Introduction to the LonWorks® System

Version 1.0

ECHELON Corporation

078-0183-01A

Model Number 19000

Echelon Corporation
4015 Miranda Avenue
Palo Alto, CA 94304, USA

Contents

1 Introduction **1-1**
Overview 1-2
Getting More Information 1-3

2 Control Networks **2-1**
The Traditional Approach 2-2
The New Control Network Approach 2-3
The Transition from Data to Control Networks 2-4
Control Network Components 2-4
Using New Technologies in Old Designs 2-6

3 The LONWORKS Protocol **3-1**
Introduction to the LONWORKS Protocol 3-2
Channel Types 3-4
Media Access 3-5
Addressing 3-5
Message Services 3-7
Network Variables 3-7
Limits 3-8
The LONWORKS Protocol Standard 3-9
Summary 3-9

4 Interoperability **4-1**
Overview 4-2
The LONMARK Association 4-3
Transceiver and Physical Channel Standards 4-3
Application Program Standards 4-4
Standard Network Variable Types (SNVTs) 4-4
Configuration Properties 4-4
LONMARK Objects and Functional Profiles 4-5
Program IDs 4-6
LONMARK Resource Files 4-7

5 The LONWORKS System **5-1**
Building a System 5-2
The Neuron Chip 5-3
Neuron Application Programs 5-4
Transceivers 5-5
LONWORKS Devices 5-5
LonPoint Modules 5-6
Routers 5-6
Development Tools 5-7
Network Interfaces, Gateways, and Web Servers 5-7
Network Operating Systems 5-8
Network Tools 5-9
LonMaker for Windows Integration Tool 5-9

LonManager Protocol Analyzer 5-11
LNS DDE Server 5-11

6 Designing Open Systems 6-1
Introduction 6-2
Open System Design Requirements 6-3
A New Design Paradigm 6-4
Hierarchical Systems 6-7
Design Guidelines 6-10
A Checklist for Open Control Design 6-11

7 Implementing Open Systems 7-1
Implementation Tasks 7-2
System Design 7-2
Network Configuration 7-2
Application Configuration 7-3
Installation 7-3
Benefits of an Open Implementation 7-3

A Appendix A – Glossary A-1

B Appendix B – Frequently Asked Questions B-1

1

Introduction

This chapter provides an overview of the LONWORKS® system.

Overview

With thousands of application developers and millions of devices installed worldwide, the LONWORKS system is the leading open solution for building and home automation, industrial, transportation, and public utility control networks. A control network is any group of devices working in a peer-to-peer fashion to monitor sensors, control actuators, communicate reliably, manage network operation, and provide complete access to network data. A LONWORKS network uses the *LONWORKS protocol*, also known as the ANSI/EIA 709.1 Control Networking Standard, to accomplish these tasks.

The LONWORKS system is based on the following concepts:

- Control systems have many common requirements regardless of application.
- A networked control system is significantly more powerful, flexible, and scaleable than a non-networked control system.
- Businesses can save and make more money with control networks over the long term than they can with non-networked control systems.

In some ways, a LONWORKS network resembles a computer data network referred to as a Local Area Network or LAN. Data networks consist of computers attached to various communications media, connected by routers, which communicate with one another using a common protocol such as TCP/IP. Data networks are optimized for moving large amounts of data, and the design of data network protocols assumes that occasional delays in data delivery and response are acceptable. Control networks contain similar pieces optimized for the cost, performance, size, and response requirements of control. Control networks allow networked systems to extend into a class of applications that data networking technology cannot reach. Manufacturers of control systems and devices are able to shorten their development and engineering time by designing LONWORKS components into their products. The result is cost effective development and consistency that allows devices from multiple manufacturers to be able to communicate.

LONWORKS networks range in sophistication from small networks embedded in machines to large networks with thousands of devices controlling fusion lasers, paper manufacturing machines, and building automation systems. LONWORKS networks are used in buildings, trains, airplanes, factories, and hundreds of other processes. Manufacturers are using open, off-the-shelf chips, operating systems, and parts to build products that feature improved reliability, flexibility, system cost, and performance.

Traditional control networks use closed islands of control linked with proprietary gateways. These gateways are difficult to install and maintain, and lock the customer into a closed, non-interoperable architecture. Ultimately, the high cost of this design approach has limited the market for control systems. The LONWORKS system is accelerating the trend away from these proprietary control schemes and centralized systems by providing interoperability, robust technology, faster development, and scale economies. Distributing the processing throughout the network and providing open access to every device lowers the overall installation and life cycle costs, increases reliability by minimizing single points of failure, and providing the

flexibility to adapt the system to a wide variety of applications. For example, in the building control industry, LONWORKS networks are used to provide a common infrastructure for all building systems. This allows the building automation system designer to eliminate excessive vertical integration, which is the often the reason for vertical isolation.

Echelon manufactures over 80 LONWORKS products to help developers, system integrators, and end-users implement LONWORKS networks. These products provide a complete LONWORKS solution including development tools, network management software, power line and twisted pair transceivers and control modules, network interfaces, technical support and training.

