INVITED PAPER

The Emergence of Industrial Control Networks for Manufacturing Control, Diagnostics, and Safety Data

There is wide use of Ethernet for system diagnostics and control, and inclusion of safety features on the same network is being debated; the trend is towards wireless communications.

By James R. Moyne, *Member IEEE*, and Dawn M. Tilbury, *Senior Member IEEE*

ABSTRACT | The most notable trend in manufacturing over the past five years is probably the move towards networks at all levels. At lower levels in the factory infrastructure, networks provide higher reliability, visibility, and diagnosability, and enable capabilities such as distributed control, diagnostics, safety, and device interoperability. At higher levels, networks can leverage internet services to enable factory-wide automated scheduling, control, and diagnostics; improve data storage and visibility; and open the door to e-manufacturing.

This paper explores current trends in the use of networks for distributed, multilevel control, diagnostics, and safety. Network performance characteristics such as delay, delay variability, and determinism are evaluated in the context of networked control applications. This paper also discusses future networking trends in each of these categories and describes the actual application of all three categories of networks on a reconfigurable factory testbed (RFT) at the University of Michigan. Control, diagnostics, and safety systems are all enabled in the RFT utilizing multitier networked technology including DeviceNet, PROFIBUS, OPC, wired and wireless Ethernet, and SafetyBUS p. This paper concludes with a discussion of trends in industrial networking, including the move to wireless for all categories, and the issues that must be addressed to realize these trends.

Manuscript received July 25, 2005; revised September 1, 2006. This work was supported by the National Science Foundation ERC for Reconfigurable Manufacturing Systems under Grant EEC95-92125.

The authors are with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109-2125 USA (e-mail: moyne@umich.edu; tilbury@umich.edu).

Digital Object Identifier: 10.1109/JPROC.2006.887325

KEYWORDS | Diagnostic networks; e-diagnostics; industrial control networks; manufacturing control; network delay characterization; network delay variability; network performance; networked control systems; safety networks

I. INTRODUCTION: TRENDS IN MANUFACTURING NETWORKS

Control networks can replace traditional point-to-point wired systems while providing a number of advantages. Perhaps the simplest but most important advantage is the reduced volume of wiring. Fewer physical potential points of failure, such as connectors and wire harnesses, results in increased reliability. Another significant advantage is that networks enable complex distributed control systems to be realized in both horizontal (e.g., peer-to-peer coordinated control among sensors and actuators) and vertical (e.g., machine to cell to system level control) directions. Other documented advantages of networks include increased capability for troubleshooting and maintenance, enhanced interchangeability and interoperability of devices, and improved reconfigurability of control systems [32].

With the return-on-investment of control networks clear, the pace of adoption continues to quicken, with the primary application being supervisory control and data acquisition (SCADA) systems [36]. These networked SCADA systems often provide a supervisory-level factory-wide solution for coordination of machine and process diagnostics, along with other factory floor and operations information. However, networks are being used at all levels of the manufacturing hierarchy, loosely defined as

device, machine, cell, subsystem, system, factory, and enterprise. Within the manufacturing domain, the application of networks can be further divided into subdomains of "control," "diagnostics," and "safety." Control network operation generally refers to communicating the necessary sensory and actuation information for closed-loop control. The control may be time-critical, such as at a computer numeric controller (CNC) or servo drive level, or event-based, such as at a programmable logic controller (PLC) level. In the control subdomain, networks must guarantee a certain level of response time determinism to be effective. Diagnostics network operation usually refers to the communication of sensory information as necessary to deduce the health of a tool, product, or system; this is differentiated from "network diagnostics" which refers to deducing the health of the network [17], [25], [26], [51]. Systems diagnostics solutions may "close-the-loop" around the diagnostic information to implement control capabilities such as equipment shutdown or continuous process improvement; however, the performance requirements of the system are primarily driven by the data collection, and actuation is usually event based (i.e., not time dependent). An important quality of diagnostics networks is the ability to communicate large amounts of data; determinism is usually less important than in control networks. Issues of data compression and security can also play a large role in diagnostic networks, especially when utilized as a mechanism for communication between user and vendor to support equipment e-diagnostics [10], [25], [51]. Safety is the newest of the three network subdomains but is rapidly receiving attention in industry [35]. Here, network requirements are often driven by standards, with an emphasis on determinism (guaranteed response time), network reliability, and capability for self-diagnosis [22].

Driven by a desire to minimize cost and maximize interoperability and interchangeability, there continues to be a movement to try to consolidate around a single network technology at different levels of control and across different application domains. For example, Ethernet, which was widely regarded as a high level-only communication protocol in the past, is now being utilized as a lower level control network. This has enabled capabilities such as web-based "drill-down" (focused data access) to the sensor level [28], [51]. Also, the debate continues on the consolidation of safety and control on a single network [22].

This movement towards consolidation, and indeed the technical selection of networks for a particular application, revolves around evaluating and balancing quality of service (QoS) parameters. Multiple components (nodes) are vying for a limited network bandwidth, and they must strike a balance with factors related to the time to deliver information end-to-end between components. Two parameters that are often involved in this balance are network average speed and determinism; briefly, network speed is a function of the network access time and bit transfer rate, while determinism is a measure of the ability to communicate data consistently from end to end within a guaranteed time.

Network protocols utilize different approaches to provide end-to-end data delivery. The differentiation could be at the lowest physical level (e.g., wired versus wireless) up through the mechanism at which network access is negotiated, all the way up through application services that are supported. Protocol functionality is commonly described and differentiated utilizing the International Standards Organization—Open Systems Interconnection (ISO-OSI) seven-layer reference model [24]. The seven layers are physical, data link, network, transport, session, presentation, and application.

The network protocol, specifically the media access control (MAC) protocol component, defines the mechanism for delegating this bandwidth in such a way so that the network is "optimized" for a specific type of communication (e.g., high data packet size with low determinism versus small data packet size with high determinism). Over the past decade "bus wars" (referring to sensor bus network technology) have resulted in serious technical debates with respect to the optimal MAC approach for different applications [15], [39].

Over the past five years, however, it has become more and more evident that the pervasiveness of Ethernet, especially in domains outside of manufacturing control (e.g., the internet), will result in its eventual dominance in the manufacturing control domain [6], [14], [45]. This movement has been facilitated in large part by the emergence of switch technology in Ethernet networks, which can increase determinism [38]. While it is not clear yet whether or not Ethernet is a candidate for safety networking, it is a strong contender in the control subdomain and has achieved dominance in diagnostics networking [36].

The body of research around control networks is very deep and diverse. Networks present challenges of timing in control systems but also opportunities for new control directions enabled by the distribution capabilities of control networks. For example, there has been a significant amount of recent work on networked control systems [2], [11]. Despite this rich body of work, one important aspect of control networks remains relatively untouched in the research community: the speed of the devices on the network. Practical application of control networks often reveals that device speeds dominate in determining system performance to the point that the speed and determinism (network QoS parameters) of the network protocol are irrelevant [31], [38] [46]. Unfortunately, the academic focus on networks in the analysis of control network systems, often with assumptions of zero delay of devices, has served to further hide the fact that device speed is often the determining factor in assessing networked control system performance.

This paper explores the emergence of industrial networks for control, diagnostics, and safety in manufacturing. Specifically, the parameterization of networks with respect to balancing QoS capabilities is explored in Section II. This parameterization provides a basis for differentiating industrial network types, which is provided in Section III; here, common network protocol approaches are introduced and then differentiated with respect to functional characteristics. The impact of device performance is also identified. In Section IV, network applications within the domain of manufacturing are explored; these include application subdomains of control, diagnostics, and safety, as well as different levels of control in the factory such as machine level, cell level, and system level. An example of a multilevel factory networking solution that supports networked control, diagnostics, and safety is provided in Section V. This paper concludes with a discussion of future trends in industrial networks with a focus on the move to wireless networking technology.

II. PARAMETERIZATION OF INDUSTRIAL NETWORKS: BALANCING QoS CAPABILITIES

The function of a network is to transmit data from one node to another. Different types of industrial networks use different mechanisms for allocating the bandwidth on the network to individual nodes and for resolving contentions among nodes. Briefly, there are three common mechanisms for allocating bandwidth: time-division multiplexing, random-access with collision detection, and random-access with collision avoidance. In time-division multiplexing, the access time to the network is allocated in a round-robin fashion among the nodes, either by passing a token (e.g., ControlNet) or having a master poll the slaves (e.g., AS-I). Because the bandwidth is carefully allocated, no collisions will occur. If random access to the network is allowed, collisions can occur if two nodes try to access the network at the same time. The collision can be destructive or nondestructive. With a destructive collision, the data is corrupted and both nodes must retransmit (e.g., Ethernet). With a nondestructive collision, one node keeps transmitting and the other backs off (e.g., CAN); in this case, the data is not corrupted. Collision avoidance mechanisms (e.g., WiFi) use random delay times to minimize the probability that two nodes will try to transmit at the same time, but collisions can still occur. These mechanisms and the most common network protocols that use them will be discussed in more detail in Section III.

Although any network protocol can be used to send data, each network protocol has its pros and cons. In addition to the protocol, the type and amount of data to be transmitted is also important when analyzing network performance: will the network carry many small packets of data frequently or large packets of data infrequently? Must the data arrive before a given deadline? How many nodes will be competing for the bandwidth, and how will the contention be handled?

Unfortunately, the academic focus on networks in the analysis of control network systems... has served to further hide the fact that device speed is often the determining factor in assessing networked control system performance.

The QoS of a network is a multidimensional parameterized measure of how well the network performs this function; the parameter measures include the speed and bandwidth of a network (how much data can be transmitted in a time interval), the delay and jitter associated with data transmission (time for a message to reach its destination, and repeatability of this time), and the reliability and security of the network infrastructure [54]. When using networks for control, it is often important to assess determinism as a QoS parameter, specifically evaluating whether message end-to-end communication times can be predicted exactly or approximately, and whether these times can be bounded.

In this section, we will review the basic QoS measures of industrial networks, with a focus on time delays, since they are typically the most important element determining the capabilities of an industrial control system. In Section III, more detailed analysis of the delays for specific networks will be given. The section concludes with a brief discussion of QoS of networked systems as it relates to the QoS of the associated enabling network technology.

A. Speed and Bandwidth

The *bandwidth* of an industrial network is given in terms of the number of bits that can be transmitted per second. Industrial networks vary widely in bandwidth, including CAN-based networks, which have a maximum data rate of 1 Mb/s, and Ethernet-based networks, which can support data rates up to 1 Gb/s[1]; although, most networks currently used in the manufacturing industry are based on 10- and 100-Mb/s Ethernet. DeviceNet, a commonly used network in the manufacturing industry, is based on CAN and has a maximum data rate of 500 kb/s. The *speed* is the inverse of the data rate, thus the time to

[1]10-Gb/s solutions are available with fiber optic cabling.

transmit 1 bit of data over the network is $T_{bit} = 1\ \mu s$ for 1-Mb/s CAN and 100 ns for 10-Mb/s Ethernet.

