

Unit 3:

Introduction to Industrial Networks

At one time, it was thought that industrial automation networks were different from the kinds of networks used for IT. In fact, the earliest automation networks were not even considered as networks at all but as serial buses. The term *fieldbus* stems from these thoughts. Naturally, each network was designed to solve one problem, then extended to solve other, perhaps related, problems. Since each supplier's business model was directed toward a slightly different business niche, the resulting *bus* turned out to be different from any other.

As long as industrial automation networks were slow and uncomplicated, no special components were required. For example, EIA-232,¹ EIA-422/423, and EIA-485 were often used for the physical layers, supported by commodity semiconductors. Early protocols were simple enough to execute on 8-bit

1. Some people remember this as RS-232, but the standards organizations, Electronic Industries Association (EIA) and the Telecommunications Industry Association (TIA), prefer to designate it as EIA or TIA-232.

microprocessors such as the 8051, Z80, or 6809. When speeds became higher and protocols became richer in functionality, custom silicon became necessary to implement these networks. Custom silicon is expensive to design (nonrecurring engineering, or NRE) and, because the volumes are small compared with volumes in the IT market, expensive to manufacture.

The first approach to fix this problem was to standardize industrial automation networks through standards committees or the establishment of de facto standards by opening the specifications to multivendor committees. The theory was that if many system suppliers used the same chip set, there would be an economy of scale and a lower manufacturing cost. It just didn't work! There were too many chip sources and independent chip designs. The not invented here (NIH) syndrome prevailed.

The trend is clearly to shelve the idea that industrial automation networks are somehow different from IT networks. The clear trend is to use commercial off-the-shelf (COTS) components and adapt them through software to industrial automation applications. Since Ethernet was the clear winner in the IT market, it is no surprise that Ethernet is the basis for the newest evolution of industrial automation networks—at least at the high-performance end—which leaves the lowest-level networks used for connecting sensors and actuators with a different solution. These networks are also migrating and converging in both directions with commodity silicon as the basis. Some low-level networks will most likely use scaled-down versions of the higher-performance industrial automation networks, whereas others will use low-cost silicon developed for other markets.

Finally, it should be obvious by now that any discussion of industrial automation networks must consider the software

used for the upper layer visible to the end user. All network architectures are described by the International Standards Organization (ISO) standard and Open Systems Interconnection (OSI) basic reference model: standard ISO/IEC 7498-1:1994. This model is illustrated in Figure 3-1 and is divided into seven parts. When we say *network protocol*, we are talking about stuff in these layers. The end user only cares about the connection to the physical layer (wires coming out the bottom, or radio signals) and the features and functions made available at the top. Yes, the protocol is important, but a lot less important than the claims made by the network designers. However, if the middle layers of two networks are different, they cannot interoperate without a gateway device between them.

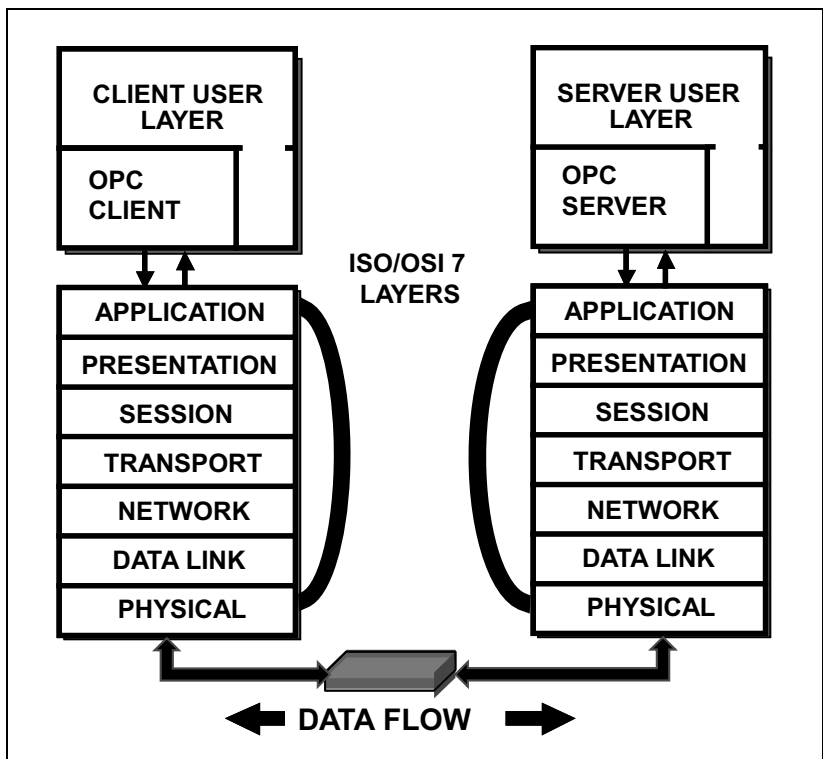


Figure 3-1. Network Layers

Notice that there are two layers above the ISO/OSI seven layers. The object linking and embedding for process control (OPC) layer has the benefit of adapting the network layers to the host system. Thus the client user layer only needs to be created knowing that it will be used with a server running compatible OPC software. With OPC, the details of the network layers are effectively hidden from view. It should also be noted that there are other methods of isolating the network application layer from the user layer software by using other network technologies, incorporated in the user layer, that do not use OPC. This is illustrated by the direct coupling of the user layer to the top of the communications protocol stack. Usually this is done to take advantage of the efficiency of the user layer connections and to make data transfers more deterministic than allowed by OPC.