This document is an introduction to the basics of the LONWORKS system. It begins with an overview of networks and protocols, highlights the technical aspects of the LONWORKS protocol, provides an overview of the components of the LONWORKS system, and ends with a discussion on achieving product interoperability. The next section provides a list of more detailed related reading. Many of the technical details discussed in this document are handled automatically by the protocol, the network operating system or network tools. The automatic handling of the lower level details of device communication is, in fact, one of the great strengths of the LONWORKS system.

Getting More Information

For more information on the LonWorks system, consult the following documents or browse Echelon's Web site at www.echelon.com. The documents listed below are available at www.echelon.com.

- *LonTalk® Protocol* (005-0017-01)
- *LONMARK Application Layer Interoperability Guidelines* (078-0120-01)
- *LONMARK® Layer 1-6 Interoperability Guidelines* (078-0014-01)
- *LONWORKS Network Services (LNS™) Architecture Strategic Overview* (39310)
- *LonMaker for Windows User's Guide* (39510)
- *LonManager® Protocol Analyzer User's Guide* (39600)
- *LonPoint® Application and Plug-in Guide* (078-0166-01)
- *LonPoint Module Hardware & Installation Guide* (078-0167-01)
- *PCC-10 PC Card User's Guide* (078-0155-01)
- *PCLTA-10 PC LonTalk Adapter User's Guide* (078-0159-01)
- *PCLTA-20 PC LonTalk Adapter User's Guide* (078-0179-01)
- *SLTA-10 Serial LonTalk Adapter User's Guide* (078-0160-01)
- LONWORKS System Data Sheets

2

Control Networks

This chapter explains how control networks enable the deployment of open control systems. The traditional approach to designing closed control systems is described and contrasted with the new approach of using open control networks. Finally, a hybrid approach where control network technology is used to continue the closed control system legacy is described.

The Traditional Approach

At one time, control logic was derived either through electromechanical relay panels or via pneumatic receiver/controllers. The advent of solid-state technology offered a means of reducing costs and increasing flexibility by using logic circuits to replace the wire or tubing and relays. Increasingly powerful algorithms were developed allowing tighter control over processes. However, the issues associated with adds, moves, and changes remained and grew increasingly troublesome as systems grew in size.

It was often the proprietary nature of the hardware and software that caused problems. Each manufacturer built their own systems and provided all the intelligent devices within the system. Though this provided a single point of responsibility for the system, it also 'locked-in' the customer and forced the customer to continue to deal with the original equipment manufacturer for the life of their system, whether it was a building, factory, or processing plant. Worse, the need to design, engineer, and produce an entire system limited the manufacturers to a handful of large companies. These companies tended to move slowly and quickly developed business models built upon the idea of customer lock-in. Compare the price/performance improvement of computing vs. building and industrial controls equipment and the dramatic difference becomes clear.

It has been historically difficult to interconnect digital controllers from different manufacturers. The incompatible communication protocols in the different systems focus on linking separate systems with relays, custom gateways, and programmed RS-232 ports. These interfaces, however, do not provide a detailed, seamless view into the different systems. They allowed only limited status and control information to be passed between the different systems. Fault status information could not be shared, information from different sensors was not always accessible, and systems could not adapt their responses in real-time based on the overall system status. It is possible to create intelligent building and industrial applications using gateways and custom programs, but they are typically not cost effective and reliability of the systems suffer. Once complete, the owner is forever married to those who provide the gateways and custom programming.

Figure 1 shows the centralized architecture that up until recently has been typical of most control systems in commercial and industrial applications. Sensors and actuators are wired to a sub-panel, which in turn connects to the controller panel via a proprietary master/slave communication bus. The controller panel contains a high-performance microprocessor running a custom application program that implements the control logic for all the I/O points connected to it. For large systems, this controller may communicate over another proprietary communication bus with other controllers. Sensors and actuators are typically 'dumb' I/O devices, meaning they have no internal intelligence or communication capabilities. The system typically has a proprietary human-machine interface (HMI). Every system must have a custom application program. This application is developed using a proprietary programming language and non-standard software tools that are manufacturer specific. Unfortunately, the manufacturers make no attempt to standardize the tool sets or programming models.

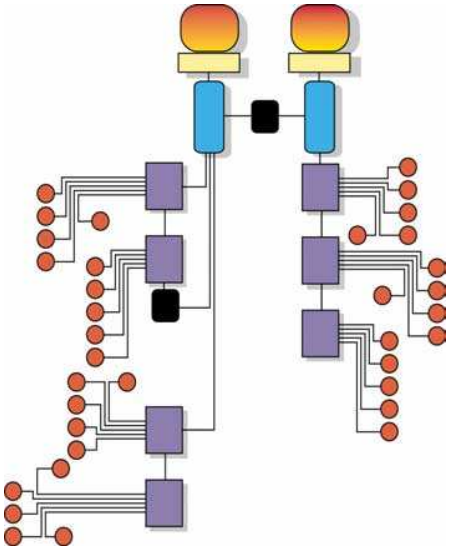

Figure 1 Centralized Architecture

Standardization requires an open control network. Much as the Internet spurned standardization for data networks, the LONWORKS system is the catalyst for standardization of control networks.