The data rate of a network must be considered together with the packet size and overhead. Data is not just sent across the network one bit at a time. Instead, data is encapsulated into packets, with headers specifying the source and destination addresses of the packet, and often a checksum for detecting transmission errors. All industrial networks have a minimum packet size, ranging from 47 bits for CAN to 84 bytes for Ethernet. A minimum "interframe time" between two packets is required between subsequent messages to ensure that each packet can be distinguished individually; this time is specified by the network protocol.

The transmission time for a message on the network can be computed from the network's data rate, the message size, and the distance between two nodes. Since most of these quantities can be computed exactly (or approximated closely), transmission time is considered a deterministic parameter in a network system. The transmission time can be written as the sum of the frame time and the propagation time

$$T_{tx} = T_{\text{frame}} + T_{\text{prop}}$$

where T_{frame} is the time required to send the packet across the network and T_{prop} is the time for a message to propagate between any two devices. Since the typical transmission speed in a communication medium is 2×10^8 m/s, the propagation time T_{prop} is negligible on a small scale. In the worst case, the propagation delays from one end to the other of the network cable for two typical control networks are $T_{\text{prop}} = 67.2\ \mu s$ for 2500-m Ethernet,[2] and $T_{\text{prop}} = 1\ \mu s$ for 100-m CAN. The propagation delay is not easily characterized because the distance between the source and destination nodes is not constant among different transmissions, but typically it is less than 1 μs (if the devices are less than 100 m apart). Some networks (e.g., Ethernet) are not a single trunk but have multiple links connected by hubs, switches, and/or routers that receive, store, and forward packets from one link to another. The delays associated with these interconnections can dominate propagation delays in a complex network and must also be considered when determining transmission delays [40].

The frame time T_{frame} depends on the size of the data, the overhead, any padding, and the bit time. Let N_{data} be the size of the data in terms of bytes, N_{ovhd} be the number of bytes used as overhead, N_{pad} be the number of bytes used to pad the remaining part of the frame to meet the minimum frame size requirement, and N_{stuff} be the number of bytes used in a stuffing mechanism (on some protocols).[3] The frame time can then be expressed by the following:

$$T_{\text{frame}} = [N_{\text{data}} + N_{\text{ovhd}} + N_{\text{pad}} + N_{\text{stuff}}] \times 8 \times T_{\text{bit}}. \quad (1)$$

In [29], these values are explicitly described for Ethernet, ControlNet, and DeviceNet protocols.

The effective bandwidth of a control network will depend not only on the physical bandwidth but also on the efficiency of encoding the data into packets (how much overhead is needed in terms of addressing and padding), how efficiently the network operates in terms of (long or short) interframe times, and whether network time is wasted due to message collisions. For example, to send one bit of data over a 500-kb/s CAN network, a 48-bit message is needed, requiring 94 μs. To send the same one bit of data over 10-Mb/s Ethernet, an 84-byte message is needed (64-byte frame size plus 20 bytes for interframe separation), requiring a 67.2 μs T_{frame}. Thus, even though the raw network speed is 20 times faster for Ethernet, the frame time is only 30% lower than CAN. This example shows that the network speed is only one factor that must be considered when computing the effective bandwidth of a network.

B. Delay and Jitter

The *time delay* on a network is the total time between the data being available at the source node (e.g., sampled from the environment or computed at the controller) and it being available at the destination node (received and decoded, where the decode level depends on where the delay is evaluated within the end-to-end communication). The *jitter* is the variability in the delay. Many control techniques have been developed for systems with constant time delays [8], [50], [59], but variable time delays can be much more difficult to compensate for, especially if the variability is large. Although time delay is an important factor to consider for control systems implemented over industrial networks, it has not been well defined or studied by standards organizations defining network protocols [56].

In order to further explain the different components that go into the time delay and jitter on a network, consider the timing diagram in Fig. 1 showing how messages are sent across a network. The source node A captures (or computes) the data of interest. There is some preprocessing that must be done to encapsulate the data into a message packet and encode it for sending over the network; this time is denoted T_{pre}. If the network is busy, the node may need to wait for some time T_{wait} for the

[2]Because Ethernet uses Manchester biphase encoding, two bits are transmitted on the network for every bit of data.

[3]The bit-stuffing mechanism in DeviceNet is as follows: if more than five bits in a row are "1," then a "0" is added and vice versa. Ethernet uses Manchester biphase encoding, and, therefore, does not require bit stuffing.

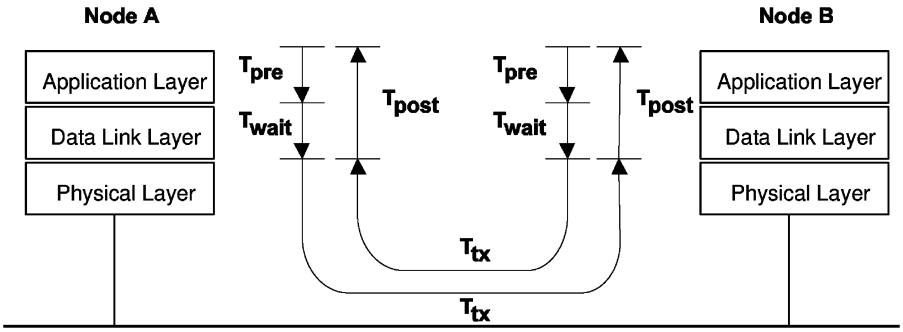

Fig. 1. *Timing diagram showing time spent sending a message from source node to destination node.*

network to become available. This waiting time is a function of the Media Access Control (MAC) mechanism of the protocol, which is categorized as part of layer 2 of the OSI model. Then, the message is sent across the network, taking time T_{tx} as described in Section II-A. Finally, when the message is received at the destination node B, it must be decoded and post-processed, taking time T_{post}. Thus, the total time delay can be expressed by the following:

$$T_{\text{delay}} = T_{\text{pre}} + T_{\text{wait}} + T_{tx} + T_{\text{post}}. \tag{2}$$

The waiting time T_{wait} can be computed based on the network traffic, how many nodes there are, the relative priority of these nodes and the messages they are sending, and how much data they send. The pre- and postprocessing times T_{pre} and T_{post} depend on the devices. Often, the network encoding and decoding are implemented in software or firmware. These times are rarely given as part of device specifications. Since they can be the major sources of delay and jitter in a network, a more detailed discussion of these delays is given here.

1) Pre- and Postprocessing Times: The preprocessing time at the source node depends on the device software and hardware characteristics. In many cases, it is assumed that the preprocessing time is constant or negligible. However, this assumption is not true, in general; in fact, there may be noticeable differences in processing time characteristics between similar devices, and these delays may be significant. The postprocessing time at the destination node is the time taken to decode the network data into the physical data format and output it to the external environment.

In practical applications, it is very difficult to identify each individual timing component. However, a very straightforward experiment can be run with two nodes on the network. The source node A repeatedly requests data from a destination node B and waits until it receives a response before sending another request. Because there are only two nodes on the network, there is never any contention, and thus the waiting time is zero. The request-response frequency is set low enough that no messages are queued up at the sender's buffer. The message traffic on the network is monitored, and each message is time stamped. The processing time of each request-response pair, i.e., $T_{\text{post}} + T_{\text{pre}}$, can be computed by subtracting the transmission time from the time difference between the request and response messages. Because the time stamps are recorded all at the same location, the problem of time synchronization across the network is avoided.

Fig. 2 shows the experimentally determined device delays for DeviceNet devices in a poll configuration; delays for strobe connections show similar trends [38]. Note that for all devices, the mean delay is significantly longer than the minimum frame time in DeviceNet (94 μs), and the jitter is often significant. The uniform distribution of processing time at some of the devices is due to the fact that they have an internal sampling time which is mismatched with the request frequency. Hence, the processing time recorded here is the sum of the actual processing time and the waiting time inside the device. The tested devices include photoeyes, input–output terminal blocks, mass flow controllers, and other commercially available DeviceNet devices.

A key point that can be taken from the data presented in Fig. 2 is that the device processing time can be substantial in the overall calculation of T_{delay}. In fact, this delay often dominates over network delays. Thus, when designing industrial network systems to be used for control, device delay and delay variability should be considered as important factors when choosing components. In the same manner, controller devices such as off-the-shelf PLCs typically specify scan times and interscan delays on the order of a few milliseconds, thus these controller delays can also dominate over network delays.

2) Waiting Time at Source Nodes: A message may spend time waiting in the queue at the sender's buffer and could

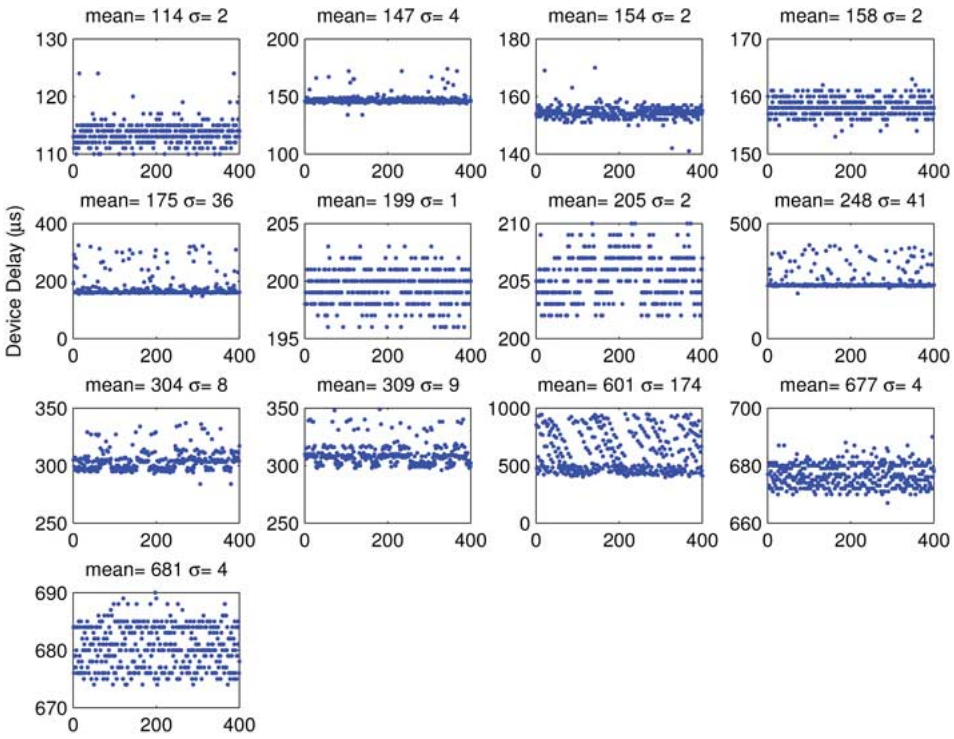

Fig. 2. ***Device delays for DeviceNet devices in poll configuration. Delays are measured with only source and destination node communicating on the network and thus focus only on device delay jitter as described in Section II-B1. Stratification of delay times seen in some nodes is due to the fact that the smallest time that can be recorded is 1 μs.***

be blocked from transmitting by other messages on the network. Depending on the amount of data the source node must send and the traffic on the network, the waiting time may be significant. The main factors affecting waiting time are network protocol, message connection type, and network traffic load.