Finally, in Figure 3-1, the network wiring is drawn with a little box between the client and server. This box represents the physical network as more than just wire and cable. Very often, there are active switches and converters in these networks for a variety of reasons that we will discuss later when each network is described. The word *cable* is often used since it includes metallic wiring and fiber optics. It also should include wireless connections, but that would require the use of the oxymoron *wireless cable*, an expression actually in use in the telecommunications market but not yet in industrial automation.

Next, we will introduce the generic types of networks used in industrial automation.

3.1 Sensor Networks

At the very lowest level of network functionality are the sensor networks. Generally, sensors themselves are at the bottom of the industrial automation system value chain and are designed

to be inexpensive since many sensors (and actuators as well) are required for typical applications. Sensors provide basic data to the control system, such as the presence of an object or a physical property such as temperature. Their mechanisms vary according to the desired accuracy and reliability.

The simplest sensor is the electromechanical limit switch used to indicate that an object is present or not. For example, limit switches often are used to detect boxes on a conveyor belt as they pass a reading station. They are also used on control valves to indicate when the valves are fully open or fully closed. Limit switches must be powered (sometimes called *biased*) to detect the open/closed switch position. When used with a PLC, each limit switch typically requires two wires to connect to the I/O terminations on a digital input card or termination module interface that is plugged into a slot on a multiplexer (most often called a *remote I/O* or *block I/O* unit). There are also digital termination modules for networked I/O on which the sensor wiring is connected. These digital values are typically reported to the controller as a bit in an input register.

Other technologies used to detect position are photocells and proximity detectors. Both of these technologies are somewhat more expensive than limit switches, but provide no more information than does the ON/OFF switch position. However, they do not have moving parts and generally have a longer service life. Photocells require a source of light in addition to the light-sensitive detector. Proximity detectors may be magnetic, and require no separate power source to detect iron or steel (magnetic) objects. When the object is composed of nonmagnetic materials, such as paper, plastic, or aluminum, a power source is used to generate an inductive field that will be modified by the mass of the object and can be detected.

Sensor networks are designed to reduce the point-to-point wiring needed to connect the limit switch, proximity sensor, solenoid valve, or photocell to the I/O interface. This is done in two ways: (1) put a network driver inside the sensor or actuator itself, or (2) bring the I/O interface close to the sensor or actuator so that the connections are very short. There are products on the market that do it both ways. The I/O interface usually terminates 4 to 16 I/O points and is connected to the PLC or other type of controller by the sensor network that transmits digital data for all points.

Sensor network I/O electrically detects the states of the sensor and converts it to a 1 or 0 in a status word. The status word is then transmitted across the network to a terminating device called a *scanner* that is usually in a remote I/O rack, a PLC, or a computer. The scanner is responsible for assembling the status words from each sensor network I/O node into a register in the device. Each sensor network has its own method for mapping the sensor status to the I/O registers. The distinguishing factor of sensor networks is that the sensor, actuator, and network node do nothing more than convert the sensor or actuator state to or from the network status word. No conditioning of the signal or any other calculation is provided.

Most sensor networks are designed to transmit bias power to the sensors so that their present status can be sensed without a separate source of power for each device. In most cases, there is a module at the network I/O node that allows termination of more than one I/O point to share the cost of the node. Typically, this is convenient since sensors are frequently clustered together around a common piece of equipment.

Some sensor networks are wired in a daisy chain or multidrop topology to reduce the field wiring as much as possible. Other sensor networks are wired in a star topology to reduce the

latency delays of data transmission. Still other sensor networks are wired in a ring topology for network reliability. Figure 3-2 illustrates these network topologies.