The New Control Network Approach

To understand good control network design, one must first understand the function of an open network. Put simply, an open network allows a number of intelligent devices to communicate directly with each other. No intervening supervisory controller is required to poll devices for information and then retransmit that information to other devices. No supervisory device is charged with responsibility for system-wide control algorithms.

This means that every device is capable of publishing information directly to other devices on the network. This information is transmitted by a sender in packets of data that are received by one or more receivers. An open control network is illustrated in **Figure 2**. The change from the master/slave architecture of **Figure 1** to the open architecture of **Figure 2** is exactly the type of change from proprietary hosts to open communication that has fueled the growth of the Internet.

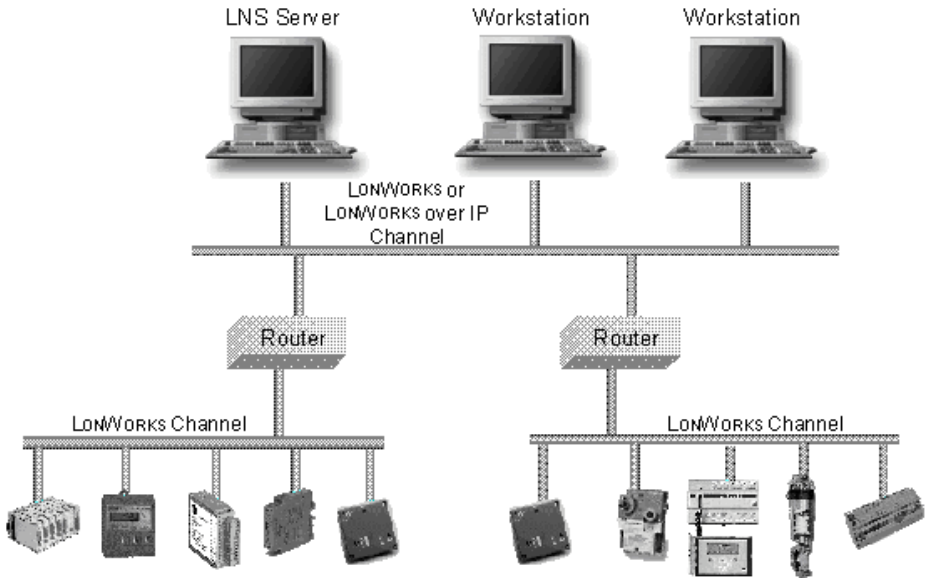

Figure 2 Open Control Network

The Transition from Data to Control Networks

Networks have been around for a number of years, yet they were not typically used for controlling devices other than large computing systems. The communication protocols employed were designed and optimized for passing large amounts of data between computers designed for batch processing. Through time, these protocols evolved to increase in scale and incorporate greater functionality and flexibility. Most, however, continued to be designed for data communication between computers or individuals.

Eventually, the cost of microprocessors reached the point that they could be incorporated into inexpensive controllers and control devices. It was at this point that design engineers began to realize the communication protocols they were using were not really tuned for optimal performance in control systems. Control networks have a number of unique requirements that make them different from data networks. These include the following:

- Frequent, reliable, secure communications between devices
- Short message formats for the information being passed
- Peer-to-peer functionality for every device
- Price points that enable small, low-cost devices

It was the need to address these control specific network requirements, together with the belief that a market standard for communications would allow interoperability that would empower the market to increase in size and efficiency, that brought about the introduction of the LONWORKS protocol.

Control Network Components

Figure 3 illustrates the key components of a control network. A control network consists of intelligent *devices* that communicate with each other

using a common *protocol* over one or more *communications channels*. Network devices are sometimes called *nodes*.

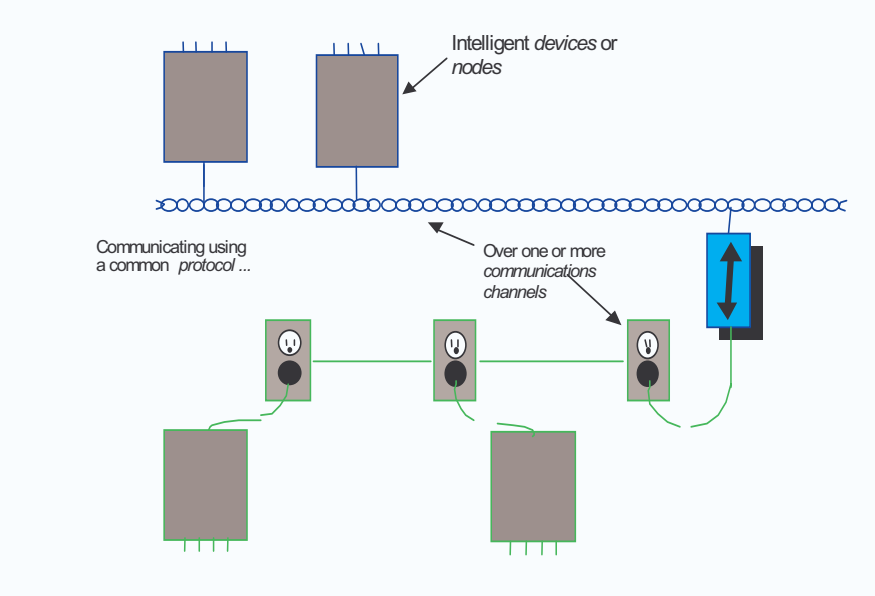

Figure 3 A Control Network

Each *device* includes one or more processors that provide its intelligence and implement the protocol. Each device also includes a component called a *transceiver* to provide its electrical interface to the communications channel.

