, Intrusion Prevention Systems deployed on industrial networks should only be configured to block traffic after careful consideration and tuning. Unless you are confident that a given signature will not inadvertently block a legitimate control command, the signature should be set to alert, rather than block (i.e. operate in "detection" mode rather than active "prevention" mode).

An ICS-aware intrusion protection system can be configured to monitor for these activities using ICCP signatures, such as those developed and distributed by Digital Bond under the QuickDraw SCADA ICS project. An application-aware firewall, industrial protocol filter, or application data monitor may be required to validate ICCP sessions and ensure that ICCP or the underlying RFC-1006 connection have not been "hijacked" and that messages have not been manipulated or falsified.

NOTE

Digital Bond removed the ICCP SNORT rules from the QuickDraw SCADA IDS signature list because of the generation of too many false negatives. With most IDS/IPS engines, preprocessors are needed to appropriately parse a protocol allowing the development of reliable rules.[50]

ADVANCED METERING INFRASTRUCTURE AND THE SMART GRID

The smart grid is a term encompassing many aspects of modern power generation, transmission and distribution. Although smart grid technology might seem irrelevant to many industrial network systems outside of the electric utility industry, it is discussed briefly here because of its broad reach and vulnerable attack surface. The smart grid is a widely distributed communication network that touches power generation and transmission systems, along with many end user networks. The smart grid represents an easily accessible network that contains many vectors to many possible targets. Once compromised, an attacker could use the network to attack the power utility's network, or to attack the networks of connected home and businesses.

The term "smart grid" is widely used and generally refers to a new era of energy distribution built around an Advanced Metering Infrastructure (AMI). AMI promises many new features designed to increase the efficiency and reduce the costs of energy distribution. Common AMI features include remote meter reading, remote billing, demand/response energy delivery, remote connect/disconnect, and remote payment and prepayment.[51]

At a high level, the smart grid requires coordination among the following systems:

- Bulk electric generation systems
- Electric transmission systems
- Electric distribution systems
- Customer information and management systems
- Usage and meter management systems
- Billing systems
- Interconnected network systems, including neighborhood area networks (often using wireless mesh technologies); metropolitan area networks (MAN); home area networks (HAN); and business area networks (BAN)

The smart grid is essentially a large, end-to-end communications system interconnecting power suppliers to power consumers (see Figure 6.26). It is made of highly diverse systems, using diverse protocols and network topologies. Smart grids even introduce new protocols. To support home- and business-based service portals, smart metering introduces HAN and BAN protocols, such as Zigbee and HomePNA, as well as power line protocols, such as IEC 61334, Control Network Power Line (PL) Channel Specification, and Broadband over Powerline (BPL). The data link and application protocols are too numerous to discuss in detail, though it is widely accepted that TCP/IP will be leveraged for network-layer communications.[52]

These specific protocols will not be discussed within this book, but it is still important to recognize that the disparate nature of these systems requires that several distinct operational models and network architectures combine to form a single end-to-end communications path, as illustrated in Figure 6.23. This means that while many distinct smart grid protocols may be used, the smart grid as a whole should be considered as a single, readily accessible communications network that is vastly interconnected.

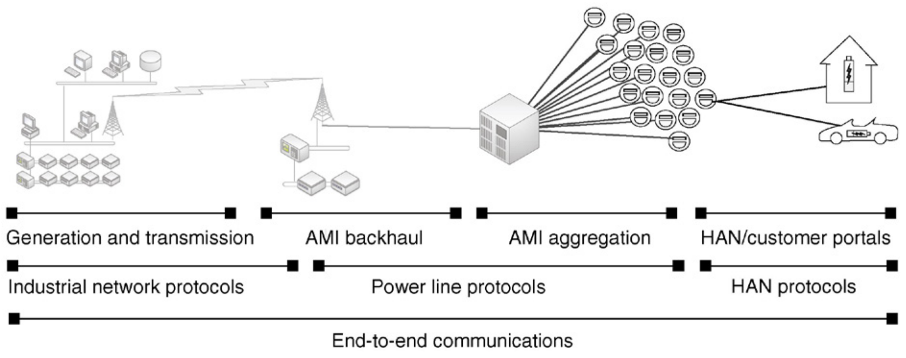

Generation and transmission
AMI backhaul
AMI aggregation
HAN/customer portals
Industrial network protocols
Power line protocols
HAN protocols
End-to-end communications

FIGURE 6.26 Smart grid operational areas and protocols.