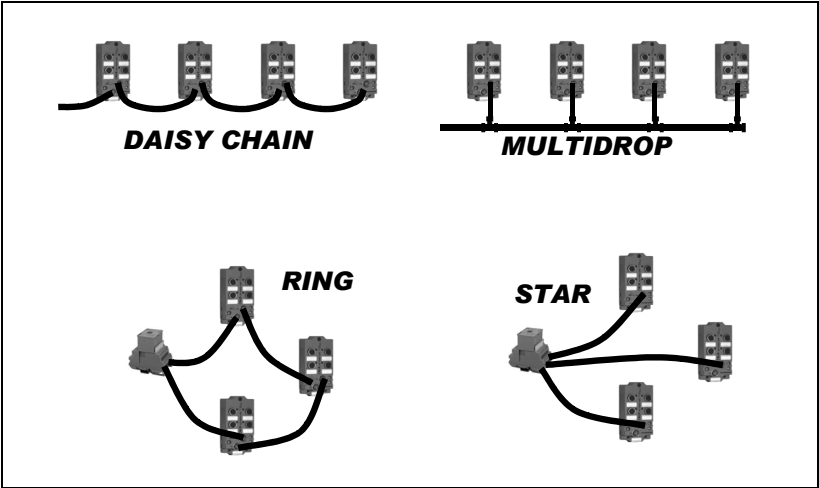


Figure 3-2. Network Topologies

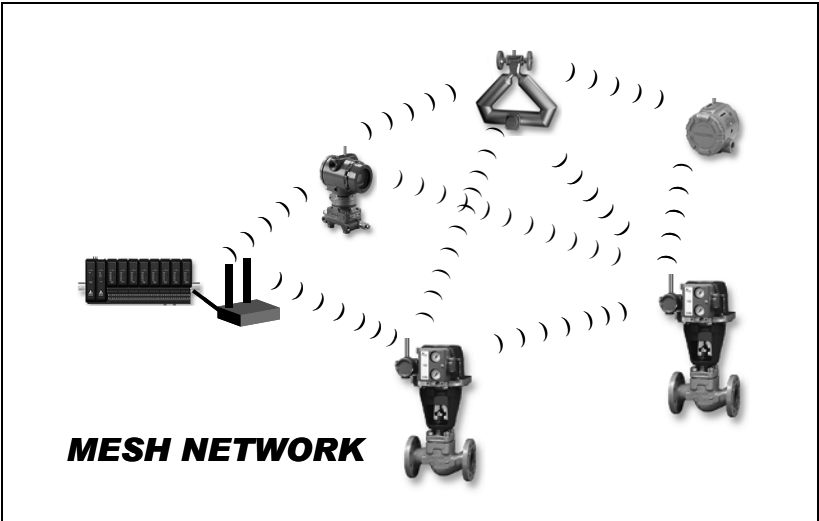


Figure 3-3. Mesh Network Topology

3.1.1 Wireless Sensor Networks (WSN)

Much of the cost of installation of wired networks is for the wire itself. Installation of wire in a plant/factory environment is costly. The natural conclusion is to eliminate the wires by using WSN. The natural topology of a WSN is the mesh, as illustrated in Figure 3-3. Notice that the sensors of a mesh also serve as communications hubs for devices that are out of radio range to reach the gateway or host device. Also note that there can be alternative paths between any two devices. These are some of the advantages of WSN.

Installation of WSN also has costs not associated with wire and fiber optics. The ability to transmit and receive data over a radio (wireless) link is not always going to work with the same degree of certainty as a wired link. Atmospheric conditions such as rain, fog, or snow can dramatically affect transmission of wireless signals. Another problem can be traced to the “canyons of steel,” a term that describes many process plants and even factories. When radio signals bounce off steel equipment, signals reaching remote devices must journey through a longer distance than the direct path. This is called a *multipath signal* and makes the signal taking the longer path arrive out of phase with the direct signal, resulting in a signal cancellation that is often called *fade*. Installation of WSN must account for multipath and the ability of signals to be received.

3.2 Fieldbus Networks

The original fieldbus was standardized by ISA as ANSI/ISA-50.02 beginning in 1992, with the final specification published in 1998. The process control version of this standard was used as the basis of the H1 specification by the Fieldbus Foundation (now called the FieldComm Group). On submission to the IEC for international standards, there was a great deal of controversy and debate, eventually leading to the adoption of seven

more network architecture options in their standard IEC 61158, first published in 2000. This IEC standard defines the term *fieldbus*. In this book, the scope of *fieldbus* includes all industrial networks that are designed for installation in the manufacturing plant or shop floor and in which there is distributed and programmable intelligence at each node of the network. Therefore, this classification includes all of those networks previously classified as *device buses*.

Often, it is necessary to measure the position of an object more precisely than present or absent. The most simple position measurement device is a Linear Variable Differential Transformer (LVDT) or where rotary motion is used, a Rotary Variable Differential Transformer (RVDT). These devices are transformers, which, when biased with an AC current, measure motion accurately as a variable reluctance proportional to position. The reluctance is then converted to a digital value to be used by the controller. PLCs usually report such values as the output of the analog-to-digital (A/D) converter that occupies a register location. Other devices, such as synchros and resolvers, are also used to measure location or rotary motion in a similar manner. Optical encoders supply rotary position as a digital signal based on an encoding wheel that rotates with the shaft of the instrument.

Variables such as temperature, pressure, flow, level, current, voltage, and pH are measured by analog instruments. The term *analog* is still used for such measurements because they represent a *scalar* value, even when the underlying mechanism may be purely digital. Such measurements result in a digital value that eventually will be converted to *engineering units*. Devices that measure such variables are considerably more complex than simple digital discrete points, and they often require many parameters to perform the scaling and filtering of the raw measurement. This leads to a requirement for bidirectional

communications with these sensors. FOUNDATION™ Fieldbus and PROFIBUS-PA were designed for these *smart sensors*. Additionally, many of the networks originally designed for communication of binary discrete sensor data can be adapted to transmit scalar values from scalar sensors and to scalar actuators.

Clearly, a different network is required to transmit digital discrete sensor data from that required to exchange parametric scalar data with smart analog sensors. It was for this task that fieldbus networks were created. The term *fieldbus* is used when a programmed device (microprocessor) is located at the network node, and the capacity exists to control its execution by downloading programs and/or configuration data. Although programs can be permanently stored in ROM, they can also be downloaded and stored for execution. The data for signal processing and perhaps control (called *configuration*) is also downloaded from a host computer. Data sets are then transferred to the host computer on demand, on schedule, or on an exception condition. Communications with a smart sensor are truly a computer-to-computer data exchange across an information network that is called a *fieldbus*.

Smart sensor information is usually transferred in terms of a set of data sampled at the same time, which is called an *atomic data set*. Time coherency is important and cannot ever be accomplished by successive queries. Time synchronization is also important to dynamic control and cannot be achieved without a synchronous network. Dynamic control such as the PID (proportional, integral, derivative) algorithm used in process control, positioning control, robotics, airframe positioning, and motion control is based on atomic data sets sampled at exact uniform time intervals.