For control network operation, the message connection type must be specified. In a master–slave network,[4] there are three types of message connections: strobe, poll, and change of state (COS)/cyclic. In a *strobe* connection, the master device broadcasts a message to a group of devices and these devices respond with their current condition. In this case, all devices are considered to sample new information at the same time. In a *poll* connection, the master sends individual messages to the polled devices and requests update information from them. Devices only respond with new signals after they have received a poll message. *COS/cyclic* devices send out messages either when their status is changed (COS) or periodically (cyclic). Although the COS/cyclic connection seems most appropriate from a traditional control systems point of view, strobe and poll are commonly used in industrial control networks [7].

As an example of waiting time in a master–slave network, consider the strobe message connection in Fig. 3. If Slave 1 is sending a message, the other eight devices must wait until the network medium is free. In a CAN-based DeviceNet network, it can be expected that Slave 9 will encounter the most waiting time because it has a lower priority on this priority-based network. However, in any network, there will be a nontrivial waiting time after a strobe, depending on the number of devices that will respond to the strobe.

The waiting time, which is the time a message must wait once a node is ready to send it, depends on the network protocol and is a major factor in the determinism and performance of a control network; it will be discussed in more detail for different types of industrial networks in Section III.

Fig. 4 shows experimental data of the waiting time of nine identical devices with a strobed message connection on a DeviceNet network; 200 pairs of messages (request and response) were collected. Each symbol denotes the mean, and the distance between the upper and lower bars equals two standard deviations. If these bars are over the limit (maximum or minimum), then the value of the limit is used instead. It can be seen in Fig. 4 that the average waiting time is proportional to the node number

[4]In this context, a master–slave network refers to operation from an end-to-end application layer perspective. Master node applications govern the method by which information is communicated to and from their slave node applications. Note that, as will be described further in Section III, application-layer master–slave behavior does not necessarily require corresponding master–slave behavior at the MAC layer.

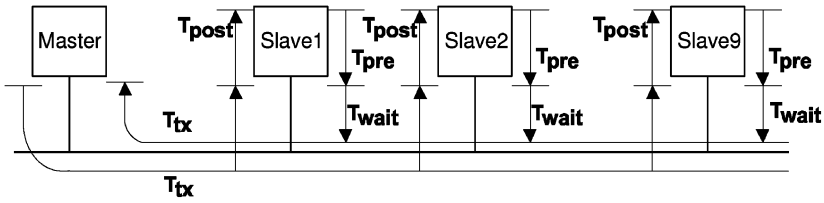

Fig. 3. *Waiting time diagram.*

(i.e., priority). Although all these devices have a very low variance of processing time, the devices with the lowest node numbers have a larger variance of waiting time than the others, because the variance of processing time occasionally allows a lower priority device to access the idle network before a higher priority one.

C. Other QoS Metrics

There are many other metrics that can be used to describe the QoS of a network [54]. Reliability of data transmission is one important factor to consider. Some networks are physically more vulnerable than others to data corruption by electromagnetic interference. Some networks use handshaking by sending of acknowledgment messages to increase the reliability. If no acknowledgment message is received, the message is resent. These handshaking techniques increase the reliability of a network but also add to the required overhead and thus decrease the overall effective bandwidth.

Security is another factor that must be considered, especially when networks and operating systems are used that can be vulnerable to internet-based attacks and viruses [10]. Most industrial fieldbuses were not designed to be highly secure, relying mainly on physical isolation of the network instead of authentication or encryption techniques. When some type of security is provided, the intent is more commonly to prevent accidental misuse of process data than to thwart malicious network attacks [57].

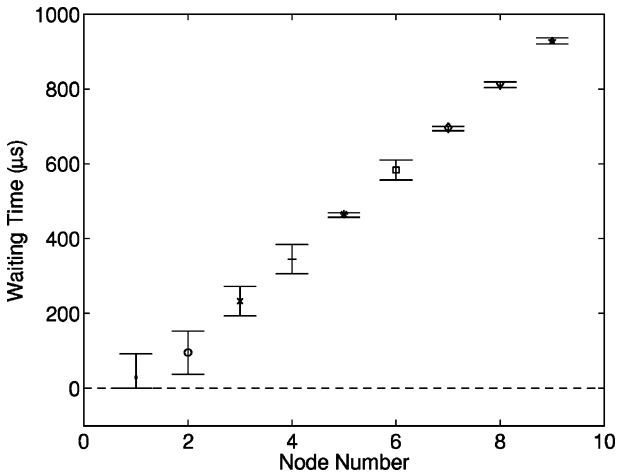

Fig. 4. *Nine identical devices with strobed message connection.*

D. Network QoS Versus System Performance

When a network is used in the feedback loop of a control system, the performance of the system depends not only on the QoS of the network but also on how the network is used (e.g., sample time, message scheduling, node prioritization, etc.) [31], [33]. For example, consider a continuous-time control system that will be implemented with networked communication. Fig. 5 shows how the control performance varies versus sampling period in the cases of continuous control, digital control, and networked control. The performance of the continuous control system is independent of the sample time (for a fixed control law). The performance of the digital control system approaches the performance of the continuous time system as the sampling frequency increases [19]. In a networked control system, the performance is worse than the digital control system at low frequencies, due to the extra delay associated with the network (as described in Section II-B). Also, as the sampling frequency increases, the network starts to become saturated, data packets are lost or delayed, and the control performance rapidly degrades. Between these two extremes lies a "sweet spot" where the sample period is optimized to the control and networking environment.

Typical performance criteria for feedback control systems include overshoot to a step reference, steady-state tracking error, phase margin, or time-averaged tracking error [18]. The performance criteria in Fig. 5 can be one of these or a combination of them. Due to the interaction of the network and control requirements, the selection of the best sampling period is a compromise. More details on the performance computation and analysis of points A, B, and C in Fig. 5 can be found in [31], including simulation and experimental results that validate the overall shape of the chart.

III. DIFFERENTIATING INDUSTRIAL NETWORKS

Networks can be differentiated either by their protocol (at any or all levels of the ISO-OSI seven-layer reference

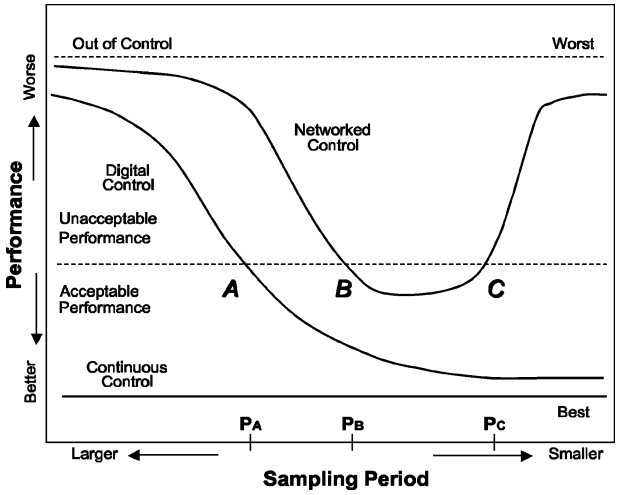

Fig. 5. *Performance comparison of continuous control, digital control, and networked control, as a function of sampling frequency.*

model [24]) or by their primary function (control, diagnostics, and safety). These dimensions of differentiation are somewhat related. In this section, we first define how network protocols are categorized technically with respect to timing and then discuss the different types of protocols that are commonly used in industrial networks. In Section IV, we describe how these different types of networks are used for different functions.

A. Categorization of Networks

When evaluating network QoS parameters associated with timeliness, determinism, etc., the protocol functionality at the data link layer is the primary differentiator among network protocol types. Specifically, the MAC sublayer protocol within the data link layer describes the protocol for obtaining access to the network. The MAC sublayer thus is responsible for satisfying the time-critical/real-time response requirement over the network and for the quality and reliability of the communication between network nodes [27]. The discussion, categorization, and comparison in this section thus focus on the MAC sublayer protocols.

There are three main types of medium access control used in control networks: time-division multiplexing (such as master–slave or token-passing), random access with retransmission when collisions occur (e.g., Ethernet and most wireless mechanisms), and random access with prioritization for collision arbitration (e.g., CAN). Implementations can be hybrids of these types; for example, switched Ethernet combines TDM and random access. Note that, regardless of the MAC mechanism, most network protocols support some form of master–slave communication at the application level; however, this appearance of TDM at the application level does not necessarily imply the same type of parallel operation at the MAC level. Within each of these three MAC categories, there are numerous network protocols that have been defined and used.

A survey of the types of control networks used in industry shows a wide variety of networks in use; see Table 1 and also [20], [32], and [56]. The networks are classified according to type: random access (RA) with collision detection (CD), collision avoidance (CA), or arbitration on message priority (AMP); or time-division multiplexed (TDM) using token-passing (TP) or master–slave (MS).

B. Time-Division Multiplexing (TDM)

Time-division multiplexing can be accomplished in one of two ways: master–slave or token passing. In a master–slave network, a single master polls multiple slaves. Slaves can only send data over the network when requested by the master; there are no collisions, since the data transmissions are carefully scheduled by the master. A token-passing network has multiple masters, or peers. The token bus protocol (e.g., IEEE 802.4) allows a linear, multidrop, tree-shaped, or segmented topology [60]. The node that currently has the token is allowed to send data. When it is finished sending data, or the maximum token holding time has expired, it "passes" the token to the next logical node on the network. If a node has no message to send, it just passes the token to the successor node. The physical location of the successor is not important because the token is sent to the logical neighbor. Collision of data frames does

Table 1 Most Popular Fieldbuses [20], [36]. Maximum Speed Depends on the Physical Layer, Not Application-Level Protocol. Note That Totals are More Than 100% Because Most Companies Use More Than One Type of Bus

Network	Type	Users	Max. Speed	Max. Devices
Ethernet TCP/IP	RA/CD	78%	1 Gb/s	1024
Modbus	TDM/MS	48%	35 Mb/s	32
DeviceNet	RA/AMP	47%	500 kb/s	64
ControlNet	TDM/TP	39%	5 Mb/s	99
WiFi (IEEE 802.11b)	RA/CA	35%	11 Mb/s	??
Modbus TCP	TDM/MS	34%	1 Gb/s	256
PROFIBUS-DP	TDM/MS and TP	27%	12 Mb/s	127
AS-I	TDM/MS	17%	167 kb/s	31

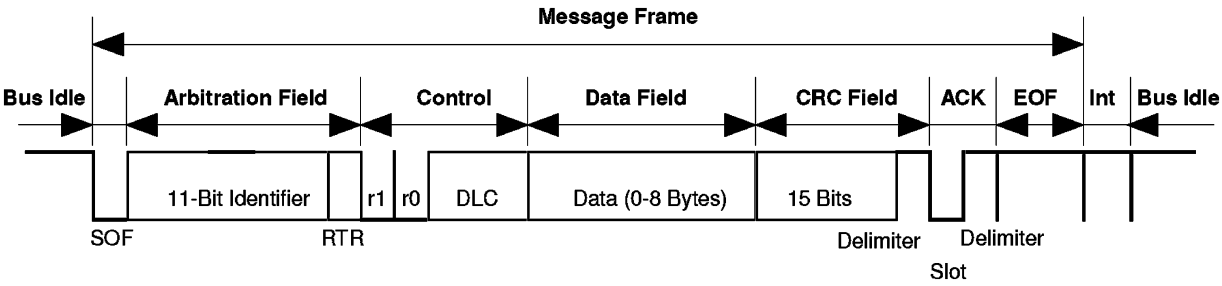

Fig. 6. ***Message frame format of DeviceNet (standard CAN format).***

not occur, as only one node can transmit at a time. Most token-passing protocols guarantee a maximum time between network accesses for each node, and most also have provisions to regenerate the token if the token holder stops transmitting and does not pass the token to its successor. AS-I, Modbus, and Interbus-S are typical examples of master–slave networks, while PROFIBUS and ControlNet are typical examples of token-passing networks. Each peer node in a PROFIBUS network can also behave like a master and communicate with a set of slave nodes during the time it holds the token [48].