A device publishes information as appropriate to the application that it is running. The applications are not synchronized, and it is possible that multiple devices may all try to talk at the same time. Meaningful transfer of information between devices on a network, therefore, requires organization in the form of a set of rules and procedures. These rules and procedures are called the *communication protocol*, often abbreviated as the *protocol*. The protocol defines the format of the message being transmitted between devices and defines the actions expected when one device sends a message to another. The protocol normally takes the form of embedded software or firmware code in each device on the network.

The path between devices exhibits various physical characteristics and is called the *communications channel*, or simply *channel*. Different transceivers may be able to interoperate on the same channel, so channels are categorized by *channel type*, and every type of transceiver must identify the channel type or types that it supports. The choice of channel type affects transmission speed and distance as well as the network topology.

All devices connected to a specific channel must have compatible transceivers with compatible configuration. It is possible to build a transceiver for any medium, though some are more difficult to implement and therefore more expensive. Transceivers are available for a variety of communications media including single twisted-pair cable, power line, radio frequency (RF), infrared, fiber optics, and coax cable.

Using New Technologies in Old Designs

Not all control system manufacturers are ready to deliver truly open platforms. One of the more common system architectures deployed today by building, discrete, and process control manufacturers is the *hierarchical system* shown in **Figure 4**. Here we see controllers, which may even incorporate a few components of the LONWORKS system, connected to isolated networks. These networks are sometimes called *device busses* or *device networks*. The emphasis is still on providing proprietary access to sensors and actuators rather than distributing the intelligence to the field devices and providing access to any point on the network from the controllers and workstations anywhere in the hierarchy. A single vendor provides software for the proprietary controller/gateways and none of the interfaces are standardized so that tools from multiple manufactures can be used. The technology of the gateways may appear to be modern, sometimes incorporating the latest technology such as Java. The gateways sometimes communicate on an open network, also called a *control bus*. But the end-result of a hierarchical architecture is still a closed-proprietary system.

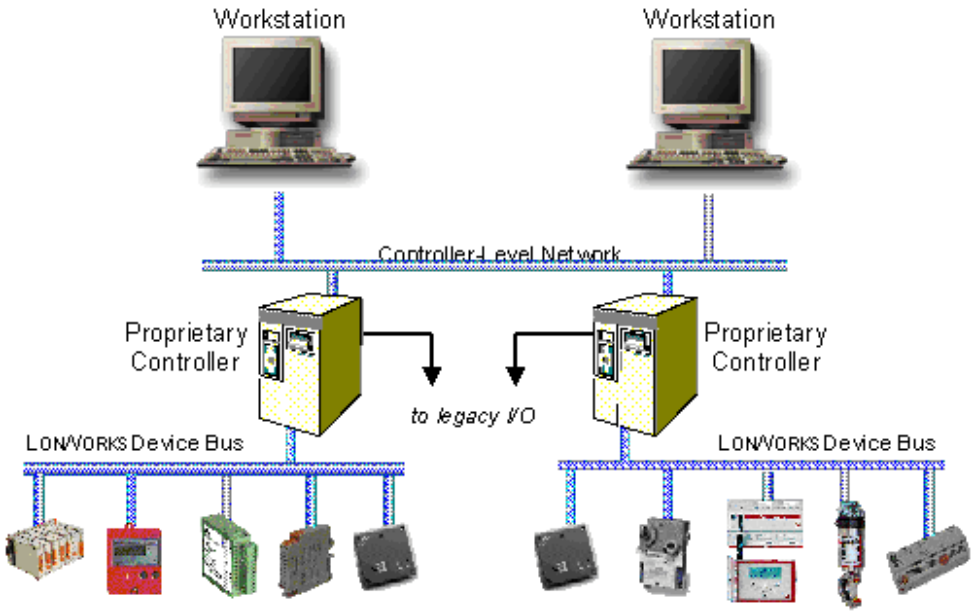

Figure 4 Typical Hierarchical System Architecture

Even when implemented with LONWORKS devices, this architecture does not capitalize on all of the power of the LONWORKS system. LONWORKS devices in this architecture typically have limited decision-making responsibility and very limited interaction with devices on other parts of the hierarchy. Their only path of communication is through the proprietary gateways. This is a step forward from a completely proprietary system, but far from true openness. The system is still closed at the next level of the hierarchy, the supervisory controllers. These devices implement most of the control relationships between I/O devices, terminal units, and other supervisory controllers. These large control panels or "black boxes" also act as a gateway for the information from the standard LONWORKS protocol into some other transport mechanism. The system controllers are often used to provide

custom drivers for connectivity to another proprietary bus or to incorporate legacy equipment into the system. This is a non-interoperable, proprietary approach to solving the problem, and far from true openness. Each manufacturer has proprietary network tools for configuration and management. Further, each typically has proprietary HMI tools making it necessary for the integrator to spend time learning how to use a variety of interfaces without standards

A hierarchical system architecture is not the optimal control solution for a number of reasons. The most important reasons to the end user directly involve life cycle costs:

- *It is unnecessarily complex*. If the control system architecture were implemented with a true peer-to-peer structure, the controller-level network could be eliminated with no loss in functionality. The end-user derives no benefit from the extra level of the hierarchy and, in fact, is negatively affected by the extra cost and complexity associated with having to install, configure, and maintain a second control level network based on a different technology.