Many of the sensors or transmitters used in process control need electrical power for operation. This power previously was delivered on 4–20 mA analog connections and now must be delivered using the fieldbus network or another power source. Often process control field instrumentation is installed in areas having the potential for combustible gases or dusts. These areas are classified as hazardous by the National Electric Code (NEC), NFPA-70,² Article 500. Even when field instrumentation is installed in explosion-proof or inert gas-purged enclosures, electrical power carried on a fieldbus cable may present a hazard if the cable is accidentally broken unless measures are taken to limit the electrical power using intrinsic safety devices called *barriers*. Intrinsic safety barriers installed on the fieldbus cable limit current flow to prevent the generation of a spark when there is sufficient energy to ignite a flammable gas, vapor, or dust mixture. Some data transmission methods are regarded as inherently safe, such as wireless, pneumatics, and fiber-optic sensors when energized by LED sources, not lasers. Because the fieldbuses intended for use in process control were to replace the 4–20 mA that delivered the signal and power on intrinsically safe two-wire instrument cable, they also needed to operate with only two conductors on similar cable. Delivery of power to field sensors with intrinsic safety is one of the most difficult achievements of some fieldbuses.

While wired sensor networks deliver sensing power to the simple devices that they connect, fieldbuses designed for connection of discrete I/O need more controlled power for the microprocessors in the network nodes, as well as for sensing power. Many of these locations in the process industries will also require intrinsic safety specifications, while applications in discrete parts manufacturing, packaging, and assembly line

2. National Fire Protection Association, Quincy, MA, USA, <http://www.nfpa.org>.

control will generally not require intrinsic safety designs. Additionally, although a small number of sensors and actuators may be grouped together at one location, most fieldbuses intended for manufacturing and assembly line control must connect these devices along a machine or transfer line that can be quite long. The number of I/O sensors and actuators for discrete parts manufacturing, packaging, and material handling can often be very large, leading to the use of multiplexing nodes connecting many I/O at one network location.

Use of wireless connections for process data acquisition and control has already begun with some nonstandard and proprietary equipment. ISA100 Wireless (ANSI/ISA-100.11a-2011, IEC 62734) is a standard designed for process control applications, and is currently installed at a number of user locations. WirelessHART (Highway Addressable Remote Transducer, IEC 62591) has been available since 2009 and is used in several process monitoring applications.

Discrete parts manufacturing industry wireless sensor network standards are of great interest and have been created, but have only been used in proprietary networks as of 2015.

3.3 Control Networks

Control level networks are intended to allow control systems to connect with each other, to serve as the path for connection of fieldbuses to control systems, and for control systems to connect to business systems. Because large amounts of data may be passed through these networks and message lengths tend to be longer, data transmission rates tend to be faster than with fieldbus networks. However, since they can be used to pass time-critical data between controllers, control networks must also be deterministic and meet the time-dependent (usually called *real time*) needs of their intended applications.

Determinism in a network context is defined as: there is a specified *worst-case delay* between the sensing of a data item and its delivery to the controlling device. Real time in this context is defined as “sufficiently rapid to achieve the objectives of the application,” and is measured as *latency* time. Determinism and latency are separate but complementary requirements. Both determinism and a specific latency are required to achieve synchronization.

If the same control network is used to exchange both real-time data between controllers and business information between controllers and business systems, clearly there must be some way to prevent business information from interfering with real-time deterministic response. Many complex protocols have been constructed for this purpose, but most control networks rely only on the underlying nature of the chosen network protocol. Usually determinism is achieved by preventing message collisions and limiting the maximum message length. Low latency is achieved by using high-speed media and minimizing the number of times a signal must be rebroadcast, such as in a mesh network.

The benefit of using a standard network protocol such as Transport Control Protocol/Internet Protocol (TCP/IP) over an Ethernet network is a lower cost. By simply selecting standard Ethernet cabling and using full-duplex Ethernet switches instead of passive hubs, a control network built on this commodity technology can guarantee that there will be no network collisions, making such networks deterministic. Using high speed such as 100 or 1000 Mbps and the standard Ethernet maximum packet length of 1,500 bytes achieves low latency and means that other applications cannot “hog the wire,” preventing time-critical data transfers. However, you still must do the math! Remember that the definition of real time and determinism requires that the network must make its bandwidth

available for time-critical data transfers in less than the maximum time period allowed for the control system. For example, if a business application were to transfer a maximum size Ethernet message (1,500 bytes) at 100 Mbps, the network would be blocked for a maximum time of about 150 μ s. Normally this magnitude of delay is perfectly acceptable for both process control and factory automation needs, but it may not be acceptable for motion control or machine control.

It would be nice if control networks and fieldbus networks could not be used for the same applications, but they can. It would also be nice if control networks were always confined to a business or control room environment, but increasingly they are being extended to the field and shop floor. In some cases, control networks are being used in applications normally requiring a fieldbus. In fact, all of the control networks were developed from one or more of the fieldbus networks and use the same application layer and user layer protocols. Since control networks are related to fieldbuses, there will continue to be a very loose dividing line between them.

Wireless control level networks are already in common use, when the popular IEEE 802.11a/b/g/n/ac standard (also known as Wi-Fi and referred to as “wireless Ethernet”) is used. Since most of the popular control level networks are based on Ethernet, substitution of Wi-Fi for any or all segments is easy and does not require any change in the application and user layers. Often substitution of Wi-Fi for any control network segment is made by the user, and will usually work well, but may not be supported by any of the standards for control level networks.

3.4 Safety Buses

When safety interlocks, E-Stop (emergency-stop) buttons, and high-level alarms were hardwired to machinery, pumps, motors, and other devices, we were sure that they would work all of the time. However, when these switching

devices are connected via a sensor network or a fieldbus, we are no longer guaranteed that a change in switch status could activate the safety condition *if there was a network interruption or delay*. This creates a new requirement—safety buses must not only be deterministic, fast enough, and reliable enough, but they must also provide some affirmative action indicating a failure of the safety bus node to communicate. Only then can we use safety devices on sensor networks and fieldbuses.