Token-passing networks are deterministic because the maximum waiting time before sending a message frame can be characterized by the token rotation time. At high utilizations, token-passing networks are very efficient and fair. There is no time wasted on collisions, and no single node can monopolize the network. At low utilizations, they are inefficient due to the overhead associated with the token-passing protocol. Nodes without any data to transmit must still receive and pass the token.

Waiting time in a TDM network can be determined explicitly once the protocol and the traffic to be sent on the network are known. For token-passing networks, the node with data to send must first wait to receive the token. The time it needs to wait can be computed by adding up the transmission times for all of the messages on nodes ahead of it in the logical ring. For example, in ControlNet, each node holds the token for a minimum of 22.4 μs and a maximum of 827.2 μs.

In master–slave networks, the master typically polls all slaves every cycle time. Slaves cannot transmit data until they are polled. After they are polled, there is no contention for the network so the waiting time is zero. If new data is available at a slave (e.g., a limit switch trips), the slave must wait until it is polled before it can transmit its information. In many master–slave networks (such as AS-Interface), the master will only wait for a response from a slave until a timer has expired. If the slave does not respond within the timeout value for several consecutive polls, it is assumed to have dropped off the network. Also, every cycle time, the master attempts to poll an inactive slave node (in a round-robin fashion) [3]. In this way, new slaves can be added to the network and will be eventually noticed by the master.

C. Random Access With Collision Arbitration: CAN

CAN is a serial communication protocol developed mainly for applications in the automotive industry but also capable of offering good performance in other time-critical industrial applications. The CAN protocol is optimized for short messages and uses a CSMA/arbitration on message priority (AMP) medium access method. Thus, the protocol is message oriented, and each message has a specific priority that is used to arbitrate access to the bus in case of simultaneous transmission. The bit stream of a transmission is synchronized on the start bit, and the arbitration is performed on the following message identifier, in which a logic zero is dominant over a logic one. A node that wants to transmit a message waits until the bus is free and then starts to send the identifier of its message bit by bit. Conflicts for access to the bus are solved during transmission by an arbitration process at the bit level of the arbitration field, which is the initial part of each frame. Hence, if two devices want to send messages at the same time, they first continue to send the message frames and then listen to the network. If one of them receives a bit different from the one it sends out, it loses the right to continue to send its message, and the other wins the arbitration. With this method, an ongoing transmission is never corrupted, and collisions are nondestructive [29].

DeviceNet is an example of a technology based on the CAN specification that has received considerable acceptance in device-level manufacturing applications. The DeviceNet specification is based on the standard CAN with an additional application and physical layer specification [7], [47].

The frame format of DeviceNet is shown in Fig. 6 [7]. The total overhead is 47 bits, which includes start of frame (SOF), arbitration (11-bit identifier), control, CRC, acknowledgment (ACK), end of frame (EOF), and intermission (INT) fields. The size of a data field is between 0 and 8 bytes. The DeviceNet protocol uses the arbitration field to provide source and destination addressing as well as message prioritization.

The major disadvantage of CAN compared with the other networks is the slow data rate, limited by the network length. Because of the bit synchronization, the same data must appear at both ends of the network simultaneously. DeviceNet has a maximum data rate of 500 kb/s for a network of 100 m. Thus, the throughput is limited compared with other control networks. CAN is also not suitable for transmission of messages of large data sizes, although it does support fragmentation of data that is more than 8 bytes into multiple messages.

D. Ethernet-Based Networks

The proliferation of the internet has led to the pervasiveness of Ethernet in both homes and businesses. Because of its low cost, widespread availability, and high communication rate, Ethernet has been proposed as the ideal network for industrial automation [6], [45]. Some question whether Ethernet will become the *de facto* standard for automation networks, making all other solutions obsolete [16], [53]. However, standard Ethernet (IEEE 802.3) is not a deterministic protocol, and network QoS cannot be guaranteed [6], [29]. Collisions can occur on the network, and messages must be retransmitted after random amounts of time. To address this inherent nondeterminism, many different "flavors" of Ethernet have been proposed for use in industrial automation. Several of these add layers on top of standard Ethernet or on top of the TCP/IP protocol suite to enable the behavior of Ethernet to be more deterministic [14]. In this way, the network solutions may no longer be "Ethernet" other than at the physical layer; they may use the same hardware but are not interoperable. As noted in [32], message transmission does not always lead to successful communication: "just because you can make a telephone ring in Shanghai does not mean you can speak Mandarin." A more effective and accepted solution in recent years has been the utilization of switches to manage the Ethernet bandwidth providing a TDM approach among time critical nodes. Rather than repeat the survey of current approaches to industrial Ethernet in [14], in this section, the general MAC protocol of Ethernet is outlined, and the general approaches that are used with Ethernet for industrial purposes are discussed. "Wireless ethernet" (IEEE 802.11) is included in this section because it shares many of the same properties as wired Ethernet, even though it is based on a different standard.

Ethernet is a random access network, also often referred to as carrier sense multiple access (CSMA). Each node listens to the network and can start transmitting at any time that the network is free. Typically, once the network is clear, a node must wait for a specified amount of time (the interframe time) before sending a message. To reduce collisions on the network, nodes wait an additional random amount of time called the backoff time before they start transmitting. Some types of messages (e.g., MAC layer acknowledgments) may be sent after a shorter interframe time. Priorities can be implemented by allowing for shorter interframe times for higher priority traffic. However, if two nodes start sending messages at the exact same time (or if the second node starts transmitting before the first message arrives at the second node), there will be a collision on the network. Collisions in Ethernet are destructive; the data is corrupted and the messages must be resent.

There are three common flavors of Ethernet: 1) hub-based Ethernet, which is common in office environments and is the most widely implemented form of Ethernet; 2) switched Ethernet, which is more common in manufacturing and control environments; and 3) wireless Ethernet. Each of these is discussed in more detail.

1) Hub-Based Ethernet (CSMA/CD): Hub-based Ethernet uses hub(s) to interconnect the devices on a network; this type of Ethernet is common in the office environment. When a packet comes into one hub interface, the hub simply broadcasts the packet to all other hub interfaces. Hence, all of the devices on the same network receive the same packet simultaneously, and message collisions are possible. Collisions are dealt with utilizing the CSMA/CD protocol as specified in the IEEE 802.3 network standard [4], [5], [55].

This protocol operates as follows: when a node wants to transmit, it listens to the network. If the network is busy, the node waits until the network is idle; otherwise, it can transmit immediately (assuming an interframe delay has elapsed since the last message on the network). If two or more nodes listen to the idle network and decide to transmit simultaneously, the messages of these transmitting nodes collide and the messages are corrupted. While transmitting, a node must also listen to detect a message collision. On detecting a collision between two or more messages, a transmitting node transmits 32 jam bits and waits a random length of time to retry its transmission. This random time is determined by the standard binary exponential backoff (BEB) algorithm: the retransmission time is randomly chosen between 0 and (2^i) slot times, where i denotes the ith collision event detected by the node and one slot time is the minimum time needed for a round-trip transmission. However, after ten collisions have been reached, the interval is fixed at a maximum of 1023 slots. After 16 collisions, the node stops attempting to transmit and reports failure back to the node microprocessor. Further recovery may be attempted in higher layers [55].

The Ethernet frame format is shown in Fig. 7 [55]. The total overhead is 26 $(= 22 + 4)$ bytes. The data packet frame size is between 46 and 1500 bytes. There is a nonzero minimum data size requirement because the standard states that valid frames must be at least 64 bytes long, from destination address to checksum (72 bytes including preamble and start of delimiter). If the data portion of a frame is less than 46 bytes, the pad field is used to fill out

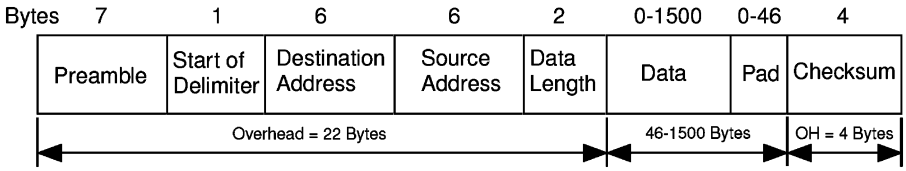

Fig. 7. ***Ethernet (CSMA/CD) frame format; 20 byte interframe space not shown.***

the frame to the minimum size. There are two reasons for this minimum size limitation. First, it makes it easier to distinguish valid frames from "garbage." Because of frame truncation, stray bits and pieces of frames frequently appear on the cable. Second, it prevents a node from completing the transmission of a short frame before the first bit has reached the far end of the cable, where it may collide with another frame. For a 10-Mb/s Ethernet with a maximum length of 2500 m and four repeaters, the minimum allowed frame time or slot time is 51.2 μs, which is the time required to transmit 64 bytes at 10 Mb/s [55].

Because of low medium access overhead, Ethernet uses a simple algorithm for operation of the network and has almost no delay at low network loads [60]. No communication bandwidth is used to gain access to the network compared with token passing protocols. However, Ethernet is a nondeterministic protocol and does not support any message prioritization. At high network loads, message collisions are a major problem because they greatly affect data throughput and time delays may become unbounded [60]. The Ethernet "capture" effect existing in the standard BEB algorithm, in which a node transmits packets exclusively for a prolonged time despite other nodes waiting for medium access, causes unfairness and substantial performance degradation [49]. Based on the BEB algorithm, a message may be discarded after a series of collisions; therefore, end-to-end communication is not guaranteed. Because of the required minimum valid frame size, Ethernet uses a large message size to transmit a small amount of data.