- *It is still proprietary*. Although the devices on the device network are LONWORKS and may even be built to the LONMARK® standard, the centralized controllers and the control algorithms they contain are not. They require custom programming with proprietary tools, and proprietary network management tools are required. This prohibits the end user from achieving one of the real goals of open standards: freedom of choice for modifications, additions, implementation of new functions, and maintenance.

- *It is not possible to communicate with any point, at any time, from anywhere on the network*. Because the architecture consists of multiple layers of control, it is not possible to communicate directly between devices on separate channels. Acquiring data translated through separate protocols twice and stored in a global database that may be minutes old is unacceptable. This architecture limits the information flow between devices, the ease of implementation of control algorithms, and ultimately the usefulness of the system. It can also significantly increase installation time.

The hierarchical system architecture is cumbersome and costly for end-users and systems integrators and it confuses the uninformed buyer who is led to believe they are purchasing an open system because it is based upon a technology that was conceived to provide openness. When implemented with LONWORKS networks at the lowest tier, the multi-tier control architecture is actually a collection of isolated LONWORKS networks. These LONWORKS networks contain relatively few peer-to-peer devices. In this architecture, even though there is interoperability on the device-level network, proprietary controllers provide system wide communication. LONWORKS devices are limited to sharing data directly with other LONWORKS devices on their local network only.

Instead of open network management software coordinating information transfer, there is proprietary black box software managing the controller-level network. This proprietary software is required because it attempts to hide the complexity of the multi-tier architecture from the end-user. The

manufacturer can therefore charge a premium for it and he can be sure the user will require his or her services at some point in the future.

3

The LONWORKS Protocol

This chapter introduces the LONWORKS protocol and describes a few of its most important features.

Introduction to the LONWORKS Protocol

The LONWORKS protocol, also known as the *LonTalk protocol* and the *ANSI/EIA 709.1 Control Networking Standard*, is the heart of the LONWORKS system. The protocol provides a set of communication services that allow the application program in a device to send and receive messages from other devices over the network without needing to know the topology of the network or the names, addresses, or functions of other devices. The LONWORKS protocol can optionally provide end-to-end acknowledgement of messages, authentication of messages, and priority delivery to provide bounded transaction times. Support for network management services allow for remote network management tools to interact with devices over the network, including reconfiguration of network addresses and parameters, downloading of application programs, reporting of network problems, and start/stop/reset of device application programs.

The *LONWORKS protocol* is a layered, packet-based, peer-to-peer communications protocol. Like the related Ethernet and Internet protocols, it is a published standard and adheres to the layered architectural guidelines of the International Standards Organization (ISO) Open Systems Interconnect (ISO OSI) reference model. The LONWORKS protocol, however, is designed for the specific requirements of control systems, rather than data processing systems. To ensure that these requirements are met with a reliable and robust communications standard, the LONWORKS protocol is layered as recommended by the International Standards Organization. By tailoring the protocol for control at each of the OSI layers, the LONWORKS protocol provides a control-specific solution that provides the reliability, performance, and robust communications required for control applications.

The seven layers of the ISO/OSI model, along with the corresponding services provided by the LONWORKS protocol, are shown in **Table 1**. This model is often used to compare the features and functionality of communication protocols. It is not a requirement that any given protocol implement every layer of this model or even that the layers be segmented as shown in the model. A truly complete and fully scalable protocol – such as the LONWORKS protocol – provides all the services described in this model.

Table 1 ISO/OSI Reference Model

	OSI Layer	*Purpose*	*Services Provided*
7	Application	Application Compatibility	Standard Objects and Types; Configuration Properties; File Transfer; Network Services
6	Presentation	Data Interpretation	Network Variables; Application Messages; Foreign Frames
5	Session	Control	Request-Response; Authentication
4	Transport	End-to-End Reliability	End-to-End Acknowledgement; Service Type; Packet Sequencing; Duplicate Detection

3	Network	Message Delivery	Unicast & Multicast Addressing; Packet Routing
2	Link	Media Access and Framing	Framing; Data Encoding; CRC Error Checking; Media Access; Collision Avoidance & Detection; Priority
1	Physical	Electrical Interconnect	Media-Specific Interfaces and Modulation Schemes (twisted pair, power line, radio frequency, coaxial cable, infrared, fiber optic)

Following is a summary of the services provided by each layer:

1 The *physical layer* defines the transmission of raw bits over a communication channel. The physical layer ensures that a 1 bit transmitted by a source device is received as a 1 bit by all destination devices. The LONWORKS protocol is media independent, so multiple physical layer protocols are supported depending on the communication medium.

2 The *link layer* defines media access methods and data encoding to ensure efficient use of a single communications channel. The raw bits of the physical layer are broken up into *data frames*. The link layer defines when a source device can transmit a data frame, and defines how destination devices receive the data frames and detect transmission errors. A priority mechanism is also defined to ensure delivery of important messages.