...safety buses must not only be deterministic, fast enough, and reliable enough, but they must also provide some affirmative action indicating a failure of the safety bus node to communicate.

Any sensor network or fieldbus can be used for safety detection, interlocking, and shutdown if it has a method to *quickly* detect a failure of the network or of the node to which the safety sensor or actuator is connected. *Quickly*, in this context, is defined as fast enough for the safety function. Although hardwired safety interlocks can be thought of as instantaneous, some reasonable time delay between activation of the interlocking signal and the safety actuation device can be specified for a safety bus. Any failure of the network or the safety nodes should then cause an immediate actuation of the safety mechanism to create a *fail-safe* condition. The *fail-safe* mechanism must be at the actuation device, not a command sent via the network.

The safety bus function is often confused with high-reliability networks that are achieved with redundant wiring paths and often redundant network nodes. The word *redundant* in the context of networking means that there are two or more routes between any message sender and the intended recipient. There are many ways to achieve network redundancy including dual redundancy, bidirectional ring topology, resilient ring structures, and grid or mesh network architectures. All of these are to be found in industrial automation networks, but *high reliability alone does not make a network suitable for safety*. Likewise, safety buses do not need to be highly reliable; they need only to cause the desired trip (fail-safe) function rapidly if they fail. When redundant networks are used for safety, they simply make network failure an unlikely source of a safety trip.

Review Questions, Industrial Network Basics

1. What are the layers above the ISO/OSI seven-layer communications stack?
2. What is the primary purpose of a sensor network?
3. What are the four topologies used for networks?
4. What makes a fieldbus network different from a sensor network?
5. What is the primary difference between a fieldbus and a control network?
6. What is the primary action of a safety bus?

3.5 User Layer Protocols

The top of the ISO/OSI seven-layer communications stack illustrated in Figure 3-1 is called the *application layer*, which serves to make all of the network services available to the applications programmer. Additional layers above the applica-

tion layer are often called *user layers* that add functionality by making network services *most often needed by a class of user* more easily available. All popular fieldbuses and control networks offer user layers for their intended users, but sensor networks are so simple that user layers are not always provided or required. User layers are termed this because they were not named by the ISO in its design of the seven-layer OSI stack. In fact, the user layer may itself be composed of several layers. A typical industrial automation network will use two different user layers for its own applications: one for the efficient cyclic data acquisition and control data transfers, and another for the information data transfers needed for operator display and interaction. There are user layers constructed for some vertical industrial applications as well.

3.5.1 Real-Time Data Acquisition and Control

Real-time data acquisition for control purposes will use a deterministic network architecture such as Modbus, FOUNDATION Fieldbus, PROFIBUS, or CIP (common to DeviceNet, ControlNet, and Ethernet Industrial Protocol, or EtherNet/IP). In most cases, these networks use the services of a corresponding network application layer to access data from the underlying network. Use of these real-time layers usually requires that the software used for the application itself is highly aware of the user layer on which it is depending.

Sensors and actuators used for discrete control terminate in a multiplexer of a PLC or a remote I/O unit. The term *multiplexer* is used generically to mean an electronic device to sense the status or position of an input sensor or to change the state of an actuator. The status of inputs and the state of outputs appear in a multiplexer as a register or a word of 16 binary bits where each bit represents an input or output (I/O) point. Process control likewise depends on data from sensors being multiplexed

and sent by the network to process controllers and the resulting output values being sent to actuators. The role of the real-time network is to transfer the status words to the PLC or sensor data to the process controller rapidly enough for it to complete its scan and control cycle and then to transfer the output status words or values back to the remote I/O registers or process actuators, all within a specified time period, typically called the *cycle time* or *scan time*.

All of the sensor networks can meet the requirements of real-time data acquisition and control that are consistent with the timing of their networks and the number of I/O points that they are expected to scan. This is typically measured in milliseconds. Very little can interfere with the operation of a sensor network.

Fieldbuses can also meet the real-time requirements for data acquisition and control as long as the definition of real time is consistent with the timing of the control cycle and the network timing, as well as the number of I/O points to be scanned. However, fieldbuses must be configured so that the PLC or process controller can regulate network access to ensure that all scan and control cycles can be completed within the specified time period, typically measured in tens of milliseconds. This is the requirement for deterministic data transfer that cannot be met by uncontrolled networks such as those used for information access.

All of the control networks can also meet the real-time requirements for data acquisition and control, but usually the timing is much more relaxed than that of a fieldbus or a sensor network, and is measured in seconds. The difference is usually in the protocol of a fieldbus that has its time synchronization at layer 2 (data link) of the network, as opposed to control net-

works that achieve timing at layers 3 and 4 (network and transport layers) of the network.

3.5.2 OPC and Information Access

Many applications in industrial automation are highly generic and intended to operate with a variety of automation systems. Originally, authors of this type of application software were required to create *drivers* for each of the automation systems to which they would connect. OPC fixed that situation by establishing a common interface (OPC *client*) for any application working with automation systems supplying an OPC *server*. Communications would then occur only between the OPC client supplied by the application supplier and the server supplied by the automation supplier, freeing the software supplier from the particularities of the automation system.