Several solutions have been proposed for using this form of Ethernet in control applications [6]. For example, every message could be time stamped before it is sent. This requires clock synchronization, however, which has not traditionally been easy to accomplish [12]; although, the IEEE 1588 standard has recently emerged to enable clock synchronization on LANs [23]. Various schemes based on deterministic retransmission delays for the collided packets of a CSMA/CD protocol result in an upper-bounded delay for all the transmitted packets. However, this is achieved at the expense of inferior performance to CSMA/CD at low to moderate channel utilization in terms of delay throughput [27]. Other solutions also try to prioritize CSMA/CD (e.g., LonWorks) to improve the response time of critical packets [39]. To a large extent, these solutions have been rendered moot with the proliferation of switched Ethernet as described in the following. On the other hand, many of the same issues reappear with the migration to wireless Ethernet for control.

2) Switched Ethernet (CSMA/CA): Switched Ethernet utilizes switches to subdivide the network architecture, thereby avoiding collisions, increasing network efficiency, and improving determinism. It is widely used in manufacturing applications. The main difference between switch-based and hub-based Ethernet networks is the intelligence of forwarding packets. Hubs simply pass on incoming traffic from any port to all other ports, whereas switches learn the topology of the network and forward packets to the destination port only. In a star-like network layout, every node is connected with a single cable to the switch as a full-duplex point-to-point link. Thus, collisions can no longer occur on any network cable. Switched Ethernet relies on this star cluster layout to achieve this collision-free property.

Switches employ the cut-through or store-and-forward technique to forward packets from one port to another, using per-port buffers for packets waiting to be sent on that port. Switches with cut-through first read the MAC address and then forward the packet to the destination port according to the MAC address of the destination and the forwarding table on the switch. On the other hand, switches with store-and-forward examine the complete packet first. Using the cyclic redundancy check (CRC) code, the switch will first verify that the frame has been correctly transmitted before forwarding the packet to the destination port. If there is an error, the frame will be discarded. Store-and-forward switches are slower but will not forward any corrupted packets.

Although there are no message collisions on the networks, congestion may occur inside the switch when one port suddenly receives a large number of packets from the other ports. If the buffers inside the switch overflow, messages will be lost [14]. Three main queuing principles are implemented inside the switch in this case. They are first-in/first-out (FIFO) queue, priority queue, and per-flow queue. The FIFO queue is a traditional method that is fair and simple. However, if the network traffic is heavy, the network QoS for timely and fair delivery cannot be

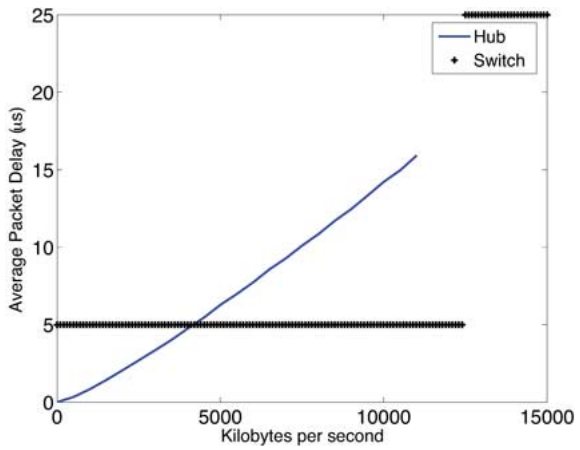

Fig. 8. *Packet delay as function of node traffic for hub and switch [40]. Simulation results with baselines (delay magnitudes) computed from experiments.*

guaranteed. In the priority queuing scheme, the network manager reads some of the data frames to distinguish which queues will be more important. Hence, the packets can be classified into different levels of queues. Queues with high priority will be processed first followed by queues with low priority until the buffer is empty. With the per-flow queuing operation, queues are assigned different levels of priority (or weights). All queues are then processed one by one according to priority; thus, the queues with higher priority will generally have higher performance and could potentially block queues with lower priority [6].

Thus, although switched Ethernet can avoid the extra delays due to collisions and retransmissions, it can introduce delays associated with buffering and forwarding. This tradeoff can be seen in Fig. 8, which shows the average packet delay as a function of node traffic. The switch delay is small but constant until the buffer saturates and packets must be resent; the hub delay increases more gradually. Examples of timing analysis and performance evaluation of switched Ethernet can be found in [28], [40], and [58].

3) Wireless Ethernet (CSMA/CA): Wireless Ethernet, based on the IEEE 802.11 standard, can replace wired Ethernet in a transparent way since it implements the two lowest layers of the ISO-OSI model [24], [61]. Besides the physical layer, the biggest difference between 802.11 and 802.3 is in the medium access control. Unlike wired Ethernet nodes, wireless stations cannot "hear" a collision. A collision avoidance mechanism is used but cannot entirely prevent collisions. Thus, after a packet has been successfully received by its destination node, the receiver sends a short acknowledgment packet (ACK) back to the original sender. If the sender does not receive an ACK packet, it assumes that the transmission was unsuccessful and retransmits.

The collision avoidance mechanism in 802.11 works as follows. If a network node wants to send while the network is busy, it sets its backoff counter to a randomly chosen value. Once the network is idle, the node waits first for an interframe space (DIFS) and then for the backoff time before attempting to send, see Fig. 9. If another node accesses the network during that time, it must wait again for another idle interval. In this way, the node with the lowest backoff time sends first. Certain messages (e.g., ACK) may start transmitting after a shorter interframe space (SIFS), thus they have a higher priority. Collisions may still occur because of the random nature of the backoff time; it is possible for two nodes to have the same backoff time.

Several refinements to the protocol also exist. Nodes may reserve the network either by sending a request to send (RTS) message or by breaking a large message into many smaller messages (fragmentation); each successive message can be sent after the smallest interframe time. If there is a single master node on the network, the master can poll all the nodes and effectively create a TDM contention-free network.

In addition to time delays, the difference between the theoretical data rate and the practical throughput of a control network should be considered. For example, raw data rates for 802.11 wireless networks range from 11 to 54 Mbits/s. The actual throughput of the network, however, is lower due to both the overhead associated with the interframe spaces, ACK, and other protocol support

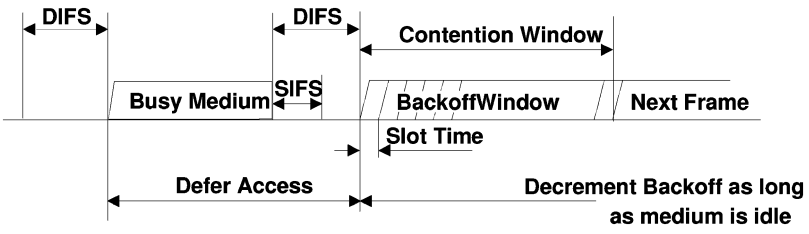

Fig. 9. *Timing diagram for wireless Ethernet (IEEE 802.11).*

Table 2 Maximum Throughputs for Different 802.11 Wireless Ethernet Networks. All Data Rates and Throughputs are in Megabits per Second

Network type	802.11a	802.11g	802.11b
Nominal data rate	54	54	11
Theoretical throughput	26.46	17.28	6.49
Measured throughput	23.2	13.6	3.6

transmissions, and to the actual implementation of the network adapter. Although 802.11a and 802.11g have the same raw data rate, the throughput is lower for 802.11g because its backwards compatibility with 802.11b requires that the interframe spaces be as long as they would be on the 802.11b network. Computed and measured throughputs are shown in Table 2 [9]. The experiments were conducted by continually sending more traffic on the network until a further setpoint increase in traffic resulted in no additional throughput.

Experiments conducted to measure the time delays on wireless networks are summarized in Table 3 and Fig. 10 [9]. Data packets were sent from the client to the server and back again, with varying amounts of cross traffic on the network. The send and receive times on both machines were time-stamped. The packet left the client at time t_a and arrived at the server at time t_b, then left the server at time t_c and arrived at the client at time t_d. The sum of the pre- and postprocessing times and the transmission time on the network for both messages can be computed as (assuming that the two nodes are identical)

$$\begin{aligned} 2 * T_{\text{delay}} &= 2 * (T_{\text{pre}} + T_{\text{wait}} + T_{tx} + T_{\text{post}}) \\ &= t_d - t_a - (t_c - t_b). \end{aligned}$$

Note that this measurement does not require that the clocks on the client and server be synchronized. Since the delays at the two nodes can be different, it is this sum of the two delays that is plotted in Fig. 10 and tabulated in Table 3.

Two different types of data packets were considered: user datagram protocol (UDP) and object linking and embedding (OLE) for process control (OPC). UDP is a commonly used connectionless protocol that runs on top of Ethernet, often utilized for broadcasting. UDP packets carry only a data load of 50 bytes. OPC is an application-to-application communication protocol primarily utilized in manufacturing to communicate data values. OPC requires extra overhead to support this application layer; consequently, the OPC packets contain 512 data bytes (in addition to the overhead). For comparison purposes, the frame times (including the overheads) are computed for the different packets.

Table 3 Computed Frame Times and Experimentally Measured Delays on Wireless Networks; All Times in Milliseconds

Network type	802.11a	802.11g	802.11b
Frame time (UDP), computed	0.011	0.011	0.055
Median delay (UDP), measured	0.346	0.452	1.733
Frame time (OPC), computed	0.080	0.080	0.391
Median delay (OPC), measured	2.335	2.425	3.692

4) Impact of Ethernet Application Layer Protocols: OPC: OPC is an open communication standard that is often used in industry to connect supervisory control and data acquisition (SCADA) systems and human–machine interfaces (HMIs) to control systems and fieldbus networks [21], [34], [52]. It is based on the Microsoft DCOM standard [43] and is the dominant factory-floor application layer protocol utilized for diagnostics and is beginning to be used for sequential control [42]. The main benefit of OPC is that it allows any products that support the standard to share data. Although OPC actually consists of many different communication specifications, its most commonly used form is called Data Access, which supports both client-server and publisher-subscriber communication models. The server maintains a table of data values, and the client can read or write updates. The overhead associated with OPC (and DCOM in general) is significant, as shown in Fig. 10. Most of this delay is due to the software implementation of the OPC protocol; OPC was never intended for a real-time environment. However, it is very useful to push data up from the low-level controls to the higher-level supervisors or diagnostic systems. It can also be used to send commands down from the HMIs to the control systems. Its high level of interoperability enables the connection of multiple control systems from different vendors in a unified manner. However, when OPC is used to send control data, the additional delay caused by the higher level application layer protocol must be considered.