3 The *network layer* defines how message packets are routed from a source device to one or more destination devices. This layer defines naming and addressing of devices to ensure the correct delivery of packets. This layer also defines how messages are routed between the source and destination devices when these devices are on different communication channels.

4 The *transport layer* ensures reliable delivery of message packets. Messages can be exchanged using an acknowledged service, where the sending device waits for an acknowledgement from the receiver and resends the message if the acknowledgement is not received. The transport layer also defines how duplicate messages are detected and rejected if a message is resent due to a lost acknowledgement.

5 The *session layer* adds control to the data exchanged by the lower layers. It supports remote actions so that a client may make a request to a remote server and receive a response to this request. It also defines an authentication protocol that enables receivers of a message to determine if the sender is authorized to send the message.

6 The *presentation layer* adds structure to the data exchanged by the lower layers by defining the encoding of message data. Messages may be encoded as network variables, application messages, or foreign frames. Interoperable encoding of network variables is provided with standard network variable types (SNVTs).

7 The *application layer* adds application compatibility to the data exchanged by the lower layers. Standard objects promote interoperability by ensuring that applications use a common semantic interpretation of the data exchanged by lower layers. Common semantic interpretation ensures that different applications will exhibit common behavior for network variable updates. The application layer also defines a file transfer protocol that is used to transfer streams of data between applications.

All communications consists of one or more *packets* exchanged between devices. Each packet is a variable number of bytes in length and contains a compact representation of the data required for each of the 7 layers. The compact representation allows LONWORKS packets to be very short, minimizing implementation cost of every LONWORKS device.

Every device on a channel looks at every packet transmitted on the channel to determine if it is an addressee. If so, it processes the packet to see if it contains data for the device's application program or whether it is a network management packet. The data in an application packet is provided to the application program and, if appropriate, an acknowledgement, response, or authentication message is sent to the sending device.

The remainder of this chapter describes some of the most important aspects of the LONWORKS protocol.

Channel Types

The LONWORKS protocol is media-independent, allowing LONWORKS devices to communicate over any physical transport media. This empowers the network designer to make full use of the variety of channels available for control networks. The protocol provides for a number of modifiable configuration parameters to make tradeoffs in performance, security, and reliability for a particular application.

A channel is a specific physical communication medium (such as twisted pair or power line) to which a group of LONWORKS devices are attached by transceivers specific to that channel. Each type of channel has different characteristics in terms of maximum number of attached devices, communication bit rate, and physical distance limits. **Table 2** summarizes the characteristics of several widely used channel types.

Table 2 Widely-Used LONWORKS Channel Types

Channel Type	*Medium*	*Bit Rate*	*Compatible Transceivers*	*Maximum Devices*	*Maximum Distance*
TP/FT-10	Twisted pair, free or bus topology, opt. link power	78kbps	FTT-10, FTT-10A, LPT-10	64-128	500m (free topology) 2200m (bus topology)
TP/XF-1250	Twisted pair, bus topology	1.25Mbps	TPT/XF-1250	64	125m
PL-20	Power line	5.4kbps	PLT-20, PLT-21, PLT-22	Environment Dependent	Environment Dependent
IP-10	LonWorks over IP	Determined by IP network	Determined by IP network	Determined by IP network	Determined by IP network

Of particular importance is the free-topology twisted pair channel, TP/FT-10, which allows devices to be connected by single-twisted-pair wire segments in any configuration – no constraints on stub length, device separation, branching, etc; just a maximum length of cable per network segment.

Media Access

All network protocols use a *media access control (MAC)* algorithm to allow devices to determine when they can safely send a packet of data. MAC algorithms are designed to either eliminate or minimize collisions. A collision occurs when two or more devices attempt to send data at the same time. MAC algorithms that eliminate collisions are typically used in very small networks, since these algorithms do not scale well to large networks. Modern networks such as Ethernet use MAC algorithms that do not prevent, but instead minimize, collisions. The Ethernet MAC algorithm is not well suited to local control applications since it performs poorly under conditions of network overload. Existing MAC algorithms such as IEEE 802.2, 802.3, 802.4, and 802.5 do not meet all the LONWORKS requirements for multiple communication media, sustained performance during heavy loads, and support for large networks.

The LONWORKS protocol uses a unique media access control (MAC) algorithm, called the *predictive p-persistent CSMA protocol*, that has excellent performance characteristics even during periods of network overload. The LONWORKS MAC algorithm allows a channel to operate a full capacity with a minimum of collisions.

As with Ethernet, all LONWORKS devices randomize their access to the medium. This avoids the otherwise inevitable collision that results when two or more devices are waiting for the network to go idle so that they can send a packet. If they wait for the same duration after backoff and before retry, repeated collisions will result. Randomizing the access delay reduces collisions. In the LONWORKS protocol, devices randomize over a minimum of 16 different levels of delay called Beta 2 slots. Thus the average delay in an idle network is eight Beta 2 slots.

A unique feature of the LONWORKS protocol is that the number of available Beta 2 slots is dynamically adjusted by every device, based on an estimate of expected network loading maintained by each device. The number of available Beta 2 slots varies from 16 to 1008, depending on this estimate.