SECURITY CONCERNS

The security concerns of the smart grid are numerous. AMI represents an extremely large network that touches many other private networks and is designed with command and control capabilities in order to support remote disconnect, demand/response billing, and other features.[53] Combined with the lack of industry-accepted security standards, the smart grid represents significant risk to connected systems that are not adequately isolated. Specific security concerns include the following:

- Smart meters are readily accessible and therefore require board- and chip-level security in addition to network security.
- Smart grid protocols vary widely in their inherent security and vulnerabilities.
- Neighborhood, home, and business LANs can be used as an ingress to the AMI, and as a target from the AMI.
- Smart grids are ultimately interconnected with critical power generation, transmission and distribution systems.
- Smart grids represent a target to private hackers (for financial gain or service theft) as well as to more sophisticated and serious attackers (for sociopolitical gain or cyber warfare).

SECURITY RECOMMENDATIONS

The best recommendation for smart grid security at this point is for electric utilities to carefully assess smart grid deployments and to perform risk and threat analysis early in the planning stages. A similar assessment of the system should be performed for end users who are connected to the smart grid who could become a potential threat vector into the business (or home) networks.

Clear delineation, separation of services, and the establishment of strong defense-in-depth at the perimeters will help to mitigate the risk from threats associated with the smart grid. This could represent a challenge (especially in terms of security monitoring) for smart grid operators, due to the broad scale of smart grid deployments, which could contain hundreds of thousands or even millions of intelligent nodes. It may be necessary then to carve out smart grid deployments into multiple, smaller and more manageable security zones.

INDUSTRIAL PROTOCOL SIMULATORS

One way to learn and understand how an industrial protocol operates is to purchase the appropriate hardware (i.e. PLC) and software (SCADA). This can be expensive and time consuming. Another more practical approach is through the deployment of client and server simulators capable of mimicking the protocol within a physical or virtualized computing environment.

Simulators are readily available for royalty-free protocols like Modbus/TCP, but can be limited for the licensed protocols. In the latter cases, one alternative approach is the use of “trial” or “demonstration” software packages. The products below were available at the time of publishing, and are provided for illustrative purposes only.

MODBUS

There are a range of Modbus simulators that will support both Modbus RTU and ASCII formats using both serial and Ethernet communication. The ModbusPal package available on Sourceforge is particularly interesting because it is based on Java allowing it to be easily transported between different platforms (Windows, Mac, Linux). It also features an “automation” capability allowing it to vary inputs and outputs providing the ability to change data at the source. ModbusPal supports “user-defined” commands using function codes 65–72 and 100–110.

Triangle Microworks Communication Protocol Test Harness provides not only protocol simulation, but actual simulation of a variety of devices as well, allowing this to be a tool used by ICS software developers as part of protocol compliance testing. The Test Harness supports a range of protocols including Modbus/TCP, DNP3, and IEC 60870-5, and is available as a paid download or a 21-day evaluation version.

Modsak is a software package from Wingpath Software Development that supports either master or client modes. A three-day trial version is available that offers a range of features, including support for Modbus “user-defined” functions.

DNP3 / IEC 60870-5

The Axon Group offers a free simulation package for DNP3 and IEC 60870-5. The Communication Test Harness from Triangle Microworks also supports DNP3 and can operate as the master station or outstation. More advanced options are available through a variety of sources that provide DNP3 protocol libraries for custom application development.

OPC

Matrikon and Kepware are two leading suppliers of OPC products to a variety of ICS industry segments, both offering demonstration versions of their OPC applications. Matrikon offers a set of free OPC test tools that support the creation of OPC clients and servers, as well as trial versions of most of their applications including various system interface servers, protocol tunnelers, and more. Kepware offers similar trial licenses for their OPC server, as well as a linking package that can be used to connect two OPC servers.

ICCP / IEC 60870-6 (TASE.2)

Triangle Microworks IEC 60870-6 (TASE.2/ICCP) Test Tool is available as a paid license or a 21-day evaluation version with support for client and server roles. The package supports ICCP blocks 1, 2, and 5 with full support of writes, reads, controls, dynamic data sets, and dataset transfer sets. It also allows for models to be created via .csv and .xml files.

PHYSICAL HARDWARE

Investing in physical hardware to support a training and test laboratory does not have to be overly expensive. Many suppliers including ABB, Allen-Bradley, Schneider Electric, Siemens and Wago offer affordable, compact programmable devices that can support multiple protocols within a single device. Nearly all products will offer support for Modbus/TCP due to its widespread use, but can also be supplied with EtherNet/IP, PROFINET, and EtherCAT capabilities. Another very economical method of obtaining physical hardware is through reseller or auction websites like eBay.