IV. APPLICATIONS OF NETWORKS IN MANUFACTURING

In this section, we briefly describe current trends in the use of networks for distributed, multilevel control and diagnostics as well as safety. There is an enormous amount of data produced in a typical manufacturing system, as thousands of sensors record position, velocity, flow, temperature, and other variables hundreds of times every minute. In addition to this physical information, there are the specifications for the parts that must be produced, the orders for how many parts of each type are needed, and the maintenance schedules for each machine. Generally, the information content can be thought of as supporting a control, diagnostics, or safety function, or some combination of these. In order to support the aggregate of these functions in a manufacturing environment, separate networks are often employed, where each network is dedicated to one or more function types,

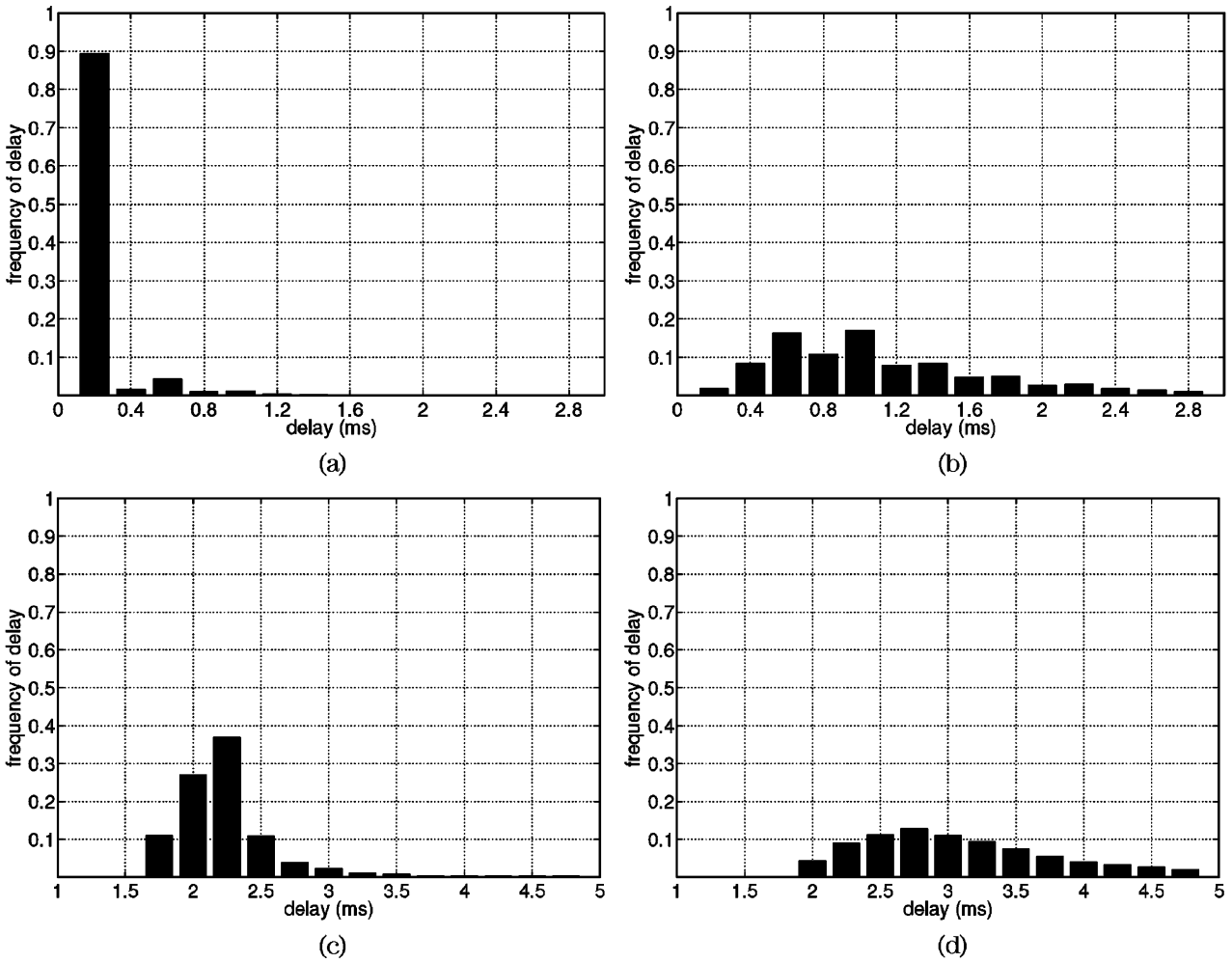

Fig. 10. *Distributions of packet delays for different values of cross-traffic throughput on 802.11a network: (a) UDP delays, 3 Mb/s cross traffic; (b) UDP delays, 22 Mb/s cross traffic; (c) OPC delays, 3 Mb/s cross traffic; (d) OPC delays, 22 Mb/s cross traffic.*

such as control and diagnostics, with the network protocol chosen that best fits (i.e., balances) the QoS requirements of the function type(s). In this section, networks for these function types are explored, focusing on the QoS requirements that often govern the network protocol choice.

A. Networks for Control

Control signals can be divided into two categories: real time and event based. For real-time control, signals must be received within a specified amount of time for correct operation of the system. Examples of real-time signals include continuous feedback values (e.g., position, velocity, acceleration) for servo systems, temperature and flow in a process system, and limit switches and optical sensors in material flow applications (e.g., conveyors). In order to support real-time control, networks often must have a high level of determinism, i.e., they must be able to guarantee end-to-end communication of a signal within a specified amount of time. Further, QoS of networked control systems can be very dependent upon the amount of jitter in the network, thus, for example, fixed determinism is usually preferred over bounded determinism.

Event-based control signals are used by the controller to make decisions but do not have a time deadline. The system will wait until the signal is received (or a timeout is reached) and then the decision is made. An example of an event-based signal is the completion of a machining operating in a CNC; the part can stay in the machine without any harm to the system until a command is sent to the material handler to retrieve it.

In addition to dividing control signals by their time requirements, the data size that must be transmitted is important. Some control signals are a single bit (e.g., a limit switch), whereas others are very large (e.g., machine vision). Generally speaking, however, and especially with real-time control, data sizes on control networks tend to be relatively small and high levels of determinism are preferred.

Control networks in a factory are typically divided into multiple levels to correspond to the factory control distributed in a multitier hierarchical fashion. At the lowest level of networked control are device networks, which are usually characterized by smaller numbers of nodes (e.g., less than 64), communicating small data packets at high sample frequencies and with a higher level of determinism. An example of networked control at this level is servo control; here, network delay and jitter requirements are very strict. Deterministic networks that support small data packet transmissions, such as CAN-based networks, are very common at this level. Although seemingly nonoptimal for this level of control, Ethernet is becoming more common, due to the desire to push Ethernet to all levels in the factory and the increasing determinism possible with switched Ethernet. Oftentimes, determinism and jitter capabilities for lower level networked control are enhanced by utilizing techniques that minimize the potential for jitter through network contention, such as master–slave operation, polling techniques, and deadbanding [29], [44].

An intermediate level of network is the cell or subsystem, which includes SCADA. At this level, multiple controllers are connected to the network (instead of devices directly connected to the network). The controllers exchange both information and control signals, but since the cells or subsystems are typically physically decoupled, the timing requirements are not as strict as they are at the lowest levels, or are nonexistent if event-driven control is enforced [41]. These networks are also used to download new part programs and updates to the lower level controllers. Token-passing and Ethernet-based networks are commonly used at this level, with the ability to communicate larger amounts of data and support for network services generally taking precedence over strict determinism.

Networks at the factory or enterprise level coordinate multiple cells and link the factory floor control infrastructure to the enterprise level systems (e.g., part ordering, supply chain integration, etc.). Large amounts of data travel over these networks, but real-time requirements are usually nonexistent. Ethernet is the most popular choice here primarily because internet support at this level is usually critical, and Ethernet also brings attractive features to this environment such as support for high data volumes, network services, availability of tools, capability for wide area distribution, and low cost.

B. Networks for Diagnostics

Diagnostic information that is sent over the network often consists of large amounts of data sent infrequently. For example, a tool monitoring system may capture spindle current at 1 kHz. The entire current trace would be sent to the diagnostic system after each part is cut (if the spindle current is used for real-time control, it could be sent over the network every 1 ms, but this would then be considered control data). The diagnostic system uses this information for higher level control, such as to schedule a tool change or shut down a tool that is underperforming.

Diagnostics networks are thus usually set up to support large amounts of data with the emphasis on speed over determinism. Ethernet is the dominant network protocol in system diagnostics networks. As with control, diagnostics is often set up in a multitier hierarchical fashion, with different physical layer technology (e.g., wireless, broadband and fiberoptic) utilized at different levels to support the data volume requirements. Also, a variety of data compression techniques, such as change-of-state reporting and store and forwarding of diagnostic information on a process "run-by-run" basis, are often used in communicating diagnostic information up the layers of the network hierarchy [25], [51].

As noted in Section I, diagnostics networks enable diagnostics of the networked system rather that the network itself (with the latter referred to as "network diagnostics"). Both types of diagnostics are commonly used in manufacturing systems. Many network protocols have built-in network diagnostics. For example, nodes that are configured to send data only when there is a change of state may also send "heartbeat" messages every so often to indicate they are still on the network.

C. Networks for Safety

One of the newest applications of networks in manufacturing is safety [1]. Traditionally, safety interlocks around manufacturing cells have been hardwired using ultra-reliable safety relays to ensure that the robots and machines in the cell cannot run if the cell door is open or there is an operator inside the cell. This hardwiring is not easy to reconfigure and can be extremely difficult to troubleshoot if something goes wrong (e.g., a loose connection). Safety networks allow the safety systems to be reconfigurable and significantly improve the ability to troubleshoot. They also allow safety functions to be more easily coordinated across multiple components, e.g., shutting down all machines in a cell at the same time and also coordinating "soft shutdown" where appropriate (safe algorithms for gradual shutdown of systems without damage to systems and/or scrapping of product). Further, safety network systems often provide better protection against operators bypassing the safety interlocks and thus make the overall system safer.

Safety networks have the strongest determinism and jitter requirements of all network function types. Safety functions must be guaranteed within a defined time, thus the network must provide that level of determinism. Further, the network must have a deterministic heartbeat-like capability; if network connectivity fails for any reason, the system must revert to a safe state within the guaranteed defined time. CAN-based networks are popular candidates for networked safety because of their high levels of determinism and the network self-diagnostic

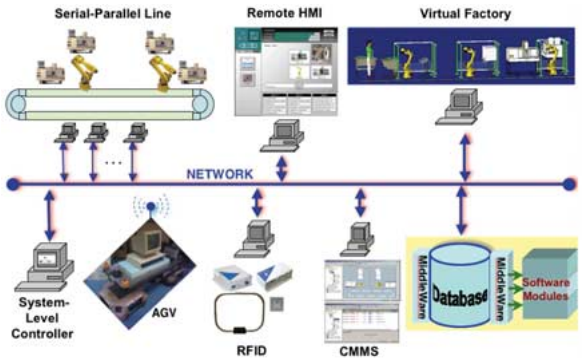

Fig. 11. ***Reconfigurable factory testbed.***

mechanisms they can utilize to determine node and network health. However, it is important to note that most network protocols, in and of themselves, are not adequate to the task of supporting safety networking. Additional protocol capabilities, usually at the higher levels (e.g., application) are often instituted to guarantee proper safety functionality [13].