This method of estimating the backlog and dynamically adjusting the media access allows the LONWORKS protocol to minimize media access delays with a small number of Beta 2 slots during periods of light load, while minimizing collisions with many Beta 2 slots during periods of heavy load.

Addressing

The *addressing* algorithm defines how packets are routed from a source device to one or more destination devices. Packets can be addressed to a single device, to any group of devices, or to all devices. To support networks

with two devices to tens of thousands of devices, the LONWORKS protocol supports several types of addresses, from simple physical addresses to addresses that designate collections of many devices. Following are the LONWORKS address types:

- *Physical Address*. Every LONWORKS device includes a unique 48-bit identifier called the *Neuron ID*. The Neuron ID is typically assigned when a device is manufactured, and does not change during the lifetime of the device.
- *Device Address*. A LONWORKS device is assigned a *device address* when it is installed into a particular network. Device addresses are used instead of physical addresses because they support more efficient routing of messages, and they simplify replacing failed devices. A network installation tool that maintains a database of the device addresses for the network assigns the device addresses. Device addresses consist of three components: a domain ID, subnet ID, and node ID. The *domain ID* identifies a collection of devices that may interoperate. Devices must be in the same domain to exchange packets. There may be up to 32,385 devices in a domain. The *subnet ID* identifies a collection of up to 127 devices that are on a single channel, or a set of channels connected by repeaters. Subnet IDs are used to support efficient routing of packets in large networks. There may be up to 255 subnets in a domain. The *node ID* identifies an individual device within a subnet.
- *Group Address*. A *group* is a logical collection of devices within a domain. Unlike a subnet, however, devices are grouped together without regard for their physical location in the domain. There may be any number of devices in a group when unacknowledged messaging is used; groups are limited to 64 devices if acknowledged messaging is used. Groups are an efficient way to optimize network bandwidth for packets addressed to multiple devices. There may be up to 256 groups in a domain.
- *Broadcast Address*. A *broadcast address* identifies all devices with a subnet, or all devices within a domain. Broadcast addresses are an efficient method to communicate with many devices, and are sometimes used instead of group addresses to conserve the limited number of available group addresses.

Every LONWORKS packet transmitted over the network contains the device address of the transmitting device (the *source address*) and the address of receiving devices (*destination address*) that can either be a physical address, a device address, a group address, or a broadcast address.

Multiple domains are used if the number of devices exceeds the allowed domain limit or if there exists a desire to separate the devices so that they do not interoperate. It is possible for two or more independent LONWORKS systems to coexist on the same physical channel, as long as each system has a unique domain ID. Devices in each system respond only to those packets corresponding to their domain ID and do not know about or care about packets addressed with other domain IDs. Devices also respond to packets addressed with their own physical address, which is usually known only to the corresponding network installation tools. When a physical network is shared, overall network response times will be affected due to the increased number of packets, so coordinated overall network design is required.

Message Services

The LONWORKS protocol offers three basic types of message delivery service and also supports authenticated messages. An optimized network will often use all of these services. These services allow trade-offs between reliability, efficiency, and security, and are listed below:

- *Acknowledged Messaging*. Provides for end-to-end acknowledgement. When using acknowledged messaging, a message is sent to a device or group of up to 64 devices and individual acknowledgements are expected from each receiver. If acknowledgements are not received, the sender times out and retries the transaction. The number of retries and the timeout are both configurable.
- *Repeated Messaging*. Causes a message to be sent to a device or group of any number of devices multiple times. This service is typically used instead of acknowledged messaging because it does not incur the overhead and delay of waiting for acknowledgements. This is especially important when broadcasting information to a large group of devices, as an acknowledged message would cause all the receiving devices to try to transmit a response at the same time.
- *Unacknowledged Messaging*. Causes each message to be sent once to a device or group of any number of devices and no response is expected. This messaging service has the lowest overhead and is the most typically used service.
- *Authenticated Service*. Allows the receivers of a message to determine if the sender is authorized to send that message. Thus, authentication prevents unauthorized access to devices and is implemented by distributing 48-bit keys to the devices at installation time.

Network Variables

The LONWORKS protocol implements the innovative concept of *network variables*. Network variables greatly simplify the tasks of designing LONWORKS application programs for interoperability with multiple vendors' products and facilitating the design of information-based, rather than command-based, control systems. A network variable is any data item (temperature, a switch value, or an actuator position setting) that a particular device application program expects to get from other devices on the network (an *input network variable*) or expects to make available to other devices on the network (an *output network variable*).

The application program in a device doesn't need to know anything about where input network variables come from or where output network variables go. When the application program has a changed value for an output network variable it simply passes the new value to the device firmware. Via a process that takes place during network design and installation called *binding*, the device firmware is configured to know the logical address of the other devices or group of devices in the network expecting that network variable, and it assembles and sends the appropriate packets to these devices. Similarly, when the device firmware receives an updated value for an input network variable required by its application program, it passes the data to the application program. The binding process thus creates logical *connections*

between an output network variable in one device and an input network variable in another device or group of devices. Connections may be thought of as "virtual wires." If one device contains a physical switch, with a corresponding output network variable called `switch on/off`, and another device drives a light bulb with a corresponding input network variable called `lamp on/off`, creating a connection by binding these two network variables has the same functional effect as connecting a physical wire from the switch to the light bulb.