OPC was originally designed for operation on the Microsoft Windows[®] operating system. It is founded on the Microsoft Component Object Model (COM) and its network distributed (Distributed Component Object Model or DCOM) equivalent. Using COM, two applications running under Windows on the same PC can pass data to each other as messages between objects. The same applications may also pass messages to each other when they are run on different PCs on the same network using DCOM. COM is based on a common operating system feature called *remote procedure calls* (RPC) that is implemented in Microsoft Windows as its message-passing interface between executing objects and is called *object linking and embedding* (OLE). When the two objects are executing on different computers, the Windows operating system uses the user-defined network protocol to communicate. Microsoft supports TCP/IP as the standard network protocol, but any protocol defined in the Windows Network Connections may be used. If the networked device is not running Windows, it must then

support COM/DCOM or SOAP. (SOAP is originally an acronym for Simple Object Access Protocol, but now it does not mean anything. It is now an open XML-based standard supported by the World Wide Web Consortium, or W3C, a standards body responsible for Internet standards.) Microsoft and many others support SOAP on many other operating systems, including several used for embedded systems appropriate for industrial automation devices. Note that using TCP/IP makes such linkages nondeterministic if the Internet is used because the way that TCP is defined makes message timing subject to random delays of Internet routing. TCP/IP used only on a local network is not subject to random delays, which makes it fully deterministic. However, using TCP may result in an inefficient use of network resources compared with other transport layer protocols based on User Datagram Protocol/Internet Protocol (UDP/IP).

The initial OPC protocol was called DA (Data Access), which used very simple data structures. The data understood by DA is a simple register, 16 bits long. The meaning of the bits is to be defined by the end user during the application. Although DA was sufficient for typical PLC applications, process control functions required additional data structure for control loop and instrumentation attributes or parameters. At first, OPC created eXtensible Markup Language (XML) schemas for defining these data structures as a layer to be implemented on top of OPC/DA. Before OPC/XML could be widely used, it was recognized that a stronger object attribute definition method was needed. This was introduced as OPC/DX (Data eXchange), again built as an additional layer operating above OPC/XML. The purpose of OPC/DX is to allow the definition of data independent of both the control system supplying the server and the data management or presentation system supplying the OPC/DX client.

OPC has now evolved to OPC/UA, or Unified Architecture. For a complete description of OPC/UA, the reader is encouraged to read the information that can be obtained in a web search for OPC/UA. These changes in OPC were made necessary by the facts that support on platforms other than Windows has become necessary, and that the original COM, DCOM, and OLE roots have become obsolete on Windows-based systems. OPC/UA is based on a series of open, IEC, and Internet standards that support distributed object-oriented data transfer such as SOAP and Electronic Device Description Language (EDDL). While the mechanisms of OPC/UA are different from earlier versions, the objectives are the same: to provide a common interface between data sources and the users of that data.

3.5.3 Microsoft .NET Architecture and Java

Microsoft has replaced its entire object modeling system and many other aspects of its architecture with .NET (pronounced dot-net) Framework. The intent of .NET is to establish a single consistent program execution framework independent of programming language. Microsoft calls this the common language runtime (CLR) environment, which is a virtual machine. Microsoft has standardized the .NET Framework specifications through the European Computer Manufacturer's Association (ECMA), an open standards body specifically for computer programming languages. This concept is almost the same as Oracle's (formerly created by Sun Microsystems) Java Virtual Machine (JVM), but it was done 5 years later and has many Windows advantages. Oracle's JVM interprets original program language statements compiled into bytecode, which are the instructions to the JVM. The .NET platform is slightly different—original programming language statements must be written to the .NET class library standards, and then are compiled into the target machine code, with most of the work done

by the class library. Microsoft claims that .NET will run faster on all targeted computers, but Oracle counters that Java runs on any computer supporting the JVM. Microsoft has support for the JVM machine in Windows. In fact, there will be almost no observable differences between .NET and Java. The biggest difference is that .NET supports a smooth migration path from COM/DCOM to the .NET object communications. This means that .NET supports OPC. Microsoft’s version of Java also supports OPC/UA.

The .NET object access method is called SOAP, which supersedes COM. The effect on OPC is to free it from the Microsoft operating system constraint. The OPC/UA specification was based on the use of .NET/SOAP.

3.5.4 Field Device Tool (FDT)

Most of the fieldbuses define attributes of devices used for sensing and control. These attribute definitions are contained in user layer specifications. Table 3-1 lists the names of the specifications for the fieldbuses and some of their properties.

Table 3-1. Fieldbus Attribute Definitions

Fieldbus	Attribute Definitions	Typical Use
FOUNDATION Fieldbus, PROFIBUS-PA, and HART	EDD (Electronic Data Definition)	Process control loop and instrumentation attributes
DeviceNet, ControlNet, and EtherNet/IP	EDS (Electronic Data Sheet)	Defines the attributes of the device

FDT was created to eliminate the need for the user to maintain the different attribute definitions for HART, PROFIBUS-PA, and FOUNDATION Fieldbus. FDT allows the field device supplier to offer a single Device Type Manager (DTM) indepen-

dent of the fieldbus to be used for a project, whereas the host device uses an FDT framework server.

Most process control suppliers seem to be supporting FDT, especially those that also support PROFIBUS-PA. FDT does not have this same level of support in the factory automation world since few PLC suppliers have created FDT Framework servers or DTMs for binary field devices. It seems that FDT and OPC/DX are two very similar methods of solving the same problems.