V. MULTILEVEL FACTORY NETWORKING EXAMPLE: RFT

The RFT at the University of Michigan is a comprehensive platform that enables research, development, education, validation, and transfer of reconfigurable manufacturing system (RMS) concepts [37]. It consists of both real and virtual machines controlled over a communication network and coordinated through a unified software architecture. The RFT is conceived to be extensible so as to allow the modular incorporation and integration of additional components (hardware and/or software, real and/or virtual). The hardware components of the RFT include a serial-parallel manufacturing line consisting of two machining cells connected by a conveyor, a suite of communication and control networks, an automated guided vehicle (AGV), and an RFID system. The software components of the RFT include a virtual factory simulation, an open software integration platform and data warehouse, an infrastructure of web-based HMIs, and a computerized maintenance management system (CMMS). A schematic of the RFT is shown in Fig. 11.

The network shown in Fig. 11 represents a multitier networked control, diagnostic, and safety network infrastructure that exists on the RFT. The serial–parallel line component of the RFT is the primary component currently being utilized to explore manufacturing networks. With respect to *control* networks, cell 1 has a DeviceNet network to connect the machines and robot controllers (including the robot gripper and the clamps in the machines); cell 2 uses PROFIBUS for the same purpose. The conveyor components (pallet stops, pallet sensors, motor, controller) communicate via a second DeviceNet network. The cell-level controllers (including the conveyor controller) communicate with the system level controller (SLC) over Ethernet via OPC and support an event-based control paradigm. The SLC has a wireless network connection with the AGV. All of these control networks are shown in Fig. 12.

The network infrastructure for collecting diagnostic data on the RFT uses OPC. For example, for every part that is machined, the spindle current on the machine is sampled at 1 kHz. This time-dense data is directly sampled using LabVIEW,[5] and then stored in the database. Compressed diagnostics data that focuses on identifying specific features of the current trace is passed to higher levels in the diagnostics network.

Networks for safety are implemented in the serial–parallel line utilizing the SafetyBUS p protocol, as shown in Fig. 13. As with the control and diagnostics system, the implementation is multitier, corresponding to levels of safety control. Specifically, safety networks are implemented for each of the two cells as well as the conveyor. The safety network interlocks the emergency stops, robot cages, and machine enclosures with the controllers. These three cell level networks are coordinated through a hierarchy to a high-level master safety network. This implementation allows for safety at each cell to be controlled individually, but also provides a capability for system-wide safe shutdown. Further, this implementation allows for multitier logic to be utilized for the implementation of safety algorithms.

The RFT implementation of multitiered networks for control, diagnostics, and safety provides a rich research environment for exploration into industrial control networks. Specifically, topics that can be addressed include: 1) coordination of control, diagnostics and/or safety operation over one, two or three separate networks; 2) distributed control design and operation over a network; 3) distribution of control and diagnostics in a hierarchical networked system; 4) compression techniques for hierarchical diagnostics systems; 5) remote control safe operation; 6) hierarchical networked safety operation and "soft shutdown"; 7) heuristics for optimizing control/diagnostics/safety network operation; 8) network standards for manufacturing; as well as 9) best practices for network systems design and operation [29], [30], [37], [45].

VI. FUTURE TRENDS

The pace of adoption of networks in industrial automation shows no sign of slowing anytime soon. The immediate advantages of reduced wiring and improved reliability have been accepted as fact in industry and are often significant enough by themselves (e.g., return-on-investment) to justify the move to networked solutions. Once the control data is on the network, it can be used by diagnostics,

[5]National Instruments, Austin, TX.

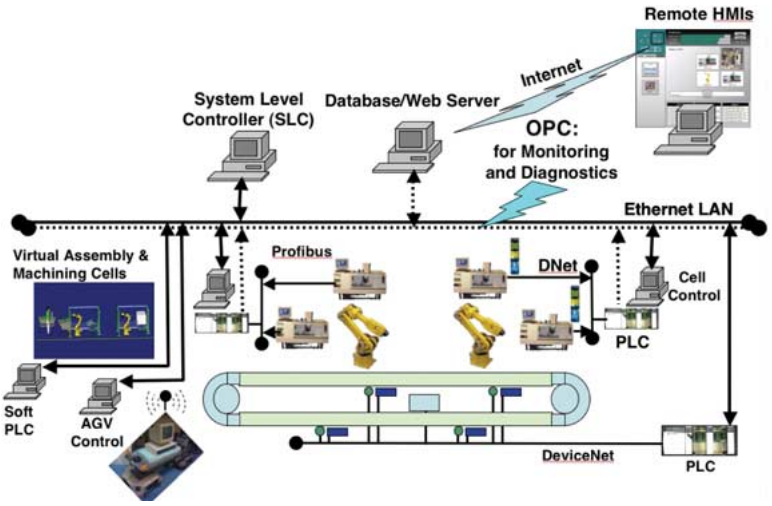

Fig. 12. *Networks on RFT. Control networks are indicated by solid lines, and diagnostics networks are indicated by dashed lines.*

scheduling, quality control, and other higher level control systems. Diagnostics network adoption will continue to lead the way, followed by control and then safety networks, but the ordering is driven by the stricter QoS balancing requirements of the latter, not by any belief of higher ROI of the former. In fact, in gauging the criticality of control and safety with respect to diagnostics, it is conceivable that significantly higher ROI may be achievable in the migration to control, especially safety networking. However, even with diagnostics networks the possibilities and benefits of e-Diagnostics and (with control) e-Manufacturing are only beginning to be explored.

Looking to the future, the most notable trend appearing in industry is the move to wireless networks at all levels [61]. Wireless networks further reduce the volume of wiring needed (although oftentimes power is still required), enable the placement of sensors in difficult locations, and better enable the placement of sensors on moving parts such as on tool tips that rotate at several thousand revolutions per minute. Issues with the migration to wireless include interference between multiple wireless networks, security, and reliability and determinism of data transmission. The anticipated benefit in a number of domains (including many outside of

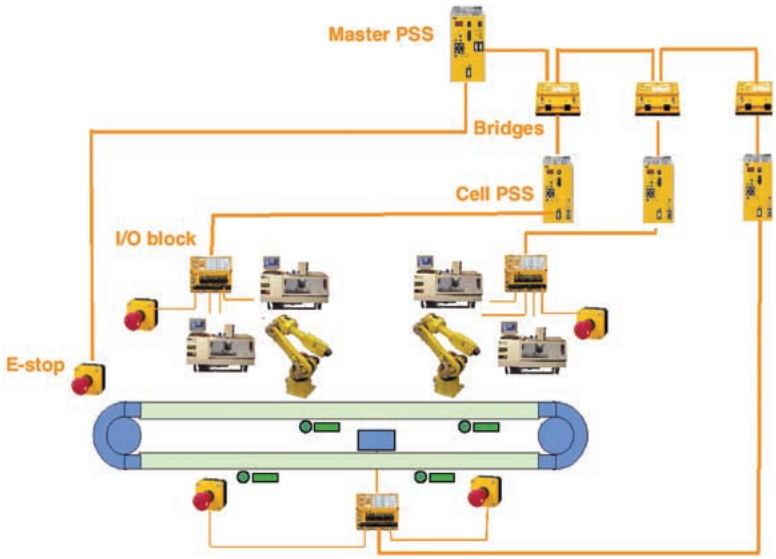

Fig. 13. *Safety network implementation on RFT.*

manufacturing) is driving innovation that manufacturing, in general, can leverage. It is not inconceivable that wireless will make significant in-roads into networked control and even safety over the next five to ten years.

Over the next five years, many among the dozens of protocols that have been developed for industrial networks over the last few decades will fall out of favor, but will not die overnight due to the large existing installed base and the long lifetime of manufacturing systems. In addition, new protocols may continue to emerge to address niches where a unique QoS balance is needed. However, it is expected that Ethernet and wireless will continue to grab larger and larger shares of the industrial networks installed base, driven largely by lower cost through volume, the internet, higher availability of solutions and tools for these network types (e.g., web-enabled tools), and the unmatched flexibility of wireless. Indeed, it is not unreasonable to expect that, in the next decade, the next major milestone in industrial networking, namely the *wireless factory*, will be within reach, where diagnostics, control, and safety functions at multiple levels throughout the factory are enabled utilizing wireless technology. ■

Acknowledgment

The authors would like to thank the students who did much of the work on which much of this paper is based, especially J. Parrott and B. Triden for their review of this manuscript, A. Duschau-Wicke for his experimental work on delays in wireless networks, and F.-L. Lian for his extensive research on networked control systems.

REFERENCES

[1] J. Alanen, M. Hietikko, and T. Malm. (2004). "Safety of digital communications in machines," VTT Technical Res. Center Finland, Tech. Rep. VTT Tiedotteita—Research Notes 2265. [Online]. Available: http://www.vtt.fi/inf/pdf

[2] P. Antsaklis and J. Baillieul, Eds., "Special issue on networked control systems," *IEEE Trans. Automat. Contr.*, vol. 49, no. 9, pp. 1421–1597, Sep. 2004.

[3] AS-I Standard. (2005). [Online]. Available: http://www.as-interface.net

[4] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[5] B. J. Casey, "Implementing Ethernet in the industrial environment," in *Proc. IEEE Industry Applications Soc. Annu. Meeting*, Seattle, WA, Oct. 1990, vol. 2, pp. 1469–1477.

[6] J.-D. Decotignie, "Ethernet-based real-time and industrial communications," *Proc. IEEE*, vol. 93, no. 6, pp. 1102–1118, Jun. 2005.

[7] *DeviceNet Specifications*, 1997.

[8] L. Dugyard and E. I. Verriest, *Stability and Control of Time-Delay Systems*. New York: Springer, 1998.

[9] A. Duschau-Wicke, "Wireless monitoring and integration of control networks using OPC," NSF Eng. Res. Center Reconfigurable Manufacturing Systems, Univ. Michigan, Tech. Rep., 2004, Studienarbeit report for Technische Universitat Kaiserslautern.

[10] D. Dzung, M. Naedele, T. P. Von Hoff, and M. Crevatin, "Security for industrial communication systems," *Proc. IEEE*, vol. 93, no. 6, pp. 1152–1177, Jun. 2005.

[11] M.-Y. Chow, Ed., "Special section on distributed network-based control sytems and applications," *IEEE Trans. Indust. Electron.*, vol. 51, no. 6, pp. 1126–1279, Dec. 2004.

[12] J. Eidson and W. Cole, "Ethernet rules closed-loop system," *InTech*, pp. 39–42, Jun. 1998.

[13] *Railway Applications—Communication, Signalling and Processing Systems Part 1: Safety-Related Communication in Closed Transmission Systems*, Irish Standard EN 50159-1, 2001.

[14] M. Felser, "Real-time Ethernet—Industry prospective," *Proc. IEEE*, vol. 93, no. 6, pp. 1118–1129, Jun. 2005.