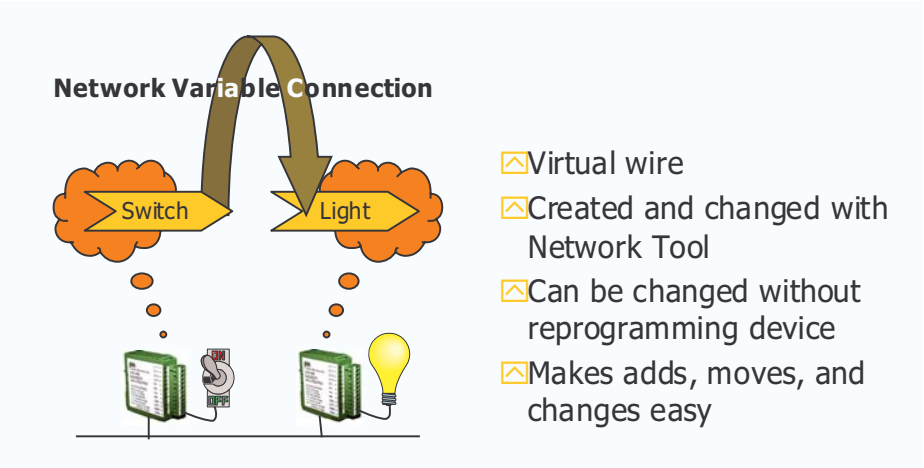

Figure 5 Network Variable Connection

Every network variable has a *type* that defines the units, scaling, and structure of the data contained within the network variable. Network variables must be the same type to be connected. This prevents common installation errors from occurring such as a pressure output being connected to a temperature input. Type translators are available to convert network variables of one type to another type. As described in the next chapter, a set of standard network variable types (SNVTs) is defined for commonly used types. Alternatively, manufacturers may define their own user-defined network variable types (UNVTs).

Network variables make possible *information-based control systems*, rather than old-style *command-based control systems*. This means that in a LONWORKS system, each device application makes its own control decisions, based on information it collects from other devices about what is going on in the system. In a command-based system, devices issue control commands to other devices, so a command-issuing device, that is typically a centralized controller, must be custom programmed to know a lot about the system function and topology. This makes it very difficult for multiple vendors to design standard control devices that can easily be integrated. Network variables make it easy for manufacturers to design devices that systems integrators can readily incorporate into interoperable, information-based control systems.

Limits

Each domain in a system using the LONWORKS protocol **can have up to 32,385 devices**. There can be **up to 256 groups in a domain** and each

group can have any number of devices assigned to it, except that when end-to-end acknowledgement is required, groups are limited to 64 devices. There can be **up to 255 subnets in a domain** and each subnet may have up to 127 devices. This information is summarized in **Figure 6**.

• Devices in a subnet	127
• Subnets in a domain	255
• Devices in a domain	32,385
• Domains in a network	2^{48}
• Maximum devices in system	$32K \times 2^{48}$
• Members in a group	
♦ Unacknowledged or Repeated	No Limit
♦ Acknowledged or Request Response	63
• Groups in a domain	255
• Channels in a network	No Limit
• Bytes in a network variable	31
• Bytes in an application or foreign frame message	228
• Bytes in a data file	2^{32}

Figure 6 LONWORKS Protocol Limits

The LONWORKS Protocol Standard

Up until a few years ago, the LONWORKS protocol was only available embedded in the Neuron Chip. This ensured consistent application by all manufacturers. Now that a large number of compliant devices have been installed, Echelon Corporation has published the LONWORKS protocol and made it an open standard under the ANSI/EIA 709.1 Control Networking Standard. The protocol is therefore freely available to anyone. To get a copy of the protocol specification, access global.ihs.com and request a copy of ANSI/EIA 709.1.

The most cost-effective manner in which to implement the LONWORKS communications protocol continues to be by purchasing a Neuron Chip. The ANSI/EIA standard, however, allows any company willing to undertake the investment to implement the protocol in the microprocessor of their choice.

Summary

In summary, the variety of services provided by the LONWORKS protocol allow for enhanced reliability, security, and optimization of network resources. The features and benefits provided by these services include:

- Supports a broad range of communication media, including twisted-pair wiring, power lines, and communication over IP networks.
- Supports networks constructed with a mix of media types and communication speeds.
- Supports efficient delivery of small messages, optimizing network usage for control applications.
- Supports reliable communication, including defense against unauthorized system use.

- Eliminates single points of failure, further increasing system reliability.
- Offers predictable response times independent of network size.
- Supports low-cost implementation of devices, tools, and applications.
- Minimizes installation and maintenance costs, resulting in lower life-cycle costs.
- Supports tens of thousands of devices — but is equally effective in networks with only a few devices.
- Permits flexible and easily reconfigurable connectivity among devices.
- Allows peer-to-peer communication thus allowing its use in both centralized and distributed control systems.
- Provides an effective mechanism for product interoperability, such that products of one manufacturer can share information about standard physical quantities with those of another manufacturer.