SUMMARY

Industrial networks use a variety of specialized protocols at multiple layers in the network to accomplish specific tasks, often with careful attention to synchronization and real-time operation. Each protocol has varying degrees of inherent security and reliability, and these qualities should be considered when attempting to secure these protocols. All of these protocols are susceptible to cyber-attack using relatively simple MitM mechanisms because industrial network protocols, in general, lack sufficient authentication or encryption. These attacks can be used to disrupt normal protocol operations or potentially alter or otherwise manipulate protocol messages to steal information, commit fraud, or potentially cause a failure of the control process itself including mechanical equipment sabotage (e.g. Stuxnet).

These protocols can be reasonably secured by understanding them and isolating each into its own carefully defined security zone with related conduits (see Chapter 9, "Establishing Zones and Conduits"). The creation of zones based purely on physical devices is possible and relatively simple because each protocol has specific uses within a control system. Since industrial network protocols are used more widely over Ethernet and TCP/IP-UDP/IP, the creation of clean zone boundaries becomes more difficult, as these boundaries begin to overlap. The use of "business" network protocols to transport fieldbus protocols should be avoided unless absolutely necessary for this reason, and be especially scrutinized and tested where they are necessary.

ENDNOTES

1. IEC 61784-1:2010 "Industrial communication networks - Profiles - Part 1: Fieldbus profiles," published June 1, 2011.
2. "Schneider Electric Modicon History," <http://www.plcdev.com/schneider_electric_modicon_history> (cited: January 7, 2014).
3. Modbus Organizations, "Modbus Application Protocol Specification," Version 1.1b, Published December 28, 2066.

4. Ibid.
5. Ibid.
6. AEG Schneider Autotmation, "Modicon Modbus Plus Nework Planning and Installation Guide," 890-USE-100.00 Version 3.0, April 1996.
7. Triangle MicroWorks, "Using DNP3 & IEC 60870-5 Communication Protocols in the Oil & Gas Industry," Revision 1, published March 26, 2001.
8. Triangle MicroWorks, "Modbus and DNP3 Communication Protocols," <http://trianglemicroworks.com/docs/default-source/referenced-documents/Modbus_and_DNP_Comparison.pdf> (cited: January 8, 2014).
9. The DNP Users Group, DNP3 Primer, Revision A. <http://www.dnp.org/About/DNP3%20Primer%20Rev%20A.pdf>, March 2005 (cited: November 24, 2010).
10. G.R. Clarke, Deon Reynders Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems, Newnes, Oxford, UK and Burlington MA, 2004.
11. The DNP Users Group, DNP3 Primer, Revision A. <http://www.dnp.org/About/DNP3%20Primer%20Rev%20A.pdf>, March 2005 (cited: November 24, 2010).
12. Ibid.
13. Digitalbond SCADAPEDIA, Secure DNP3. <http://www.digitalbond.com/wiki/index.php/Secure_DNP3>, August 2008 (cited: November 24, 2010).
14. Ibid.
15. The DNP Users Group, DNP3 Primer, Revision A. <http://www.dnp.org/About/DNP3%20Primer%20Rev%20A.pdf>, March, 2005 (cited: November 24, 2010).
16. A.B.M. Omar Faruk, Testing & Exploring Vulnerabilities of the Applications Implementing DNP3 Protocol, KTH Electrical Engineering, Stockholm, Sweden, June 2008.
17. V.M. Igure, Security assessment of SCADA protocols: a taxonomy based methodology for the identification of security vulnerabilities in SCADA protocols, VDM Verlag Dr. Müller Aktiengesellschaft & Co. KG, 2008.
18. "Industrial Ethernet: A Control Engineer's Guide," Cisco, April 2010.
19. Prof. Dr.-Ing. J. Schwager, "Information about Real-Time Ethernet in Industry Automation," Reutlinger University, <http://www.pdv.reutlingen-university.de/rte/>, (cited: January 10, 2014).
20. Industrial Ethernet Facts, "System Comparison: The 5 Major Technologies," Ethernet POWERLINK Standardization Group, Issue 2, February 2013.
21. Industrial Ethernet Facts, "System Comparison: The 5 Major Technologies," Ethernet POWERLINK Standardization Group, Issue 2, February 2013.
22. Open Device Vendor Assocation (ODVA), "Common Industrial Protocol (CIP)," Publication PUB00122R0–ENGLISH, 2006.
23. Open-Device Vendors Association, "Securing EtherNet/IP Networks," PUB00269R1, 2011.
24. PROFIBUS Nutzerorganisation e.V., "PROFINET Security Guidelines: Guideline for PROFINET," Version 2.0, November 2013.
25. The EtherCAT Technology Group, Technical introduction and overview: EtherCAT—the Ethernet Fieldbus. <http://www.ethercat.org/en/technology.html#5>, May 10, 2010 (cited: November 24, 2010).
26. P. Doyle, Introduction to Real-Time Ethernet II. The Extension: A Technical Supplement to Control Network, vol. 5, Issue 4, Contemporary Control Systems, Inc., Downers Grove, IL, July 2004.
27. Ethernet POWERLINK Standardization Group, CANopen. <http://www.ethernet-powerlink.org/index.php?id=39>, 2009 (cited: November 24, 2010).