[15] M. Felser and T. Sauter, "The fieldbus war: History or short break between battles?" in *Proc. IEEE Int. Workshop Factory Communication Systems (WFCS)*, Västerås, Sweden, Aug. 2002, pp. 73–80.

[16] ——, "Standardization of industrial Ethernet—The next battlefield?" in *Proc. IEEE Int. Workshop Factory Communication Systems (WFCS)*, Vienna, Austria, Sep. 2004, pp. 413–421.

[17] M. Fondl. (2003, Sep. 16). "Network diagnostics for industrial Ethernet," in *The Industrial Ethernet Book*. [Online]. Available: http://ethernet.industrial-networking.com

[18] G. F. Franklin, J. D. Powell, and A. Emani-Naeini, *Feedback Control of Dynamic Systems*, 3rd ed. Reading, MA: Addison-Wesley, 1994.

[19] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Boston, MA: Addison-Wesley, 1998.

[20] Grid Connect. Grid Connect Fieldbus Comparison Chart. [Online]. Available: http://www.synergetic.com/compare.htm

[21] D. W. Holley, "Understanding and using OPC for maintenance and reliability applications," *IEE Computing Contr. Eng.*, vol. 15, no. 1, pp. 28–31, Feb./Mar. 2004.

[22] "IEC standard redefines safety systems," *InTech*, vol. 50, no. 7, pp. 25–26, Jul. 2003.

[23] IEEE. (2002). 1588: Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. [Online]. Available: http://ieee1588.nist.gov

[24] Amer. Nat. Standards Inst., OSI Basic Reference Model, ANSI, ISO/7498, 1984.

[25] International SEMATECH. (2004, Apr.). *Proc. Int. SEMATECH e-Manufacturing/ e-Diagnostics Workshop*. [Online]. Available: http://ismi.sematech.org/emanufacturing/meetings/20040419/index.htm

[26] H. Kaghazchi and D. Heffernan. (2004). "Development of a gateway to PROFIBUS for remote diagnostics," in PROFIBUS Int. Conf., Warwickshire, U.K. [Online]. Available: http://www.ul.ie/~arc/techreport.html

[27] S. A. Koubias and G. D. Papadopoulos, "Modern fieldbus communication architectures for real-time industrial applications," *Comput. Industry*, vol. 26, pp. 243–252, Aug. 1995.

[28] K. C. Lee and S. Lee, "Performance evaluation of switched Ethernet for networked control systems," in *Proc. IEEE Conf. Industrial Electronics Soc.*, Nov. 2002, vol. 4, pp. 3170–3175.

[29] F.-L. Lian, J. R. Moyne, and D. M. Tilbury, "Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet," *IEEE Control Syst. Mag.*, vol. 21, no. 1, pp. 66–83, Feb. 2001.

[30] F.-L. Lian, J. R. Moyne, D. M. Tilbury, and P. Otanez, "A software toolkit for design and optimization of sensor bus systems in semiconductor manufacturing systems," in *Proc. AEC/APC Symp. XIII*, Banff, Canada, Oct. 2001.

[31] F.-L. Lian, J. R. Moyne, and D. M. Tilbury, "Network design consideration for distributed control systems," *IEEE Trans. Contr. Syst. Technol.*, vol. 10, no. 2, pp. 297–307, Mar. 2002.

[32] P. S. Marshall, "A comprehensive guide to industrial networks: Part 1," *Sensors Mag.*, vol. 18, no. 6, Jun. 2001.

[33] P. Martí, J. Yépez, M. Velasco, R. Villà, and J. M. Fuertes, "Managing quality-of-control in network-based control systems by controller and message scheduling co-design," *IEEE Trans. Industrial Electron.*, vol. 51, no. 6, Dec. 2004.

[34] G. A. Mintchell, "OPC integrates the factory floor," *Control Eng.*, vol. 48, no. 1, pp. 39, Jan. 2001.

[35] J. Montague, "Safety networks up and running," *Contr. Eng.*, vol. 51, no. 12, Dec. 2004.

[36] ——, "Networks busting out all over," *Contr. Eng.*, vol. 52, no. 3, Mar. 2005.

[37] J. Moyne, J. Korsakas, and D. M. Tilbury, "Reconfigurable factory testbed (RFT): A distributed testbed for reconfigurable manufacturing systems," in *Proc. Japan—U.S.A. Symp. Flexible Automation*. Denver, CO: Amer. Soc. Mechanical Engineers (ASME), Jul. 2004.

[38] J. Moyne and F. Lian, "Design considerations for a sensor bus system in semiconductor manufacturing," in *Proc. Int. SEMATECH AEC/APC Workshop XII*, Sep. 2000.

[39] J. R. Moyne, N. Najafi, D. Judd, and A. Stock, "Analysis of sensor/actuator bus interoperability standard alternatives for semiconductor manufacturing," in *Sensors Expo Conf. Proc.*, Cleveland, OH, Sep. 1994.

[40] J. Moyne, P. Otanez, J. Parrott, D. Tilbury, and J. Korsakas, "Capabilities and limitations of using Ethernet-based networking

technologies in APC and e-diagnostics applications," in *Proc. SEMATECH AEC/APC Symp. XIV*, Snowbird, UT, Sep. 2002.

[41] J. Moyne, D. Tilbury, and H. Wijaya, "An event-driven resource-based approach to high-level reconfigurable logic control and its application to a reconfigurable factory testbed," in *Proc. CIRP Int. Conf. Reconfigurable Manufacturing Systems*, Ann Arbor, MI, 2005.

[42] [Online]. Available: http://www.opcfoundation.org

[43] *OPC DA 3.00 Specification*. OPC Foundation. (2003, Mar.). [Online]. Available: http://www.opcfoundation.org

[44] P. G. Otanez, J. R. Moyne, and D. M. Tilbury, "Using deadbands to reduce communication in networked control systems," in *Proc. Amer. Control Conf.*, Anchorage, AK, May 2002, pp. 3015–3020.

[45] P. G. Otanez, J. T. Parrott, J. R. Moyne, and D. M. Tilbury, "The implications of Ethernet as a control network," in *Proc. Global Powertrain Congr.*, Ann Arbor, MI, Sep. 2002.

[46] J. T. Parrott, J. R. Moyne, and D. M. Tilbury, "Experimental determination of network quality of service in Ethernet: UDP, OPC, and VPN," in *Proc. Amer. Control Conf.*, 2006.

[47] G. Paula, "Java catches on for manufacturing," *Mechanical Eng.*, vol. 119, no. 12, pp. 80–82, Dec. 1997.

[48] *PROFIBUS Standard*, IEC 61158 type 3 and IEC 61784. [Online]. Available: http://www.profibus.com

[49] K. K. Ramakrishnan and H. Yang, "The Ethernet capture effect: Analysis and solution," in *Proc. 19th Conf. Local Computer Networks*, Minneapolis, MN, Oct. 1994, pp. 228–240.

[50] J.-P. Richard, "Time-delay systems: An overview of some recent advances and open problems," *Automatica*, vol. 39, no. 10, pp. 1667–1694, 2003.

[51] A. Shah and A. Raman. (2003, Jul.). "Factory network analysis," in *Proc. Int. SEMATECH e-Diagnostics and EEC Workshop*. [Online]. Available: http://ismi.sematech.org/emanufacturing/meetings/20030718/index.htm

[52] B. Shetler, "OPC in manufacturing," *Manufacturing Eng.*, vol. 130, no. 6, Jun. 2003.

[53] P. Sink, "Industrial Ethernet: The death knell of fieldbus?" *Manufacturing Automation Mag.*, Apr. 1999.

[54] S. Soucek and T. Sauter, "Quality of service concerns in IP-based control systems," *IEEE Trans. Indust. Electron.*, vol. 51, pp. 1249–1258, Dec. 2004.

[55] A. S. Tanenbaum, *Computer Networks*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1996.

[56] J.-P. Thomesse, "Fieldbus technology in industrial automation," *Proc. IEEE*, vol. 93, no. 6, pp. 1073–1101, Jun. 2005.

[57] A. Treytl, T. Sauter, and C. Schwaiger, "Security measures for industrial fieldbus systems—State of the art and solutions for IP-based approaches," in *Proc. IEEE Int. Workshop Factory Communication Systems (WFCS)*, Vienna, Austria, Sep. 2004, pp. 201–209.

[58] E. Vonnahme, S. Ruping, and U. Ruckert, "Measurements in switched Ethernet networks used for automation systems," in *Proc. IEEE Int. Workshop Factory Communication Systems*, Sep. 2000, pp. 231–238.

[59] K. Watanabe, E. Nobuyama, and A. Kojima, "Recent advances in control of time delay systems—A tutorial review," in *Proc. IEEE Conf. Decision and Control*, 1996, pp. 2083–2088.

[60] J. D. Wheelis, "Process control communications: Token bus, CSMA/CD, or token ring?" *ISA Trans.*, vol. 32, no. 2, pp. 193–198, Jul. 1993.

[61] A. Willig, K. Matheus, and A. Wolisz, "Wireless technology in industrial networks," *Proc. IEEE*, vol. 93, no. 6, pp. 1130–1151, Jun. 2005.

ABOUT THE AUTHORS

James R. Moyne (Member, IEEE) received the B.S.E.E., B.S.E. in mathematics, M.S.E.E., and Ph.D. degrees from the University of Michigan, Ann Arbor.

He is currently an Associate Research Scientist in the Department of Mechanical Engineering, University of Michigan, and Director of the Reconfigurable Factory Testbed within the Engineering Research Center for Reconfigurable Manufacturing Systems. He is also Director of Automotive Technology at Brooks Automation. His research areas include industrial network systems and network protocols, advanced process control, software control, and database technology. He is the author of a number of refereed publications in each of these areas. He headed the team that developed the first industrial network conformance test laboratories for DeviceNet, ControlNet, and Modbus/TCP.

Dr. Moyne co-chairs the sensor bus subcommittee of Semiconductor Equipment and Materials International, the standards organization for the semiconductor industry.

Dawn M. Tilbury (Senior Member, IEEE) received the B.S. degree in electrical engineering, *summa cum laude*, from the University of Minnesota, and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley.

She is currently an Associate Professor of Mechanical Engineering at the University of Michigan, Ann Arbor. She is coauthor of the textbook *Feedback Control of Computing Systems*. She was a member of the 2004–2005 class of the Defense Science Study Group (DSSG) and is a current member of DARPA's Information Science and Technology Study Group (ISAT). Her research interests include distributed control of mechanical systems with network communication, logic control of manufacturing systems, and uncertainty modeling in cooperative control.

Dr. Tilbury won the 1997 EDUCOM Medal for her work on the web-based Control Tutorials for Matlab. She received an NSF CAREER award, in 1999, and is the 2001 recipient of the Donald P. Eckman Award of the American Automatic Control Council. She is a member of ASEE, ASME, and SWE and is an elected member of the IEEE Control Systems Society Board of Governors.