OPC states that it is a high-level protocol for standardizing host to controller communications. FDT claims to be a high-level protocol for standardizing field device to controller communications. Both are built on Microsoft COM/DCOM and use XML-encoded data frames (in OPC/DX, XML, and UA). OPC states that it is intended for Ethernet-based automation networks, whereas FDT is designed for fieldbus networks. Clearly, as Ethernet-based control networks converge into the domain of fieldbus, there will eventually be a conflict between OPC and FDT/DTM.

Another potential conflict for FDT is the standardization of EDDL by IEC 61804. The developers of FDT state that it was specifically developed to resolve the engineering problem of defining device attributes for smart devices. This is one of the goals of EDDL as well as to provide real-time high-speed data access to device data, which was not a goal for FDT.

The major advantage of FDT over both EDDL and OPC has been to give the field device supplier the ability to construct comprehensive visualization tools for detailed analysis of the data contained in the field device, and especially for use in calibration and diagnostics. These tools, using FDT, are independent of the control system supplier.

In 2007, the FDT Group joined the EDDL Cooperation Team (ECT) with the purpose of working out the differences and avoiding end user confusion over these two approaches. This work has now been completed and is known as FDI (Field Device Integration). The EDDL work has since become ANSI/ISA-61804, while the FDT specifications have become ANSI/ISA-62453. Many of the graphic capabilities pioneered by FDT for display of local instrument data have now been added to EDDL. The FDI specification is now under control by the FieldComm Group but it has not yet been approved as a standard.

3.6 Convergence and Downward Migration

The end user should not care about the underlying network protocol for an automation network, but only about the difficulty and cost of installing the automation system and the support of the network by the suppliers. Most of the work of EDDL, FDT, and OPC has been to build a layer of abstraction above the network protocol to make it transparent to the end user and the system suppliers. However, the end user will still be concerned with installation. Many of the more modern network protocols are directed toward reducing the cost of installation by using commodity network components and wiring.

Eventually, proprietary networks will be fully displaced by industrial networks conforming to international standards, or at least industry-accepted “open” specifications. Most of the differences between networks originally designed for factory automation and those created for process control will begin to disappear except for the highest levels of the user layer. For example, there is practically no difference between the installed cost of FOUNDATION Fieldbus HSE, EtherNet/IP, or PROFINET, each of which uses standard, commercial, off-the-

shelf Ethernet wiring and components. However, their user layers are quite different.

3.6.1 Wireless on the Shop Floor

3.6.1.1 Wi-Fi

The reduced cost of Ethernet-based networks is driving this fast, low-level, and low-cost technology to the field or shop floor. Another Ethernet side effect can be seen in the application of wireless technology in the Wi-Fi group of wireless protocols. Wi-Fi is essentially wireless Ethernet. Any higher-level application layer and user layer can communicate via Wi-Fi at data rates up to about 1 Gbps, without knowledge of the fact that it is on a radio link. Wi-Fi is the most common wireless technology, however, it has significant problems for operation in the electrically noisy environment of a process plant or the shop floor in a manufacturing factory.

The popular Wi-Fi-a/b/g standards (IEEE 802.11a, b, and g) can achieve a theoretical maximum of 54 Mbps using one of the 2.4 GHz channels. However, the Wi-Fi-n standard (IEEE 802.11n), which has a maximum (theoretical) rated specification of 600 Mbps, is even more interesting. Wi-Fi-n allows the bonding of up to four radio channels, including the channels in the 5 GHz band to achieve its high speed. The feature of Wi-Fi-n that is most appealing to industrial use is its adoption of multiple-input multiple-output (MIMO) technology. MIMO has the demonstrated potential to eliminate the adverse effects of reflections that cause multipath distortion appearing as signal fade. MIMO achieves improved reception through detecting the multipath signals and either eliminating them or phase-shifting them to amplify the received signal. Experiments suggest that using Wi-Fi-n can achieve excellent behavior in both process plants and factories notorious for their “canyons of steel,” the cause of poor performance of Wi-Fi-a/b/g.

Of even greater interest is Wi-Fi-ac (IEEE 802.11ac). Both commercial and industrial versions of Wi-Fi-ac devices are already being sold widely. Like Wi-Fi-n, Wi-Fi-ac uses MIMO and channel bonding to increase bandwidth, but exclusively in the 5 GHz band. Up to eight channels can be bonded to achieve the theoretical speed of 1 Gbps. Up to 25 Wi-Fi channels are available in the 5 GHz band. Of course, use of wireless technology brings problems of security and privacy to industrial networks, which was never much of an issue with wired networks.

Wireless technology provides excellent solutions to the problem of the high cost of industrial wiring, and also provides an ultimate barrier to electrical surges introduced to field equipment through field wiring. The cost of these wireless advantages is the difficulty of supplying power to field devices, which were often powered by the same cable used to conduct the data exchange between the field device and a host system. As of mid-2015, several manufacturers were offering wireless field devices conforming to standard specifications that were designed for battery operation. Taking advantage of the low-energy design of wireless devices, a whole new industry is forming to scavenge power (also called *energy harvesting*) from available light, vibration, and thermal differences. It is likely that the potential cost reduction of avoiding wire/cable installation and maintenance will provide an expanding market for wireless sensors in the future. Additionally, new applications for industrial measurements and controls are being found for wireless devices. These applications were often not economical when they required wired connection.

3.6.1.2 ISA100 Wireless

ISA100 Wireless field instruments and signal converters for HART instruments are commercially available. The ISA100 standards committee has developed a comprehensive standard for industrial wireless communications. The first standard is

ANSI/ISA-100.11a-2011, *Wireless Systems for Industrial Automation: Process Control and Related Applications*, intended for process data acquisition and limited control needs in the process industries. The equivalent international standard is IEC 62734. The standard is now supported by the Wireless Compliance Institute (WCI), an operating affiliate of ISA, and is commercially known as ISA100 Wireless.