28. SERCOS International, Technology: Introduction to SERCOS interface. <http://www.sercos.com/technology/index.htm>, 2010 (cited: November 24, 2010).
29. SERCOS International, Technology: Cyclic Operation. <http://www.sercos.com/technology/cyclic_operation.htm>, 2010 (cited: November 24, 2010).
30. SERCOS International, Technology: Service & IP Channels. <http://www.sercos.com/technology/service_ip_channels.htm>, 2010 (cited: November 24, 2010).
31. OPC Foundation, "What is OPC?," <http://www.opcfoundation.org/Default.aspx/01_about/01_whatis.asp?MID=AboutOPC>, (cited: January 9, 2014).
32. OPC Foundation, "Certified Products," <http://www.opcfoundation.org/Products/Products.aspx>, (cited: January 9, 2014).
33. Ibid.
34. Digital Bond, British Columbia Institute of Technology, and Byres Research. OPC Security White Paper #2: OPC Exposed (Version 1-3c), Byres Research, Lantzville, BC and Sunrise, FL, November 13, 2007.
35. Microsoft Corporation, RPC Protocol Operation. <http://msdn.microsoft.com/en-us/library/ms818824.aspx> (cited: November 4, 2010).
36. European Organization for Nuclear Research (CERN), A Brief Introduction to OPC™ Data Access. <http://itcofe.web.cern.ch/itcofe/Services/OPC/GeneralInformation/Specifications/RelatedDocuments/DASummary/DataAccessOvw.html>, November 11, 2000 (cited: November 29, 2010).
37. "OPC Security Whitepaper #3: Hardening Guidelines for OPC Hosts," DigitalBond, British Columbia Institute of Technology, Byres Research, November 13, 2007.
38. Digital Bond, British Columbia Institute of Technology, and Byres Research. OPC Security White Paper #2: OPC Exposed (Version 1-3c), Byres Research, Lantzville, BC and Sunrise, FL, November 13, 2007.
39. Ibid.
40. Ibid.
41. Ibid.
42. Ibid.
43. Ibid.
44. Ibid.
45. DigitalBond, "Bandolier," <http://www.digitalbond.com/tools/bandolier/> (cited: January 9, 2014).
46. DigitalBond, "QuickDraw SCADA IDS," <http://www.digitalbond.com/tools/quickdraw/> (cited: January 9, 2014).
47. J.T. Michalski, A. Lanzone, J. Trent, S. Smith, SANDIA Report SAND2007-3345: Secure ICCP Integration Considerations and Recommendations, Sandia National Laboratories, Albuquerque, New Mexico and Livermore, California, June 2007.
48. Ibid.
49. J. Michalski, A. Lanzone, J. Trent, S. Smith, "Secure ICCP Integration: Considerations and Recommendations," Sandia Report SAND2007-3345, printed June 2007.
50. DigitalBond, "Bandolier Security Audit File for SISCO ICCP Server," <https://www.digitalbond.com/blog/2011/02/14/bandolier-security-audit-file-for-sisco-iccp-server/>, (cited: January 9, 2014).
51. UCA® International Users Group, AMI-SEC Task Force, AMI System Security Requirements, UCA, Raleigh, NC, December 17, 2008.

52. National Institute of Standards and Technology, NIST Special Publication 1108: NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0, February 2010.
53. UCA® International Users Group, AMI-SEC Task Force, AMI system security requirements, UCA, Raleigh, NC, December 17, 2008.
54. Open Device Vendors Association (ODVA), "Common Industrial Protocol," PUB00122R0-ENGLISH, 2006.