ISA100 Wireless improves on the IEEE 802.15.4-2006 standard by hopping among the 16 channels in the 2.4 GHz industrial, scientific, and medical (ISM) band. ISA100 Wireless has options to adapt its network for a wide variety of applications including segmentation of the network and peer-to-peer messaging. Each network segment may use a different hopping pattern and its own allocated time slot duration. These choices were made to allow large networks to be formed where segments may overlap. Field routers have also been defined to reduce the number of hops required to reach the host device, and to bridge geographically separated network segments.

ISA100 Wireless includes a very high security method³ for new devices to join the network that depends on an asymmetric encryption 256-bit join key, scheduled 128-bit encryption key rotation, and a required identity in a white list. Mesh networks are formed under direction of a network manager and always include duocast messaging where all messages are transmitted to two network neighbors at the same time providing a dual-path redundancy. These features are designed to make ISA100 Wireless as reliable as wired networks.

ISA100 Wireless mesh networking allows devices at the edge of the network to not route messages for other devices. This can increase security by preventing unauthorized devices from

3. Elliptic curve PKI (public key infrastructure).

accessing plant networks from outside the plant. It can also reduce the cost of devices by making them simpler. Additionally, ISA100 Wireless uses Internet-conforming IP addressing to make data from field devices addressable remotely. The transport layer implements secure end-to-end message delivery and confirmation and is based on the use of Internet-conforming User Datagram Protocol (UDP) messages, but avoids the trial-and-error methods used by TCP.

The application layer of ISA100 Wireless is completely object-oriented, in which data in field devices can be addressed using IEC 61804 standard EDDL protocol. For networks not using this standard, all messages may be encapsulated and tunneled to the requesting host device.

3.6.1.3 *WirelessHART*

WirelessHART process field transmitters are being offered commercially. In addition, a simple device to convert wired HART transmitters and valve positioners to the WirelessHART protocol is available. WirelessHART is specified by the IEC 62591 standard supported by the HART Communication Foundation, now a part of the FieldComm Group.

WirelessHART uses the IEEE 802.15.4-2006 standard modified to hop among the 15 universally available frequencies (channels) specified by that standard in the 2.4 GHz ISM band. The slot time is constant for the entire network, usually 10 ms. Transmission is encrypted using a 128-bit key to achieve a high degree of security. Field devices are all part of a mesh network (see Figure 3-3) with a secure method of building and repairing the mesh defined by its proprietary link layer protocol. The advantages of a mesh network are redundancy, increased total distance, and removal of the line-of-sight restriction.

A simple proprietary transport layer protocol is defined to ensure end-to-end message delivery and confirmation when required. Like wired HART, data access is polled using HART commands, including all maintenance functions of the WirelessHART network. WirelessHART devices may also be set to transmit data using a publishing method. WirelessHART devices are provisioned (network setup) through a wired connection, and they may join an active network only if provisioned for that network.

WirelessHART was developed to provide a simple wireless network for field instrumentation and to enable a wireless method to access diagnostic data in HART instruments installed in the past. It has been estimated that there are more than 25 million of these HART instruments that can only provide their digital diagnostic data to a hand-held terminal, since the control systems to which they are connected cannot access that data over the connecting 4–20 mA wire. WirelessHART is seen as an answer to this problem.

The fact that WirelessHART and ISA100 Wireless are addressed to the same market has been noted by many users and the IEC as well. They are technically similar but very different in detail, causing many users to request that they be merged. The ISA100.12 subcommittee was formed to achieve convergence between WirelessHART and ISA100 Wireless. At the completion of their work in 2013, this committee could not find a path to converge the two standards without obsoleting products already designed and in-service. They noted that gateways can be designed to connect both networks to a common host, when necessary; that the two networks can independently operate in the same area using the collision recovery methods of each network; and manufacturers were free to design devices that can operate on either network by firmware download or run-time selection. The FieldComm Group has

defined their ROM specification to be used by gateways to link to both ISA100 Wireless and WirelessHART.

3.6.1.4 ZigBee

ZigBee is an organization specifying additional higher-layer protocols using the same standard, IEEE 802.15.4, as both ISA100 Wireless and WirelessHART. It was designed to operate on the shop floor and to avoid interference with Wi-Fi. It is also low cost, requires little power, and can transport Ethernet messages. Although ZigBee may operate with a star topology like Wi-Fi, it also allows operation in a mesh network topology as previously illustrated in Figure 3-3.

While a number of industrial products have been designed to use ZigBee, few have addressed the process control market. Most ZigBee applications are for remote meter reading, building automation, and some remote control operations.

As of mid-2015, most wireless applications for factory automation are being designed using Wi-Fi in locations where wired Ethernet has previously been used. There are few applications for wireless factory (discrete) automation sensors. Honeywell makes a line of wireless switches using their *LimitlessTM* brand name, and based on the IEEE 802.15.4 standard, but does not use either ZigBee or ISA100 Wireless technology.

Review Questions, Network Architecture

1. What are the uses for a user layer protocol?
2. Explain determinism. Why is it important for data acquisition and control?
3. What is the purpose of OPC/DX?

4. Why is Microsoft .NET architecture more important for control systems than Java JVM?
5. Field Device Tools are used for information access to what level network?
6. How are WirelessHART and ISA100 Wireless different?